IgniteWorx: Design and Evaluation of a System-Supported Methodology for IIoT Project Setup

vorgelegt von

Dirk Slama (Dipl. Inform, MBA)

ORCID: 0000-0003-1588-4649

von der Fakultät VII – Wirtschaft und Management

der Technischen Universität Berlin

zur Erlangung des akademischen Grades

Doktor der Wirtschaftswissenschaften

- Dr. rer. oec. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Frank Straube, TU-Berlin
Gutachter: Prof. Dr. Rüdiger Zarnekow, TU-Berlin
Gutachter: Prof. Dr. Heiner Lasi, Ferdinand Steinbeis Institut

Tag der wissenschaftlichen Aussprache: 17.9.2019

Berlin 2019

Kurzfassung

Ein Projektmanager im industriellen Internet der Dinge (Industrial Internet of Things, IIoT) muss sich häufig vielen verschiedenen Herausforderungen stellen: neue Geschäftsmodelle, sich ändernde Geschäftsprozesse, umfassende interne und externe Stakeholder-Netzwerke und unvollständige Anforderungen. Außerdem zumeist ein komplexes technisches Umfeld mit Embedded Hard- und Software, Kommunikationstechnologie, neuen IIoT Backend-Anwendungen sowie der Integration in bestehende Anwendungslandschaften.

Diese Doktorarbeit ist im Bereich der Wirtschaftsinformatik angesiedelt. Die initiale Forschungsarbeit hat gezeigt, dass es heute einen Mangel an ausgereiften Methodenwerkzeugen zur Unterstützung des IIoT-Projektmanagers gibt. Als ersten Schritt in Richtung der Entwicklung solcher Methodenwerkzeuge adressiert diese Arbeit das Problem der IIoT Projektinitiierung. Der Stand der Forschung zeigt, dass die Projektinitiierung einen wesentlichen Einfluss auf den Erfolg des gesamten Projektes hat.

Ein wichtiges Ziel dieser Arbeit ist es, ein solides akademisches Fundament mit praktischer Anwendbarkeit zu kombinieren. Die Arbeit folgt den Prinzipien der konstruktionsorientierten Forschung. Sie beantwortet die Frage, wie eine systemunterstützte Methodik gestaltet sein kann, welche den IIoT-Projektmanager bei der Projektinitiierung unterstützt. Die Forschung wird in enger Zusammenarbeit mit einem Industrie-Konsortium durchgeführt, um dadurch auf ein starkes industrielles Netzwerk für den Erfahrungsaustauch zugreifen zu können.

Das zugrundeliegende Forschungsprojekt beinhaltet die initiale Iteration sowie ein weiteres Inkrement. In der ersten Iteration liegt der Fokus auf der Entwicklung und Evaluation der systemunterstützten Methodik. Als Name wurde IgniteWorx gewählt. “Ignite” bezieht sich dabei auf eine existierende IIoT-Methodik, die vom Autor dieser Arbeit mitentwickelt wurde. “Worx” bezieht sich darauf, dass das Ergebnis nicht abstrakt ist, sondern eine konkrete, systemunterstützte Methodik darstellt. IgniteWorx führt den Projektmanager durch ein strukturiertes Projekt-Assessment, dessen Ergebnisse dann als Grundlage für die Empfehlung einer konkreten IIoT-Projektstruktur genutzt werden. Die Arbeit untersucht sowohl die Inhalte der IIoT-Methodik, als auch die Anforderungen des Support-Systems. Für die Methodikinhalte liegt der Fokus auf dem Projekt-Assessment, den möglichen IIoT-Projektstrukturen sowie den Beziehungen zwischen diesen beiden Elementen. Der IIC Project Explorer dient zur Demonstration des Ansatzes. In der zweiten Iteration wird eine Erweiterung zum Erfassen von Nutzer-Feedback entwickelt. Die Evaluation basiert auf einer Fallstudie, die vier Projekte aus den Bereichen Fabrikation, Automobilindustrie und Gebäudemanagement umfasst. Die Auswertung zeigt, dass IgniteWorx anwendbar und relevant ist. Damit ist die Forschungsfrage hinreichend beantwortet.

Abstract

The challenges that a project manager in the Industrial Internet of Things (IIoT) must potentially face are manifold. These include new and unproven business models, new or changing business processes, complex internal and external stakeholder networks, and frequently changing or partly ill-defined requirements. Further challenges are related to a broad range of technical topics including embedded hardware and software, local connectivity, global communication networks, new IIoT backend applications, and integration with existing applications.

This thesis is based on information systems research. Initial research has shown that there is a lack of proven, IIoT-specific project management methodologies available to support a project manager in addressing these challenges. As a first step toward the development of such a methodology, this thesis is addressing the problem of IIoT project setup. As can be derived from existing research on traditional IT projects, the initial setup shapes the entire project and has a strong impact on its success.

An important goal of this thesis is to combine a sound academic foundation with practical applications. It is following the principles of design science as an outcome-oriented research methodology. The thesis answers the question of how a system-supported methodology should be designed, which supports project managers in finding a suitable setup for their IIoT projects. The research is done in close collaboration with an industry consortium to leverage a strong industrial network for best practices and feedback.

The underlying research project includes the initial iteration, plus one incremental iteration. In the first iteration, the focus is on designing and evaluating such a system-based methodology. The name IgniteWorx was chosen; "Ignite" refers to an existing IIoT methodology, which was co-authored by the author of this thesis, and "Worx" refers to the fact that the result is not an abstract one but is specifically designed for support by an online system. IgniteWorx supports the IIoT project manager by guiding him or her through a project assessment, which is then used to generate recommendations for the project setup. For this, the thesis examines both the content side as well as the system support side. On the content side, the main focus is on how to structure the IIoT project assessment, the recommendations for the project setup, and the required mappings between these two perspectives. A concrete demonstration of the resulting design is introduced, based on the IIC online Project Explorer tool. In the second, incremental iteration, the focus is on extending IgniteWorx to allow for incorporation of user feedback on the recommendations, effectively creating a built-in quality improvement process. The evaluation is based on a case study that includes four industrial projects from manufacturing, automotive, and building management. The cross-case findings conclude that IgniteWorx can be applied in practice and delivers relevant results. This means that the research question has been adequately answered.

Keywords: Industrial Internet, Internet of Things, Industry 4.0, Project Management, Solution Design, Resource Acquisition, Cost Estimation, Risk Management, Trust and Security, Reliability and Resilience, Verification and Validation, Service Operations, Expert Systems, Recommender Systems, Semantic Systems

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
APM	Asset Performance Management
ASIC	Application-Specific Integrated Circuit
ASIL	Automotive Safety Integrity Level
ATAM	Architecture Tradeoff Analysis Method
AOI	Automated Optical Inspection
BN	Bayesian Network
CBM	Condition-Based Maintenance
CDD	Capability-Driven Development
CDN	Contend Distribution Network
CIoT	Consumer Internet of Things
CMDB	Configuration Management Database
CMMI	Capability Maturity Model Integration
CoE	Center of Excellence
COTS	Commercial off the Shelf
COCOMO	Constructive Cost Model
CPM	Critical Path Method
CPS	Cyber Physical Systems
DA	Decision Analysis
DAS	Distributed Agent Systems
DDoS	Distributed Denial-of-Service Attack
DMZ	Demilitarized Zone
DSL	Domain-Specific Language
DSR	Design Sciences Research
DSS	Decision Support System
DTA	Decision Tree Analysis

EAM	Enterprise Asset Management; also Enterprise Application Management
ECU	Electronic Control Unit
ELDA	Event-driven Lightweight Distilled Statecharts Agents
FEDS	Framework for Evaluation in Design Science
FOTA	Firmware over the Air (Update)
FPA	Function Point Analysis
FSM	Field Service Management
GPU	Graphical Processing Unit
HIL	Hardware in the Loop
HMI	Human-Machine Interaction
IIC	Industrial Internet Consortium
IIoT	Industrial Internet of Things
IoE	Internet of Everything
IoT	Internet of Things
IPT	IIoT Project Toolkit
IT/OT	Information Technology/Operational Technology
ITIL	Information Technology Infrastructure Library
ITS	Intelligent Transportation Systems
ITSM	IT Service Management
ISMS	Information Security Management System Standards
IIRA	Industrial Internet Reference Architecture
M2M	Machine to Machine
MCDA	Multi-Criteria Decision Analysis
MDA	Model-Driven Architecture
MDP	Markov Decision Process
MES	Manufacturing Execution Systems
MIL	Model in the Loop
ML	Machine Learning

MTBF	Mean Time between Failure
OEE	Overall Equipment Effectiveness
OPM3	Organizational Project Management Maturity Model
OR	Operations Research
OTA	Over the Air
OTA	Online Trust Alliance
OWL	Ontology Web Language
PCB	Printed Circuit Board
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PMM	Project Management Methodology
PoC	Proof of Concept
QAS	Quality Assurance System
RAD	Rapid Application Development
RAMI	Reference Architecture Model of Industrie 4.0
RBS	Risk Breakdown Structure
RFI	Request for Information
RFP	Request for Proposal
RFQ	Request for Quotation
RS	Recommender System
RMA	Resilience Modeling and Analysis
SDM	Software Development Method
SIG	Special Interest Group
SIL	Software in the Loop
SME	Situational Method Engineering
SOA	Service-Oriented Architecture
SOC	System on Chip
SOP	Start of Production
SOTIF	Safety of the Intended Functionality

SPICE	Software Process Improvement and Capability Determination
SPL	Software Product Line Process
SR	Semantic Reasoner
SSL	Secure Socket Layer
SVM	Support Vector Machine
SVVP	Software Verification and Validation Plan
TCO	Total Cost of Ownership
TCU	Telematic Control Unit
TOGAF	The Open Group Architecture Framework
TOGAF ADM	TOGAF Architecture Development Method
TPM	Trusted Platform Module
UI	User Interface
UML	Unified Modeling Language
VDA	German Association of the Automotive Industry
VDA QMC	Quality Management Center of the VDA
VPN	Virtual Private Network
WBS	Work Breakdown Structure
WMFP	Weighted Micro Function Points

Table 1: List of abbreviations

Acknowledgments

This thesis would not have been possible without the support of my family, for which I am extremely grateful. A number of lifechanging events happened along the way, including—to name but a few—one wedding, three children, and a new home. Finding the time to work on this thesis was not always easy, either for me or the rest of the family, and still you supported me wherever possible. Thank you for that (and the rest!).

A special thank-you goes to Prof. Zarnekow. His continuous support, guidance, and persistent questioning was a significant success factor for this project, as well as his patience with the aforementioned lifechanging events. Thank you for believing in this project.

Another special thank-you goes to Prof. Lasi. The collaboration with the German Country Team of the IIC, as well as his constant drive to apply scientific concepts to real-world, industrial testbeds, was another important success factor for this project.

The inspiring work with my master students—Philip Richert, Max Ficht, Vishnu Sassikumar, and Marten Oberpichler—has helped to significantly advance different important aspects of this thesis. The same holds true for the collaboration with the PhD students in the IIC German Country Team, especially Daniel Burkhardt, Patrick Weber, and Simon Hiller. Thank you all.

Furthermore, I would like to thank the IIC for its support of this thesis: the Steering Committee for allowing me to use feedback from the IIC Project Explorer in the second project iteration, as well as a general thank-you for the opportunity; Stephen Mellor, CTO of the IIC; Jacques Durand from the IIC BSSL WG; as well as many IIC members for the interesting discussions and brainstorming; the Quoin Team—Huu Da Tran and Matt Counts—for technical excellence in the implementation of the IgniteWorx concepts for the IIC; and, of course, Andreas Kraft for his UX ingenuity.

Thank you to Dr. Marco Lang and Dr. Stefan Ferber for giving me the opportunity; Antonio Martinez, Peter Strink, Martin Dölfs, Dr. Nils-Holger Schmidt, Kai Hackbarth, Laurenz Kirchner, and Tiemo von Hinckeldey for their feedback on the application of IgniteWorx to real-world projects; Oisin Benson for his insights into the state of the art of artificial intelligence; Dr. Dirk Krafzig for his insights on IoT solution design and SOA; and, finally, my Enterprise IoT co-authors, Dr. Frank Puhlmann, Jim Morrish, and Dr. Rishi Bhatnagar, for laying the foundation.

Thank you all!

1 Introduction

This thesis presents an ambitious research project, driven by the goal to provide IIoT project managers with actionable tools to help them manage their projects more successfully by reducing time to market, keeping project costs under control, managing project risks more effectively, and, finally, delivering high-quality solutions that fulfill customer expectations.

Initial research has shown that there is a lack of proven, IIoT-specific project management methodologies available (see section 1.4.2) to support a project manager in addressing the typical challenges of an IIoT project (see section 1.4.1).

As a first step toward the development of such a methodology, this thesis addresses the problem of IIoT project setup. As can be derived from existing research on traditional IT projects, the initial setup shapes the entire project and has a strong impact on its success (see section 2.1).

To set the context, the introduction first provides an overview of the structure of this thesis. This is followed by an in-depth discussion of opportunities presented by the IIoT, as well as the specific challenges faced by an IIoT project manager. Derived from this, the research question and methodology are introduced.

1.1 Structure of This Thesis

As described later in the discussion of the research methodology (see section 1.6), this thesis is based on design sciences research (DSR), following the specific approach from Peffers et al. (2007). The structure of the thesis closely mirrors the selected DSR process. As shown in Figure 1, the main elements of this thesis are the introduction, iterations I and II, and, finally, conclusions and outlook.

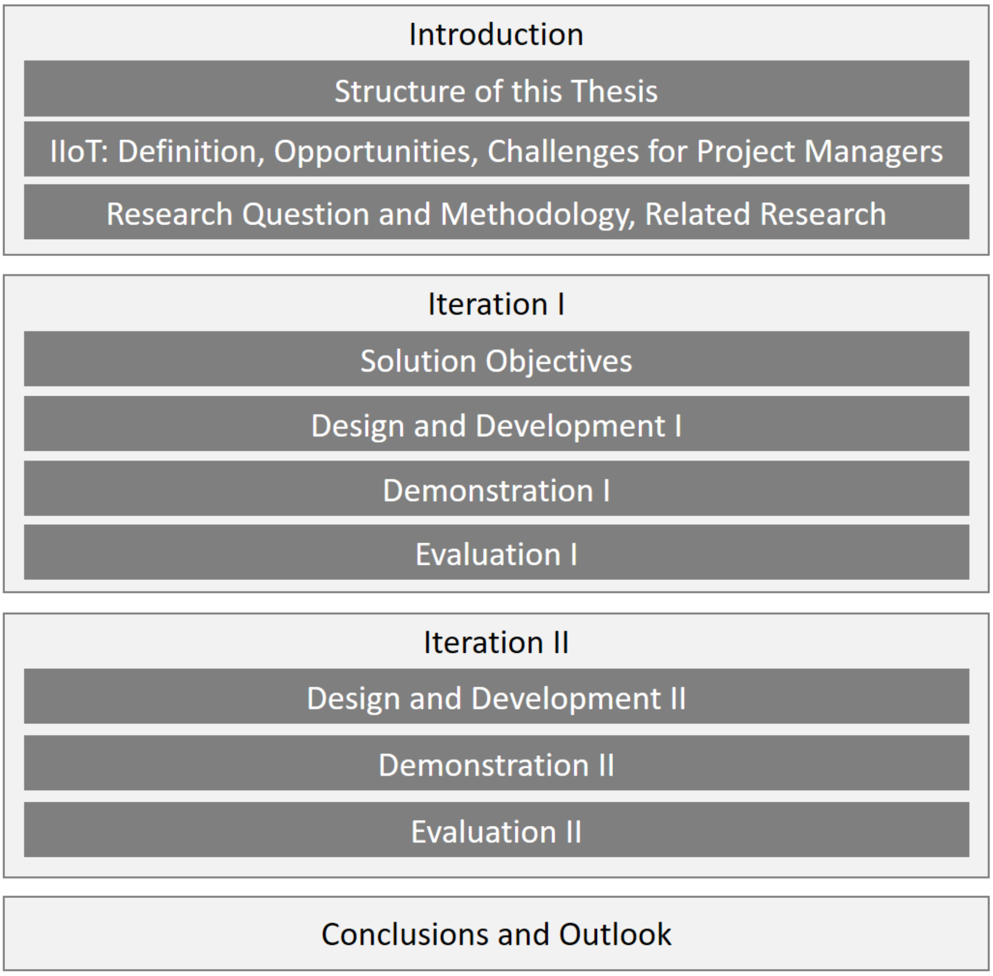

Figure 1: Structure of this thesis

The introduction outlines opportunities and challenges in IIoT, especially from the perspective of the project manager. Based on this analysis, a discussion of the matching research methodology and the approach chosen for this thesis is presented, including the research question.

The first iteration includes all required DSR process phases, from solution objectives to evaluation. The main goal of this first iteration is to design and evaluate the core of a system-supported methodology that supports project managers in finding a suitable setup for their IIoT projects.

The second iteration is an incremental step and provides some in-depth design extensions based on the findings from the first iteration.

The conclusions and outlook summarize the key findings from both iterations and provide an outlook into planned future research.

1.2 IIoT: Definition and Delimitation

The focus of this thesis is on methodologies for the Industrial Internet of Things (IIoT). However, it is difficult to provide a clear and succinct definition of IIoT. One problem is that there are many related concepts, with partially overlapping meanings. Related terms and concepts include, for example, Internet of Things (IoT), Consumer Internet of Things (CIoT), Internet of Everything (IoE), Machine-2-Machine (m2m), Cyber Physical Systems (CPS), and Industry 4.0 (sometimes also Industrie 4.0).

The Industrial Internet Consortium defines IIoT as the “Internet of things, machines, computers and people, enabling intelligent industrial operations using advanced data analytics for transformational business outcomes,” according to Karmarkar et al. (2018).

However, there are many alternative proposals. Based on a survey of forty web sites (top-ranking matches on Google and Google Scholar), the minimal consensus seems to be that the IIoT focuses on industrial use cases or applications.

A number of differentiating factors can be identified based on this research, including:

- Industries
- Use cases
- Technologies
- Reach and scope
- Agility
- Others

The first—and most often cited—differentiating factor is the range of industries addressed by the different concepts. Table 2 shows an overview of the findings based on the aforementioned web survey. IoT is usually seen as an umbrella term, supporting all industries. The industrial IoT focuses on industrial applications in general, e.g., manufacturing, utilities, and oil and gas. Healthcare is included in some articles. Vehicles also seem to be supported by IIoT. This is unambiguous for industrial vehicles, trucks, etc., while for cars it is not always clear. The consumer IoT is largely seen as supporting consumer devices and smart homes. Some articles also refer to cars and healthcare. Industry 4.0 is mainly associated with manufacturing.

	IoT	Industrial IoT	Consumer IoT	Industry 4.0
Consumer Devices	✓		✓	
Smart Home	✓		✓	
Vehicles	✓	✓	(✓)	
Healthcare	✓	(✓)	(✓)	
Manufacturing	✓	✓		✓
Other Industries	✓	✓		

Table 2: Support for different industries

Another key differentiator is the use case perspective. For example, in Buntz (2017), the following use cases are described as the most common for IIoT (based on interviews of 73 professionals with involvement in IoT projects):

1. Asset tracking and monitoring
2. Automation of manual processes
3. Predictive maintenance
4. Improving safety and security
5. Buildings: energy efficiency and/or automation
6. Enhanced customer engagement and customer satisfaction
7. Data intelligence to do strategic planning
8. Transforming from a product-based model to a services-based model for customers
9. More agile and efficient product design process

Most of the above examples would apply to all of the industries summarized in Table 2. One special case is the fifth point, which would only apply to buildings. There are many other examples seen as typical IIoT use cases. For instance, Tracy (2017) adds smart metering and fleet management to the list.

Some analyst firms also use the concepts of reach and scope to differentiate among the different IoT categories. For example, Lueth (2014) defines "reach" as "who/what is impacted by the concept" and "scope" as "what is being altered by the concept." As examples for reach, Lueth includes machines (narrow scope), objects and devices, people, and the world (broadest reach). As examples for scope, he includes the virtual world (narrow scope) and the physical world (broad scope). In this categorization, m2m would support the narrowest scope and reach, while IoT has the broadest reach and Industry 4.0 the broadest scope.

MachinaResearch further differentiates among Intranets of Things, Subnets of Things, and Internet of Things. Here, the idea is that these concepts can be mapped to scope and agility (Slama, Puhlmann et al., 2015). Scope in the MachinaResearch definition ranges from standalone connected devices (narrow scope) to fully cross-enterprise solutions (broad scope). Agility is defined as fixed parameters/homogeneous assets (low agility) to semantically rich/highly diverse assets (high level of agility).

For the purpose of this thesis, the definition of the IIC (Karmarkar et al., 2018) is used, as shown earlier, since it has a number of benefits:

- It starts with the Internet of Things; this means that the available research on IoT can also be included in this thesis. In fact, IoT and IIoT are used interchangeably throughout the text unless specifically differentiated where need be.

- It then includes machines, computers, and people, which is in line with most of the other definitions surveyed.
- The definition then mentions the enablement of "intelligent industrial operations," which supports all of the IIoT industries and use cases outlined above.
- The reference to "advanced data analytics" is sufficiently broad.
- Finally, the reference to "transformational business outcomes" is in line with the goals of this thesis, especially to provide actionable support for project managers to achieve this.

1.3 IIoT: Opportunities

IoT, IIoT, and related concepts have seen very aggressive growth projections, e.g., in Lund et al. (2014). Unfortunately, these projections are not currently supported by scientific research (according to research on Google Scholar).

However, there is an abundance of material available that provides qualitative arguments for IIoT opportunities. For example, Porter and Heppelmann (2014b) offer a comprehensive overview of the transformational potential of the IoT/IIoT (they are referring to the IoT, but all use cases satisfy the criteria for IIoT given above).

Prof. Porter is well known in the industry for his work on value chain analysis (Porter, 1998a), as well as competitive forces (Porter, 1998b). Looking at the IIoT opportunity through these lenses, Porter and Heppelmann are making the case for *"smart, connected products"* (Porter and Heppelmann, 2014b): "Smart, connected products offer exponentially expanding opportunities for new functionality, far greater reliability, much higher product utilization, and capabilities that cut across and transcend traditional product boundaries. The changing nature of products is also disrupting value chains, forcing companies to rethink and retool nearly everything they do internally. These new types of products alter industry structure and the nature of competition, exposing companies to new competitive opportunities and threats. They are reshaping industry boundaries and creating entirely new industries."

Again, the boundaries between the IoT and IIoT are fluid in the way smart, connected products are described here. However, since most examples given in the paper match the definition of IIoT from above, this can be seen as a strong supportive argument for IIoT opportunities.

Slama, Durand et al. (2015) describe the impact of the IIoT as a transformational driving force on the value chain of enterprises (see Figure 2). By being able to connect previously unconnected products, new business models can be enabled along the value chain. This includes the use of IIoT field data to help optimize product designs, as well as using it to improve marketing and sales, distribution, and aftermarket services.

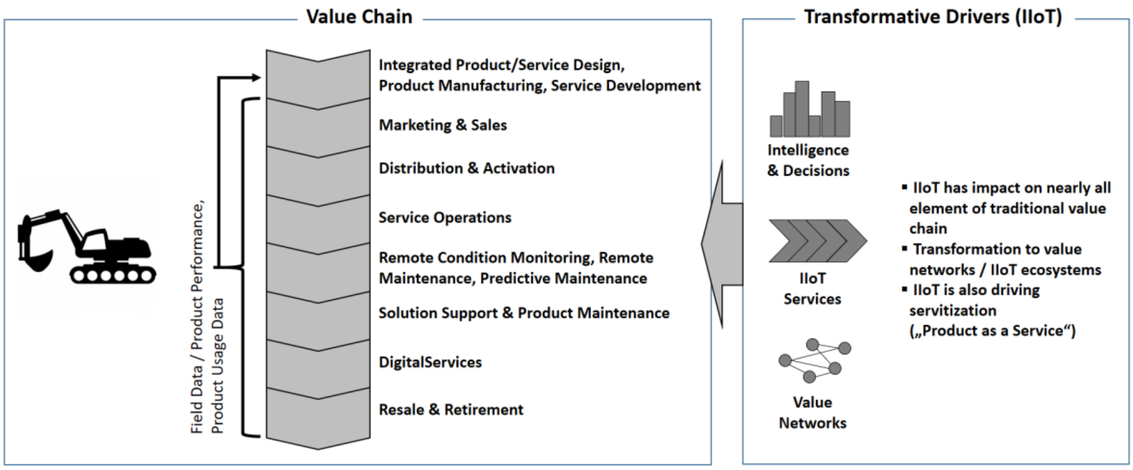

Figure 2: Transformational impact of IIoT on existing value chains (Slama, Durand et al., 2015)

Additional scientific research also supports these arguments. For example, Popescu and others (2015) provide a study that concludes, "The overall results provide strong evidence for the IIoT's capacity to determine economic growth, the consequences and soundness of the IIoT's economic expansion, and the upshot of utilizing IoT technologies in outstanding economic spheres."

To leverage the opportunities provided by the IIoT described here, the challenges in successful IIoT adoption must be understood as well before looking at how best to overcome them.

1.4 IIoT: Challenges for Project Managers

The adoption of emerging concepts like IoT/IIoT does not come without challenges. Gartner Group has been doing research on the adoption of emerging technologies for decades. In the 2018 Hype Cycle for the Internet of Things, Gartner has located IoT itself in the so-called "Sliding into the Trough" phase. This is no scientific evidence but an indicator that IoT adoption in fact is challenging since Gartner conducts many customer interviews each year.

Since the focus of this thesis is on IoT and IIoT project management, the following sections first look at IoT/IIoT project failure rates and risks. Second, existing IoT/IIoT project management methodologies and their potential shortcomings are examined.

1.4.1 IoT Project Risks and Failure Rates

There is currently very little scientific work specifically focused on IoT—or even IIoT—project failure rates and risks (using Google Scholar with search terms like "IoT challenges" or "IoT project risks" yields no satisfying results).

However, there are a number of industry studies and whitepapers examining the topic.

1.4.1.1 Cisco Survey

Cisco has performed a web-based survey on IoT project risk factors and failures. The survey includes 1,845 IT and business decision makers, working for companies with at least 100 employees. Participants confirmed that they are working in an organization in the process of completing or that has already completed IoT initiatives. Industries included retail, hospitality, energy, transportation, manufacturing, local governments, and healthcare (which would actually make this survey qualify as an IIoT survey, according to the definition in section 1.2).

According to the Cisco survey, only 26 percent of all surveyed companies are successful with their IoT initiatives. Five main reasons are mentioned as key factors slowing IoT progress:

- time to completion
- quality of data
- internal expertise
- IoT integration
- budget overruns

Sixty percent of survey participants stated that IoT initiatives are proving more complex in reality than initially anticipated.

1.4.1.2 Association of Equipment Manufacturers

The Association of Equipment Manufacturers published an article called “5 Reasons IoT Projects Fail” (Weis, 2018), which is based on input from the international consulting firm McKinsey & Co. The five reasons provided are as follows:

1. Trying to do everything at once
2. Adhering to a rigid development and deployment cycle
3. Taking a “lone wolf” approach to technological development
4. Treating an IoT initiative as a simple technology project
5. Failing to adapt organizational capability, culture, and processes

1.4.1.3 OpenSensors.io Interview

Yodit Stanton, CEO of OpenSensors.io, has created a list of ten reasons IoT projects fail (Stanton, 2017). While the article is kept in colloquial language, the concrete examples provided indicate that the content comes from a practitioner with real-world experience. The ten reasons are:

1. Thinking of IoT as one industry (referring to the heterogeneity of the industry and the need for industry-specific solutions).
2. Everyone is becoming a tech magpie (referring to the difficultly of selecting appropriate technology from a large set of options).

3. Talking about that architecture, middleware is only 1 percent of the solution.
4. Thinking you can data jujitsu your way out of crap readings from lots of cheap sensors (referring to the importance of investing in appropriate hardware and sensors).
5. Not building a multidisciplinary team (referring to the need to include "hardware people, software people, networking specialists, QA people, project managers").
6. Not spending 40 percent or more of your time and money on logistics (referring to the complexity of managing the distribution network for physical assets, like sensors).
7. Not understanding the importance of provisioning until after shipping.
8. Trying to run before you can walk (referring to the need to scale IoT solutions incrementally).
9. Not learning from computing history (recommended to avoid reinventing the wheel).
10. Not appreciating how big of a deal all this is (referring to the transformational potential of the IoT).

1.4.1.4 Enterprise IoT

Finally, Slama, Puhlmann et al. (2015) also provide a number of IoT project challenges, based on the many expert interviews in the book. One point that is stressed multiple times is the complexity of an IoT project caused by the many disciplines typically required for IoT implementation, including enterprise software development, embedded software development, potentially global telecommunication networks, data analytics, hardware development, etc. The fact that IoT projects usually involve dealing with assets deployed in the field is also discussed in detail, including many resulting challenges, like having to deal with the asset`s manufacturing organization, regulatory requirements, and resulting technical challenges.

In addition, Slama, Puhlmann et al. (2015) also offers a discussion of the "clash of two worlds," as summarized in Figure 3. This refers to the different cultures that must typically be aligned in an IoT project: on one side, the so-called "machine camp," and on the other, the "Internet camp." The culture in the machine camp is described as one dominated by manufacturing experts with a high level of risk aversion caused by experiences in the manufacturing world, working on long-term projects with long lead times, and using waterfall-like project setups driven by many requirements for verification and validation. The Internet camp, on the other hand, is described as a high-risk culture with extremely short project release cycles and a focus on point solutions and minimal-viable-product philosophy, using agile project methods.

Many Internet-based applications are described as perpetual beta versions, with many fast patches and updates. The physical assets, on the other hand, require a long lead time for the physical product design and manufacturing line setup and allow no easy updates to the physical configuration afterward.

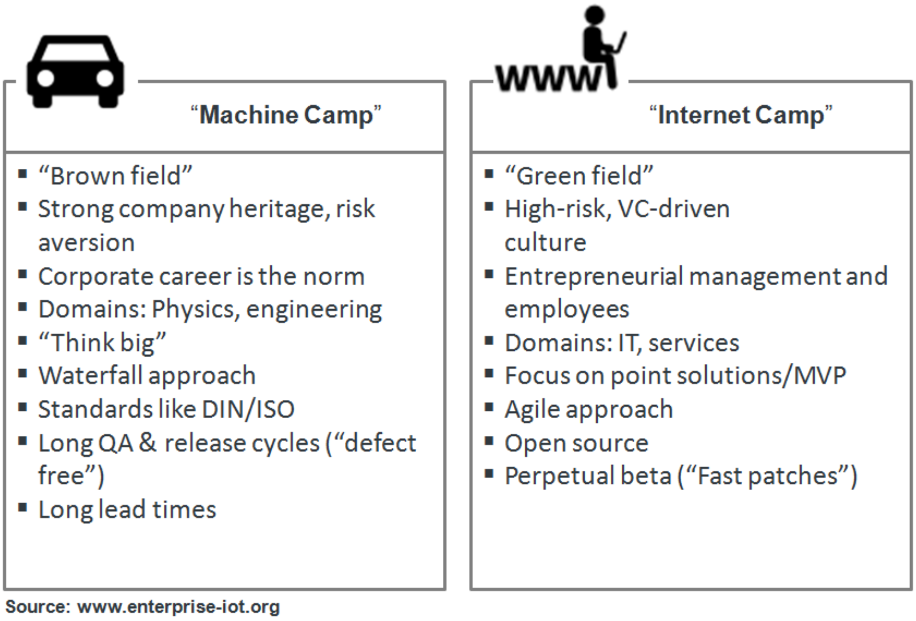

Figure 3: Clash of two worlds (Slama, Puhlmann et al., 2015)

1.4.1.5 Lessons from IT Projects—Standish Report

IIoT projects usually include hardware and software development, so naturally the perspective of the “normal” IT world on project risks and failure rates is also an interesting one in this context. One example of a long-running report on this topic is the so-called Standish “CHAOS” Report (Standish Group, 2009), which has published project benchmarks since 1994 with fairly dramatic project failure rates. As shown in the excerpt in Table 3, project failure rates started with 84 percent in 1994 and fell to 68 percent in 2009.

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
2000	28	49	23
2009	32	44	24

Table 3: Evolution of Standish benchmark (excerpt)

The report also proves a ranking of project success factors based on a survey of executive IT managers. The list is as follows:

1. user involvement
2. executive management support
3. clear statement of requirements
4. proper planning
5. realistic expectations
6. smaller project milestones

7. competent staff
8. ownership
9. clear vision and objectives
10. hardworking, focused staff

The Standish report applies to IT projects, not to IoT projects. However, since every IoT project usually also includes a significant IT subproject, this input is also relevant here.

1.4.1.6 Lessons from Hardware Projects

The other aspect of an IIoT project is often the hardware side. Especially if the IoT project involves custom hardware design and manufacturing, these risks must also be understood. Again, this topic seems not to be supported by detailed scientific evidence. However, there are some potentially relevant individual contributions to be found on the Internet.

Eric Graves is an aerospace and mechanical engineer. In his blog (Eric Graves, 2016), he identifies the following risks for hardware development projects:

1. Lack of control over the hardware supply chain
2. Manufacturability, i.e., the ability to manufacture the initial prototype
3. Meeting unit cost requirements
4. Gaps in product/market fit
5. Scope changes during the project
6. Ineffective communication
7. Regulatory challenges

1.4.1.7 Limited Reliability of Quoted Sources

As stated before, there is almost no peer-reviewed, scientific research regarding IoT project risks and failure rates. In this thesis, the aim is to identify individual and industrial contributions regarding this topic, e.g., based on surveys or expert statements.

However, all arguments brought forward here are circumstantial evidence at best, and some of the surveys cited here have actually been heavily scrutinized. Notably, Eveleens and Verhoef (2010) published a paper entitled “The Rise and Fall of the Chaos Report Figures” in which they use statistical methods to prove that the numbers in the reports are “highly influenced by forecasting biases.” Similar arguments are introduced in Glass (2005).

This means that if the criticisms of the numbers in the Standish surveys are valid, the numbers in other surveys, like the Cisco IoT project survey (Cisco, 2017), must also be critically reviewed.

1.4.1.8 Conclusions

No scientific proof for IoT project failure rates could be produced. Similarly, no scientifically validated list with project risks could be produced, less so a scientifically validated weighing of potential risks in IoT projects.

However, the discussion shows that there is a significant level of complexity in IoT projects, resulting in a long list of potential risks to be mitigated.

Consequently, it makes sense to look at the availability of established IoT project management methodologies, which can help address these risks and ensure a smooth and efficient project execution.

1.4.2 Lack of established IIoT Project Management Methodologies

Improving project performance in general is a top priority of most organizations, according to Wysocki (2013) and Yardley (2002). Wells also (2012) concludes, “As a way of addressing this [need for improving project performance], project management methodologies (PMMs) are regularly employed with the aim of increasing project efficiency and effectiveness.”

So it would seem logical to also look for IIoT project management methodologies to address the issues outlined in the previous section, helping to deliver IIoT projects successfully, on time, and on budget. However, the question is if there are yet sufficiently well-proven and well-established IoT or even IIoT project methodologies to support this.

In Jacobson et al. (2017), the authors ask, “Is There a Single Method for the Internet of Things?” concluding that, given the complexity of IoT projects, it seems more likely that an IoT method must combine elements of existing methodologies and then add IoT-specific elements to it.

In Giray and Tekinerdogan (2018), the authors look at “Situational Method Engineering for Constructing Internet of Things Development Methods.” As part of their research, they identified six software development methods (SDMs) with a focus on IoT, as summarized in Table 4 (using the same abbreviations as proposed in this paper).

Method	Abbreviation
Ignite | IoT methodology	Ignite
IoT methodology	IoT-Meth
IoT application development	IoT-AD
Event-driven lightweight distilled state charts-based agents methodology	ELDAMeth
A software product line process to develop agents for the IoT	SPLP-IoT

A general software engineering (SE) methodology for IoT	GSEM-IoT

Table 4: Software development methods for IoT

While the list from Giray and Tekinerdogan (2018) seems comprehensive based on additional research (e.g., on Google Scholar), one important question is how far the methodologies outlined in it are well established and widely used today.

To help answer this question, each methodology must be examined with respect to a metric that will help to judge the level of adoption. Inspired by metrics often found in the Open Source community (see, for example, Link et al. (2017)), the following metrics are applied:

- Is there a significant number of academic papers supporting the methodology?
- Is there a significant number of blogs or similar sources with frequent contributions and high readership?
- Is there a professional organization behind the methodology?
- Are professional training courses available?
- Are software tools available to support the methodology?
- Are there success stories supporting the methodology?

The following looks at each of the methodologies identified by Giray and Tekinerdogan (2018), followed by a summary and conclusions.

1.4.2.1 Ignite

The Ignite | IoT methodology was published as part of the Enterprise IoT book by Slama, Puhlmann et al. (2015). Google Scholar shows forty-five citations (Sept. 8, 2018). The book's website provides a complete online version of Ignite, which has been adopted by the IIC as part of its IPT toolkit. Ignite has also been adopted by the Eclipse Foundation as part of the Eclipse IoT Open Source project (see (Eclipse Foundation).

Ignite consists of two main elements: Ignite | IoT strategy execution and Ignite | IoT solution delivery. This thesis mainly focuses on the solution delivery, which includes the project perspective. The Ignite | Iot solution delivery is usually triggered as part of the IoT project initiation, as described in the previous section. Ignite first separates between the perspective of the asset at the center of the IoT project on one hand and the actual IoT project on the other hand. Asset-related activities include product design and manufacturing. The main IoT project activities are embedded IT project delivery, telecommunications project, and traditional enterprise IT project. Figure 4 provides an overview of the main elements.

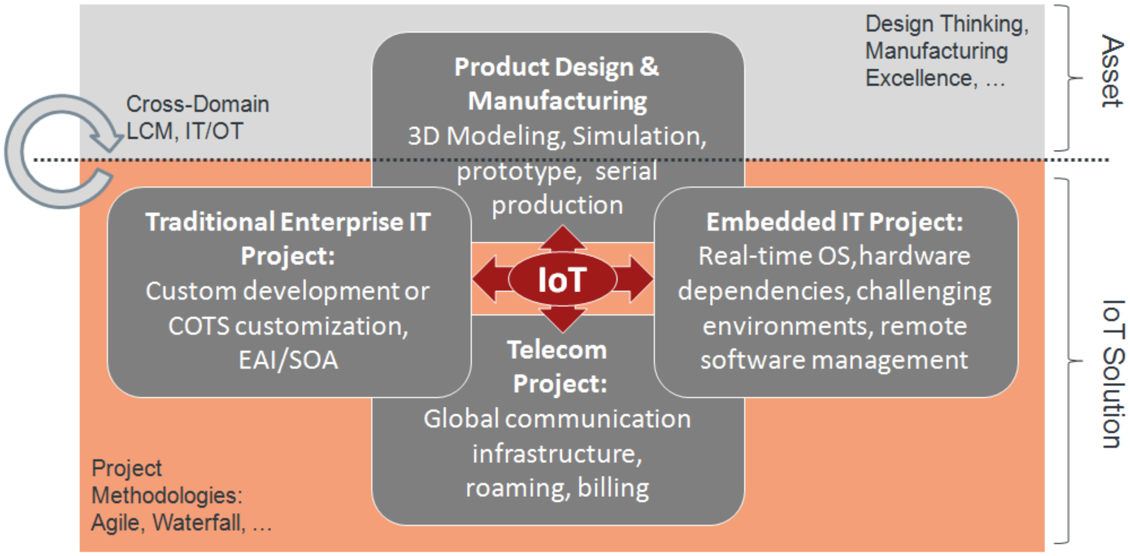

Figure 4: Ignite IoT solution delivery (Slama, Puhlmann et al., 2015)

The asset-related activities defined in Ignite include 3D modeling, simulation, prototyping, and serial production. Key capabilities mentioned are design thinking and manufacturing excellence.

The elements of the IoT solution are broken down into:

- Embedded IT project: This subproject is responsible for delivering all required hardware and software components to be deployed on or near the asset/in the field.
- Telecom project: This subproject is responsible for providing the—potentially global—communications infrastructure required for the solution.
- Enterprise IT project: This subproject is responsible for delivering the new IoT backend solution (which connects to the embedded solution via the telecommunications infrastructure), as well as the integration with existing backend systems.

To align these different perspectives, Ignite has developed a generic framework for the organizational setup of an IoT project, as shown in Figure 5. This organizational setup defines six main workstreams for the development of the IoT solution:

- Project Management: Responsible for overall project management and alignment of the different workstreams, including IoT solution architecture management
- Cross-Cutting: Responsible for activities cutting across the other workstreams, including security, asset lifecycle management, and solution integration and testing
- Solution Infrastructure and Operations: Responsible for setting up required infrastructure and operations processes, including DevOps, solution support organization, etc.

- Backend Services: Implementation and operations of new IoT backend applications and integration into existing application landscape and business processes
- Communication Services: Responsible for acquisition, setup, and management of required communication services
- On-Asset Components: Responsible for all hardware and software components deployed on the asset

In addition, Ignite proposes to set up another workstream, which is concerned with asset preparation. This workstream is designed to help align the IoT project organization and the organization responsible for asset design and manufacturing. Making this responsibility explicit in the organization is described as a key success factor.

The organizational setup proposed by Ignite is loosely based on a plan/build/run concept, as can be seen in Figure 5. This has drawn some criticism from the Agile community since, here, plan/build/run seems to imply a waterfall-like approach. To address this, Slama, Puhlmann et al. (2015) includes an open letter from a member of the Agile community, plus a response from the authors.

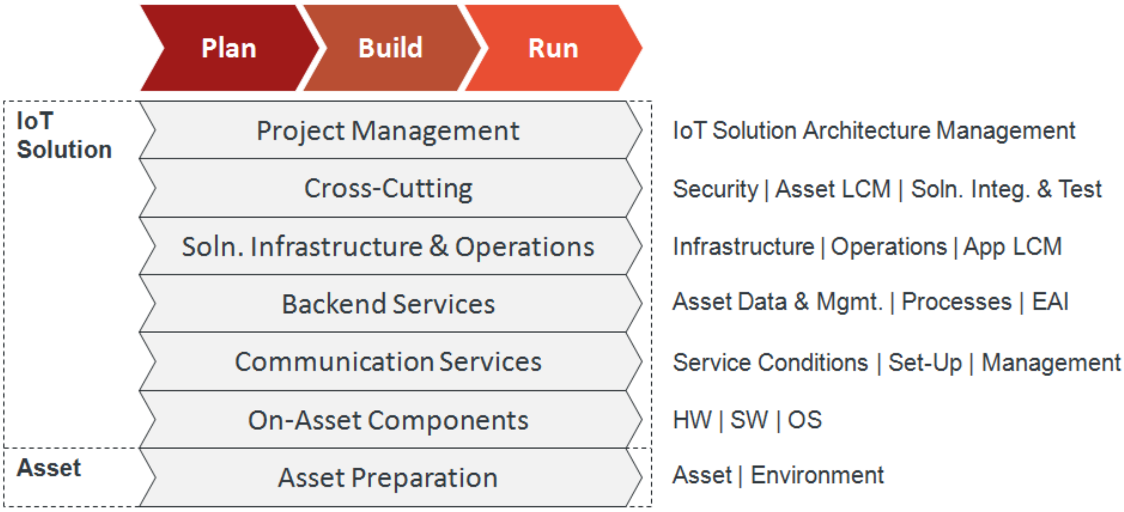

Figure 5: Ignite project organization setup (Slama, Puhlmann et al., 2015)

Furthermore, the plan/build/run approach assumed by Ignite might also need to be adapted to better fit with modern IT management paradigms as outlined, for example, in Zarnekow et al. (2006), which describes a paradigm shift “From Plan-Build-Run to Source-Make-Deliver.”

Another important element of Ignite is an IIoT-specific solution design process, which includes a framework of related design artifacts. An overview is shown in Figure 6. The Ignite IoT solution design process proposes the following phases:

- Analysis, projections, planning: This phase includes problem statement, stakeholder analysis, site survey, solution sketch, ignite project assessment, quantity structure, and milestone plan.
- Functional design: This phase includes process maps and/or use cases, UI mockups, data-centric domain model, asset integration architecture (defined by ignite), and SOA landscape (service-oriented architecture).
- Technical design: Finally, the technical design includes the detailed software architecture, technical infrastructure, and hardware design

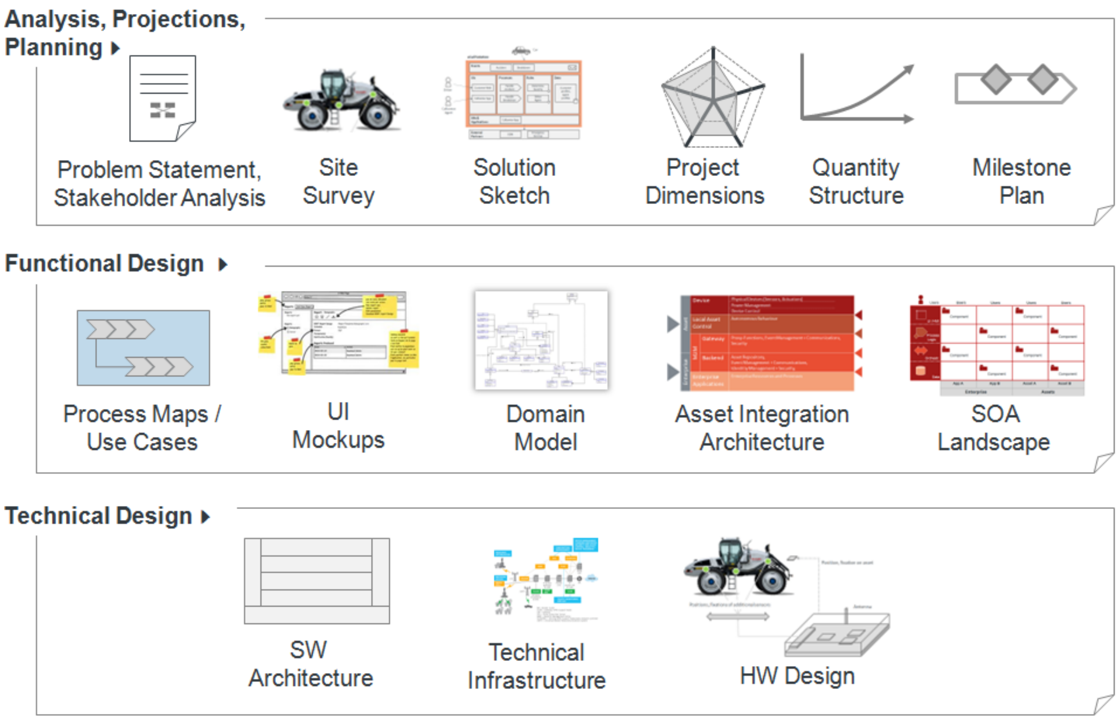

Figure 6: Ignite solution design process (Slama, Puhlmann et al., 2015)

The Ignite methodology, as outlined in Slama, Puhlmann et al. (2015), refers to IoT, or, more specifically, Enterprise IoT. At the time of writing, “IIoT” was not a well-established term. The authors used the term “enterprise IoT” to refer to IoT projects executed in the context of a large enterprise. No differentiation was made between enterprises focusing on industrial solutions and those focusing on consumer solutions. However, a key assumption of Ignite is that projects are executed at scale and in an enterprise context. Almost all of the project examples provided by enterprise IoT fall into the category of IIoT use cases, as outlined earlier. Hence, this thesis works under the assumption that Ignite can be seen as a methodology well suited to support IIoT projects.

Ignite clearly differs from manufacturing-centric frameworks such as Industry 4.0 (Lasi et al., 2014):

- Ignite makes an explicit differentiation between directly asset-related activities (referred to in Ignite as “asset preparation”) and the actual IoT solution, which interfaces with the asset. The main focus of Ignite is on the IoT solution, not on the design and manufacturing of the actual asset.
- Industry 4.0, on the other hand, focuses more on the manufacturing process, even though it also acknowledges both sides. For example, the Industry 4.0 RAMI architecture model explicitly defines two dimensions, “type” and “instance.” Type refers to the product design and manufacturing setup, while instance refers to the output of the manufacturing process (Hankel, 2015).

Because of the different focus areas (type for Industry 4.0, instance for Ignite), Ignite offers an Industry 4.0 extension to close this gap. This I4.0 extension to Ignite aims to derive requirements for the product design and manufacturing process based on the analysis of the IoT solution requirements (see section 2.4.1).

1.4.2.2 The IoT Methodology

The IoT methodology (IOTM) was introduced in 2014 by Tom Collins (Collins, 2014). The methodology suggests an IoT lifecycle based on three phases:

1. brainstorm: co-creation, ideation, validation
2. build: architecture, implementation, deployment
3. tune: identify, classify, act

For the brainstorming phase, the IoT methodology proposes the use of an IoT canvas and an adoption of the business model/Lean canvas (Link, 2016). In this IoT canvas, the things and the users are two key elements, with linking elements including end points, middleware, automation, data model, third-party services, and widgets.

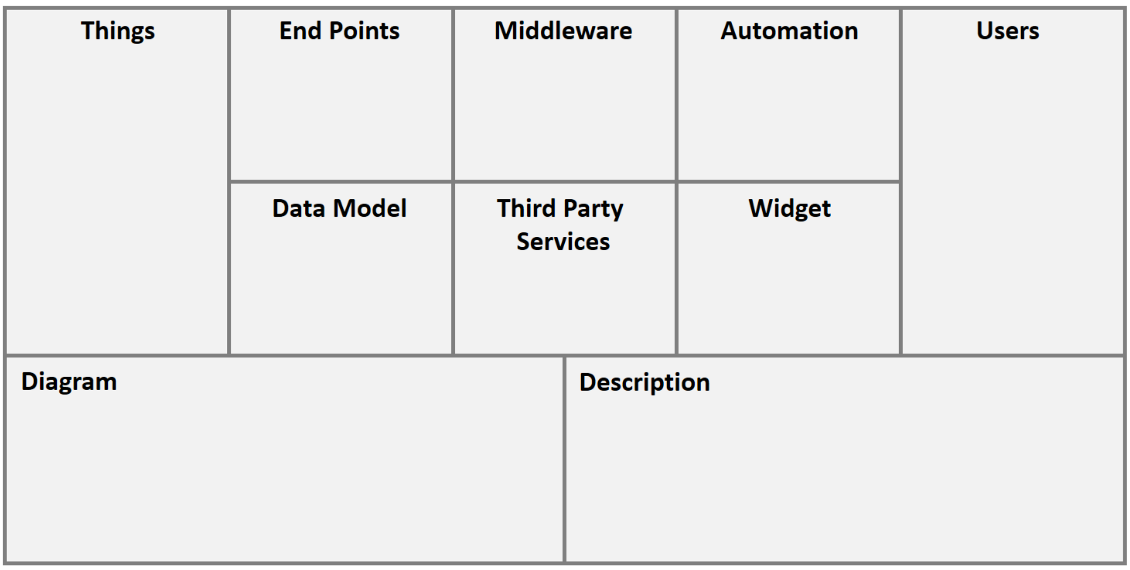

Figure 7: IoT canvas according to Collins (2014)

For the build phase, the IoT methodology proposes to create a mapping between the IoT canvas and the target architecture, including end points, connectivity, middleware, and IoT services and applications. An example is provided based on a smart home use case.

The website for the IoT methodology promotes an IOTM co-creation workshop, which is an indication of the availability of more structured training material. An IOTM toolbox is promoted, including various templates, scoring cards, and assessment tools. An active Twitter feed (@iotmethodology) provides news.

1.4.2.3 IoT Application Development

Patel and Cassou (2015) presented a paper that defines a development methodology for IoT applications. They propose to separate IoT application development into different concerns, namely, domain-specific concepts, functionality-specific concepts, platform-specific concepts, and deployment-specific concepts. They heavily emphasize automation techniques during the different phases of IoT application development, e.g., using domain-specific languages (DSLs) and code-generation techniques.

The focus seems to be on the development of the theoretical concepts. Web research has not found evidence that the concepts are further developed into a methodology designed for broader adoption. There is no organization supporting the methodology, and no trainings are offered.

1.4.2.4 ELDAMeth

Fortino and Russo (2012) presented a paper on using an agent-oriented methodology for simulation-based prototyping of distributed agent systems. The focus is on distributed agent systems (DAS) and event-driven lightweight distilled statecharts-based agents (ELDA). The ELDAMeth methodology includes three phases: modeling, simulation, and implementation. ELDAMeth is supported by ELDATool, a statechart-based tool for prototyping multiagent systems. ELDAMeth is specific to one technology and one tool. It does not support broader concepts of an IoT methodology, e.g., IoT requirement management, IoT architecture design, etc.

1.4.2.5 A Software Product Line Process to Develop Agents for the IoT

Ayala et al. (2015) have presented a paper on defining a software product line process to develop agents for the IoT. They position agents as “a good option for developing self-managed IoT systems due to their distributed nature, context-awareness and self-adaptation.”

The process defined in this paper has three main phases: domain engineering, application engineering, and finally the weaving process, which merges the outputs of the first two phases. The process is described in Figure 8.

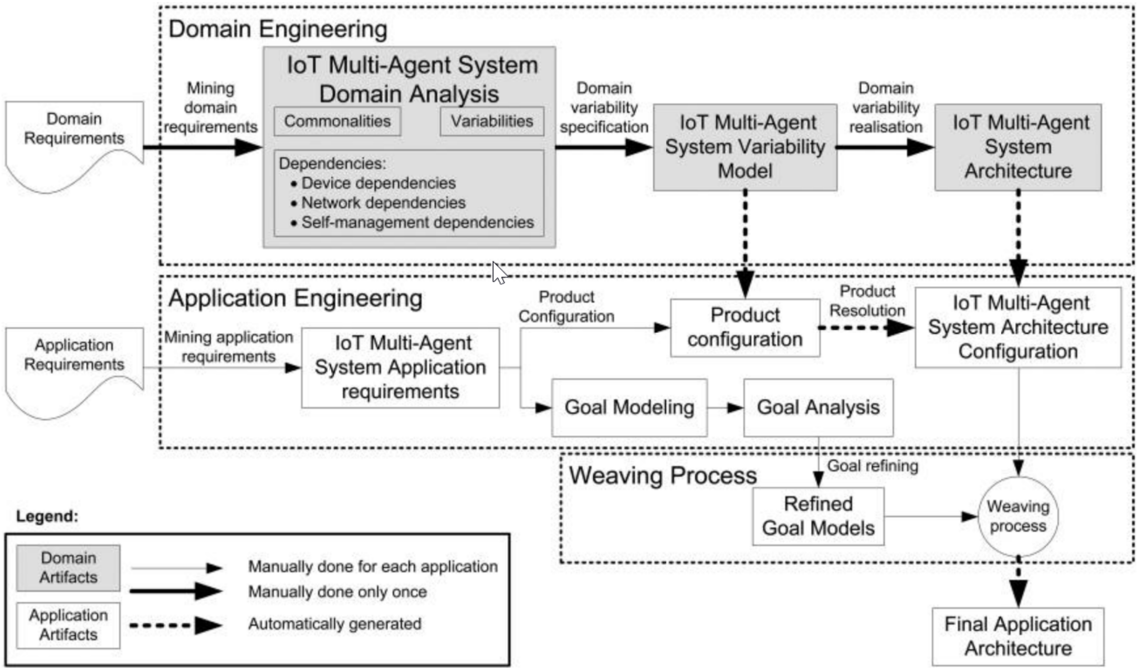

Figure 8: SPL process for Self-StarMAS agents, from Ayala et al. (2015)

The software product line process (SPL) outlined in this paper is highly specific to multiagent systems, especially to one technology (Self-StarMAS agents). Web research has not shown a dedicated organization behind this approach or any available training.

1.4.2.6 A General Software Engineering (SE) Methodology for IoT

Zambonelli (2016) has presented a paper where he outlines concepts and abstractions for IoT engineering. He suggests three main phases:

- Analysis:
  - Stakeholder and user analysis
  - Identification of main functionalities, including goals, policies, and functions
- Design:
  - Designing the orchestration and coordination between groups and coalitions
- Development:
  - Design and implementation of individual avatars
  - Smart Things and the deployment and enrichment of the infrastructure

Again, this seems to be an academic paper but not an established methodology. Web research has not shown a dedicated organization behind this approach or any available training.

1.4.2.7 Situational Method Engineering for IoT

In Giray and Tekinerdogan (2018), the authors are looking at the discipline of situational method engineering (SME) for IoT. Verrijn-Stuart and Olle (1994) provide the following definition of SME: "Situational Method Engineering aims at harmonisation of methods by providing rules to configure project-specific methods out of fragments from existing standard methods." The conceptual model for SME is summarized by Giray and Tekinerdogan (2018) in Figure 9.

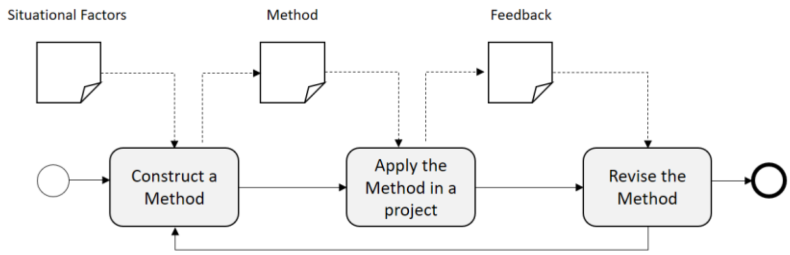

Figure 9: Conceptual model for SME according to Giray and Tekinerdogan (2018)

The situational factors identified by Giray and Tekinerdogan (2018) are:

- Business context and requirements: regulations, standards, requirements stability
- Organization: size, maturity, management commitment, structure
- Team: size, geographic distribution, domain experience, technical experience
- Customer: availability, domain experience, resistance
- System: size, reuse, technology maturity, existing IoT devices, existing backend services, degree of innovation

To map the above situational factors to a concrete IoT methodology, Giray and Tekinerdogan (2018) propose a concrete method fragment descriptor, which maps situational factors and method fragments so that eventually a tailored software development method (SDM) can be constructed to fit the needs of an individual project.

In their paper, they describe two concrete cases:

- Case 1: An IoT SDM for Small-Scale Farm Management System—a tailored methodology to develop an information system for small-scale farms, supporting the basic agricultural decision-making process
- Case 2: An IoT SDM for Large-Scale Integrated Farm Management System—a tailored methodology to develop a sophisticated, integrated farm management system

Based on the different situational factors of the two cases, two tailored IoT SDMs are created. The IoT SDM for Case 2 is shown in Figure 10. It includes business requirements and business process documentation as well as use case definitions, system architecture design, site survey, selection of IoT devices and platform, prototype development, and, finally, incremental development until the final solution is approved.

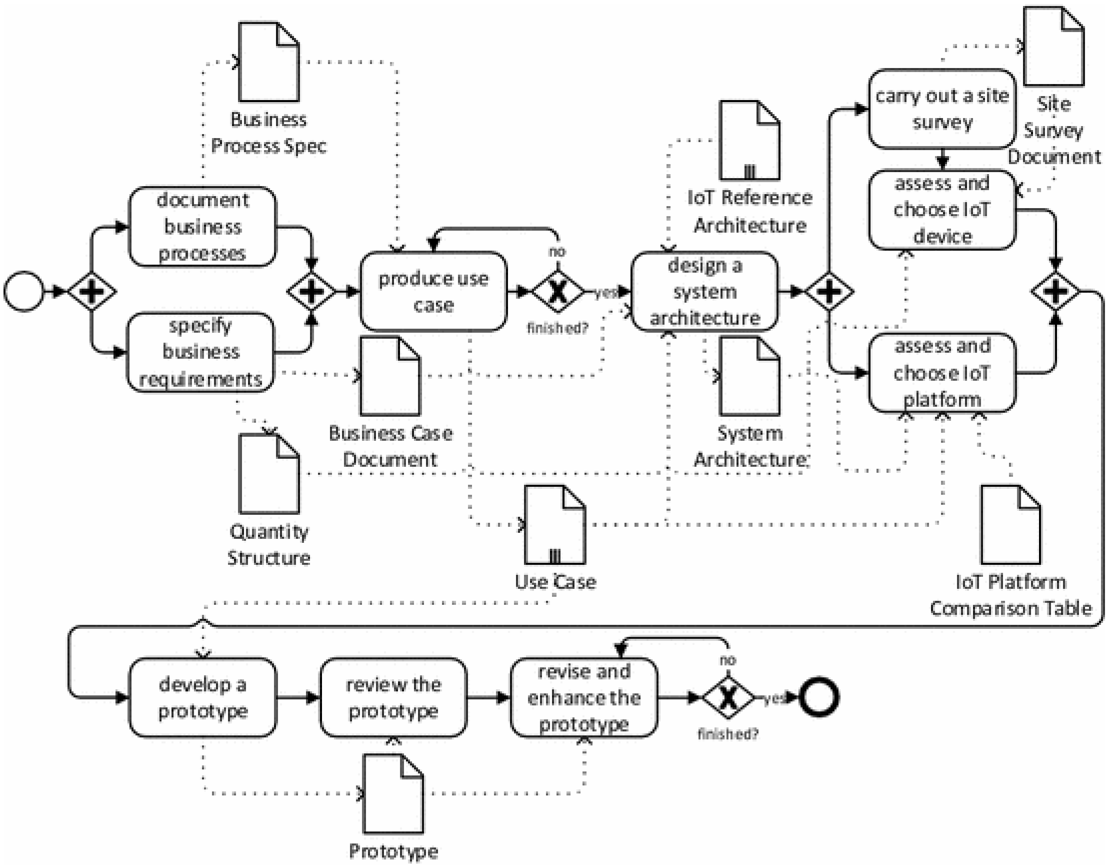

Figure 10: IoT SDM for Case 2, from Giray and Tekinerdogan (2018)

It is important to note that the work presented here is not a concrete IoT project management methodology. Rather, the proposal from Giray and Tekinerdogan (2018) is to use the situational method engineering approach to construct individual methodologies that best match the situational factors of the individual project. Also, the two presented IoT SDMs for Cases 1 and 2 are not designed as concrete SDMs that can be directly applied to real-world problems. They are provided as abstract concepts to illustrate the SDM approach for IoT.

1.4.2.8 Summary and Conclusions

The analysis of the different IoT methodologies is summarized in Table 5. Based on the metrics defined in the beginning and the findings in the individual analysis, the table includes academic foundation (e.g., number of papers providing this), online activities (e.g., number of blogs, online discussions, and articles), availability of training and support provided by at least one organization backing the methodology, and availability of tools, including paper-based tools like templates, or online tools.

As a benchmark, the table includes Project Management Professional (PMP), the professional certification for project management experts offered by the Project Management Institute (PMI). According to PMI (Project Management Institute, 2018), there were 833,025 active PMP-certified individuals. In the assessment table, PMP is assigned five stars for academic foundation since Google Scholar returns 16,900 results (Sept. 2018). Again, it is given five stars for online activities (Google returns 25,400,000 results in Sept. 2018) and five stars each for training and support, as well as tools (19,200,000 Google results in Sept. 2018).

Methodology	Academic Foundation	Online Activities	Training & Support	Tools
PMP (benchmark)	● ● ● ● ●	● ● ● ● ●	● ● ● ● ●	● ● ● ● ●
Ignite		●	●	●
IoT-Meth		●	●	●
IoT-AD	●			
ELDAMeth	●			●
SPLP-IoT	●			
GSEM-IoT	●			
SME4IoT	●			

Table 5: IoT methodology comparison

Table 5 does not provide a precise benchmark supported by a detailed statistical model. Instead, it serves to visualize the qualitative findings of the research summarized in the previous chapters. As can be seen, there are two IoT methodologies that appear to at least support the basic requirements of online activities, training and support, and tools: Ignite and the IoT methodology. Both have been ranked the weakest on academic foundation since both come from a professional background. All other methodologies identified by Giray and Tekinerdogan (2018) seem to have an academic foundation but are weak on the other points.

However, even Ignite and the IoT methodology don’t appear to be widely adopted, based on the criteria outlined earlier; the proof for online activities is too small in both cases.

1.5 Research Question, Scope, and Goals

Derived from the learnings in the initial problem analysis, the research question of this thesis is defined in Table 6.

Research Question
How can a system-supported methodology be designed and evaluated that supports project managers in finding a suitable setup for their IIoT projects?

Table 6: Research question

The scope of the central design artifact is kept deliberately focused on the actual setup of the IIoT projects to fit the realistic output of a thesis. An important assumption is that the setup of an IIoT project has a significant impact on the performance of the project as a whole (especially time to market, quality of the deliverables, and cost) since the project setup defines the shape of all project activities that follow. This assumption has been supported by research on IT projects in general (see section 2.1), and is applied here to IIoT projects as well. Consequently, the research scope in preparation for the design artifact of this thesis is defined as IIoT project management as a whole.

Derived from the overall research question, the following aspects should be examined as well in this thesis:

1. What are the main challenges of IIoT project managers today?
2. Which methodologies and frameworks already exist to help address them?
3. What actually constitutes the setup of an IIoT project?
4. How can a methodology or framework be designed that generalizes the problem and provides actionable guidance for IIoT project setup?
5. How should the ideal design of a system look that supports such a methodology?
6. How can real-world experience and feedback be incorporated back into the system to optimize the quality of the methodology over time?
7. How can the quality of the proposals developed in this thesis be improved?

This thesis has a strong focus on real-world industrial applications. It has been developed in close collaboration with the IIC German country team to ensure practical relevance. Consequently, the research is looked at from two perspectives: first, the academic perspective and, second, the industrial transfer perspective.

Academic goals of this thesis include:

- Build on state-of-the-art IIoT research, especially IIoT project management and best practices
- Generalize the results of the analysis into a holistic framework
- Improve the quality of the results based on scientific methods

Industrial transfer goals for this thesis include:

- Provide the foundation for a system design that can be implemented in an industrial context
- Provide the foundation for system functionality that will provide real benefit to industrial users, in particular IIoT project managers

1.6 Research Methodology

After having established the context of this thesis, as well as the problem and motivation, the following sections look at a suitable research methodology to address the issues outlined earlier.

This thesis examines the design and evaluation of methodologies and support systems for IIoT project managers. The chosen research methodology must be one that specifically supports the creation and evaluation of new artifacts. For this purpose, design science is examined as a suitable methodology in the following. Furthermore, to strengthen the evaluation aspect of this thesis, case study research is also considered.

1.6.1 Design Science Research Methodology for Information Systems Research

In 1991, Nunamaker Jr et al. (1990) already introduced the idea of integrating system development into the research process. Simon (1996) envisions design science as a pragmatic research paradigm that enables the creation of innovative artifacts to address real-world problems.

Six years after the initial paper from Hevner et al. (2004), Hevner and Chatterjee (2010) describe design science as "an effective means of addressing the relevancy gap that has plagued academic research, particularly in the management and information systems disciplines." They argue that natural science research methods are more "appropriate for the study of existing and emergent phenomena," while problems that require creative, novel, and innovative solutions "are more effectively addressed using the type of paradigm shift offered by design science."

Peffers et al. (2007) describe principles, practices, and procedures to carry out design-science-oriented research. The design science process described here includes six steps: problem identification and motivation, definition of the solution objectives, design and development, demonstration, evaluation, and communication. Peffers et al. (2007) also describe different possible research entry points for each of these steps, as can be seen in Figure 11. These entry points include problem-centered initiation, objective-centered initiation, design and development-centered initiation, and client/context initiated.

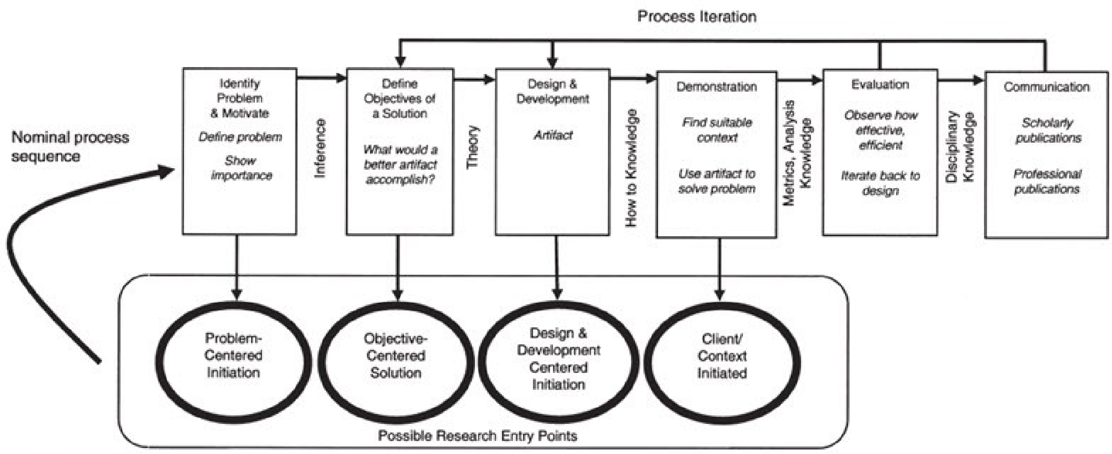

Figure 1. DSRM Process Model

Figure 11: Design science research methodology for information systems research (Peffers et al., 2007)

Peffers et al. (2012) have performed an evaluation of design sciences artifacts based on the evaluation of 148 design science research articles published in selected journals. They analyze these articles to develop taxonomies of design science artifact types, as well as methods for artifact evaluation. Based on this, they then provide two studies. The first one focuses on "instantiation evaluated by prototype," where instantiation is an artifact type, and prototype is an evaluation method type. The second example study focuses on "method evaluated by case study." They conclude, "The case study lends itself for use in evaluating the efficacy of a designed object that is intended to be used in a complex organizational setting where a simple experiment or other simple test could not be used to adequately show the efficacy or performance of the object." Consequently, the following section looks at case study research in more detail.

1.6.2 Case Study Research

Benbasat et al. (1987), Robson (2002), and Yin (2003) all provide widely cited definitions of a case study, agreeing that it is an empirical method aimed at "investigating contemporary phenomena in their context."

Robson (2002) calls it a research strategy, emphasizing the need to use multiple sources of evidence. Yin (2003) notes that the "boundary between the phenomenon and its context may be unclear."

Easterbrook et al. (2008) make an interesting differentiation between exploratory and confirmatory case studies.

Runeson and Höst (2009) define a case study research process, which includes five major process steps: case study design, preparation for data collection, collection of evidence, analysis of collected data, and, finally, reporting (see Figure 12).

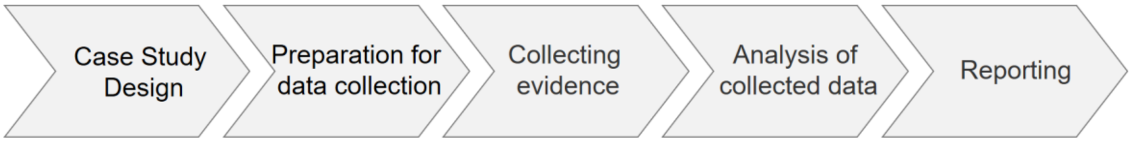

Figure 12: Case study research process

According to Runeson and Höst (2009), case study design includes the definition of the case, the case study protocol, and ethical considerations. The case study design must clearly define the case and its unit of analysis. Case study objectives, hypotheses, preliminary research questions, and theoretical bases should be clearly defined. The case should be adequately defined in terms of size, domain, process, and subjects. Considerations include: Are data triangulation (multiple sources) and method triangulation (multiple methods) reflected in the design? How clear is the rationale behind the selection of subjects, roles, artifacts, and viewpoints?

After the initial case study design, Runeson and Höst (2009) describe the process of data collection, including first-, second-, and third-degree data sources, interviews, observations, archival data, metrics, and checklists.

In data analysis, Runeson and Höst (2009) differentiate between quantitative data analysis and qualitative data analysis. For qualitative data, they differentiate between hypothesis-generating techniques and hypothesis-confirmation techniques, based on Seaman (1999). Hypothesis-generating techniques can be used for exploratory case studies, while hypothesis-confirmation techniques are used for explanatory case studies. Triangulation and replication are cited as examples of hypothesis-confirmation methods, based on Seaman (1999). Another important factor is the validity of a study since it directly related to the trustworthiness of the results.

Finally, the reporting communicates the findings of the study. Jedlitschka and Pfahl (2005) propose guidelines for reporting of experiments that have been evaluated by Kitchenham et al. (2008). An important point in the proposed reporting standards is cross-study comparisons through systematic reviews. Since case studies are usually based on qualitative data, the level of standardization here is lower.

1.6.3 Approach Taken by This Thesis

Given the positive evaluation of both design science (for methodology design and evaluation) and case study research (for evaluation specifically), this thesis combines both approaches. For design science, the approach from Peffers et al. (2007) was chosen. For case study research, the process outlined in Runeson and Höst (2009) was chosen. The following analysis shows that in the context of the goals of this thesis, the two provide a good foundation.

Starting with the process outlined in Peffers et al. (2007), the required iterations and phases for this research project were mapped to it (see Figure 13).

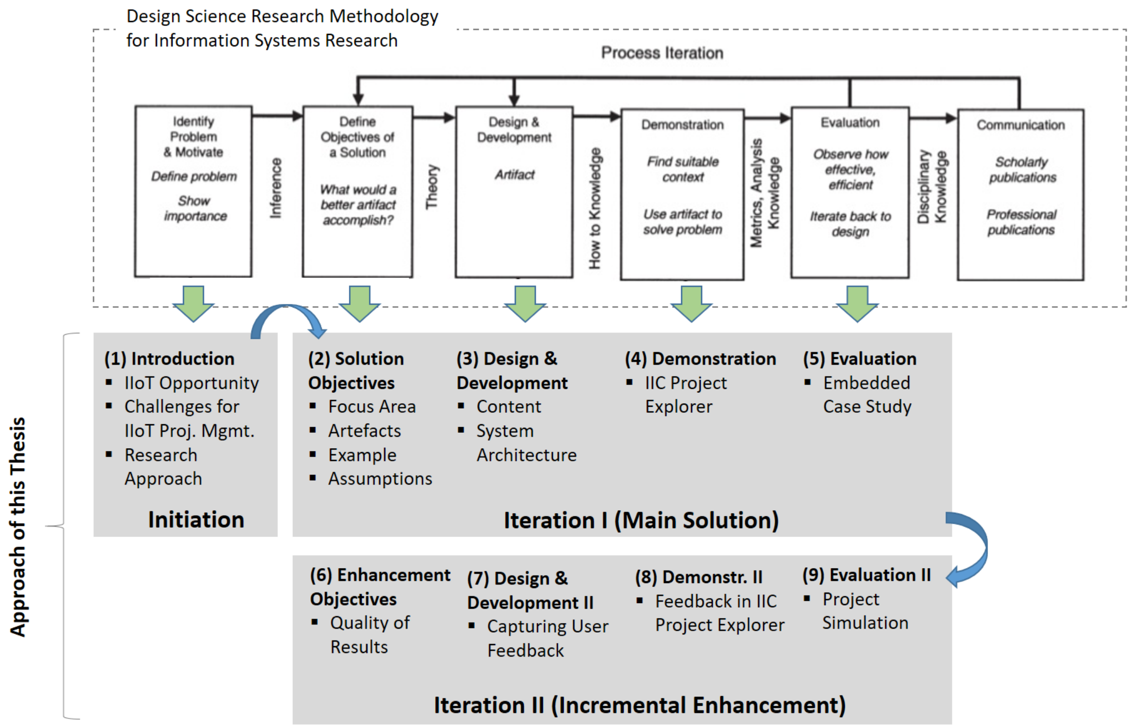

Figure 13: Research methodology and approach taken by this thesis

This research project includes the initial process iteration, plus a second, incremental iteration added to incorporate results from the first iteration and to provide a deep dive on selected topics that emerged as important in the first iteration.

Phase one of this research project is the initiation phase, looking into the problem definition and motivation. The matching section of this thesis provides an overview of IIoT challenges and opportunities, as well as a specific view on the challenges faced by project managers in the IIoT. It also contains an overview of the chosen research methodology and structure of this thesis.

Iteration one, phase two looks at solution objectives. This includes a definition of the focus areas of this project, reuse of existing artifacts, a concrete example, and assumptions and constraints.

Iteration one, phase three considers the concrete design and development of the solution, which has been named IgniteWorx. The design and development of IgniteWorx has two main aspects: design and development of concrete content for the system, as well as design and development of the supporting IT solution.

It should be noted that while iteration one, phase four (demonstration) looks at a concrete reference implementation based on the concepts developed in this thesis, this implementation is actually *not* in scope of the thesis itself. Rather, it is an implementation done by the IIC based on the open-sourced design concepts created as part of this thesis (see copyright overview, page 44).

Iteration one, phase five is an evaluation based on an embedded case study (Yin, 2003). In this case study, four different projects are evaluated against the IgniteWorx framework defined in this thesis.

Iteration two performs an incremental enhancement of the results of iteration one, focusing specifically on the issue of capturing user feedback. This includes a discussion of the objectives of this incremental enhancement (6), the design and development (7), the demonstration (8), and finally a simulation-based evaluation of the enhancements (9).

Finally, the thesis ends with a general evaluation of the process and the key findings, as well as an outlook into potential future work.

1.7 Related Work

Building on the work by Peffers et al. (2007), this thesis follows the design science research methodology, describing a full, initial iteration as well as an incremental second iteration. In the first, full iteration, the following related work has been especially important:

- IoT/IIoT Foundation and Delimitation: IIC (2018d) and Lin et al. (2015) for the IIoT perspective. Slama, Puhlmann et al. (2015) for IoT in the enterprise context. Lasi et al. (2014) for the Industry 4.0 perspective.
- IoT Methodologies: Giray and Tekinerdogan (2018) for defining the scope of the research done on this topic in this thesis. Wysocki (2013) and Yardley (2002) on project priorities. Jacobson et al. (2017) on a unifying view. Slama, Puhlmann et al. (2015) for the plan-build-run perspective on IoT projects. Zarnekow et al. (2006) for the paradigm shift "From Plan-Build-Run to Source-Make-Deliver."
- Project Management Methodologies: Grau (2013) for defining the scope of the research done on this topic. Wang and Gibson Jr (2010) on the importance of pre-project planning. Rausch et al. (2005), PMI (2017), DOT (2007) for general background. Mahalakshmi and Sundararajan (2013) for a discussion on traditional versus agile approaches. Wells (2012) for considerations on effectiveness. Joslin and Müller (2014) for impact analysis.
- IgniteWorx defines ten key areas to be considered for IoT project setup (so-called result sets). For these ten areas, the following research was considered:
  - Project Management Methodology (A): Similar to the above, again following Grau (2013) for defining the scope.

- Solution Design (B): Reyes-Delgado et al. (2016) for the broad perspective. Alwadain et al. (2013) for comparative analysis. Krafzig et al. (2005) for details on SOA-driven solution design.
- Technology Selection (C): Bouwers et al. (2009) for evaluation criteria. Chan et al. (2000) for evaluation methodologies. Durrani et al. (1998) for managing the technology acquisition process.
- Resource Acquisition (D): Fowler et al. (2019) for build versus buy. Hughes (2018) for on-premise versus cloud. Miller (2010) for vendor selection best practices.
- Cost Estimation (E): Zarnekow and Brenner (2005) for the TCO perspective. Boehm et al. (1995) for the historical perspective. Giannopoulos (2006), Menzies et al. (2014), and Jones (2007) for best practices and general software design and implementation cost estimation considerations. Kalmar and Kertesz (2017) for IoT cloud cost estimation. Dash and Acharya (2011) for cost estimation in distributed systems. Debardelaben et al. (1997) for cost estimation in embedded systems.
- Risk Management (F): Boehm (1991) for the historical perspective. Chapman (1997) for the origins for project risk analysis and management. Yeo and Ren (2009) for risk management maturity analysis. Latifi and Zarrabi (2017) for COBIT 5 and IoT.
- Trust and Security (G): Susanto et al. (2011) for information security management system standard. ENISA (2017) for baseline security recommendations for IoT. IIC (2018c) for IIoT security maturity modeling. Online Trust Alliance (2018) for the broader trust perspective.
- Reliability and Resilience (H): Axelrod (2009) for a general investigation of software resilience. Delic (2016) on the resilience of IoT systems. Microsoft Trustworthy Computing group (2013) for methodological support.
- Verification and Validation (I): Herrmann (2001) provides a general guideline for Verification and Validation (V+V). MITRE (2014) looks at V+V in its system engineering guidelines. Jaikamal (2009) looks at model-based support for V+V.
- Service Operations (J): AXELOS (2019) and ITIL as the foundation for ITSM. BMC (2016) for ITIL best practices. Miklovic (2015) for asset performance management. (Wong) for field service management.

▪ IgniteWorx Architecture: Architecture selection criteria for IgniteWorx follow the ATAM approach, described by Clements et al. (2002). Lasi (2012) and Holsapple (2000) for decision support systems. Ogu and Y.A. (2013) for expert systems. Ricci et al. (2015b) for recommender systems. Berners-Lee et al. (2001) and Blomqvist (2014) for semantic reasoning systems. Fuerst (2001) and Schlereth and Skiera (2012) for survey engines. Saaty and Saaty (2000) for analytic hierarchy process. Poole and Mackworth (2017) and Neapolitan and Jiang (2018) for artificial intelligence. Yatsalo et al. (2015) for multi-criteria decision analysis. Wątróbski and Jankowski (2015) for hybrid architectures.

- Evaluation: A key element of the first evaluation is the case study, which follows the case study design principles outline by Yin (2003) as well as Runeson and Höst (2009).

In the second, incremental iteration, the following related work has been especially important for capturing and evaluating end-user feedback:

- Law and Abrahão (2014) provide a discussion on how evaluation feedback shapes software development.
- Law et al. (2014) look at quantitative measurements for UX improvements.
- Jan et al. (2016) provide an innovative approach to investigate various software-testing techniques and strategies.
- Alkhalid and Labiche (2016) and Alkhalid (2018) offer important insights regarding functional tests.

1.8 Related Industry Organizations

An important context for this research is the work done in the Industrial Internet Consortium (IIC), introduced here. Finally, the Research Group of the German Country Team of the IIC is described, which also helped to advance the research done as part of this thesis.

1.8.1 Industrial Internet Consortium

A significant amount of the research for this thesis has been conducted in the context of, or in collaboration with, the IIC. According to Wikipedia (2018), the IIC is "an open membership organization, with 258 members as of 22 November 2016. The IIC was formed to accelerate the development, adoption and widespread use of interconnected machines and devices and intelligent analytics. Founded by AT&T, Cisco, General Electric, IBM, and Intel in March 2014, the IIC catalyzes and coordinates the priorities and enabling technologies of the Industrial Internet."

The IIC has published a number of documents about the IIoT, including the Industrial Internet Reference Architecture (IIRA) (Bleakley et al., 2015). IIC also runs an extensive testbed program, with thirty active testbeds listed on its website (IIC, 2018b). The testbeds are from various industries, including manufacturing, energy, transportation and logistics, and automotive. According to the analysis of the term IIoT as defined in section 1.2, these testbeds qualify as industrial IoT testbeds if looking at the supported industry segments.

1.8.2 IIoT Research Group

The IIC runs regional teams that work locally on IIC topics. The IIC German Regional Team was the first active country team and has initiated an IIoT research group (Ferdinand-Steinbeis-Institut, 2018). The research scope of this group is summarized in Figure 14.

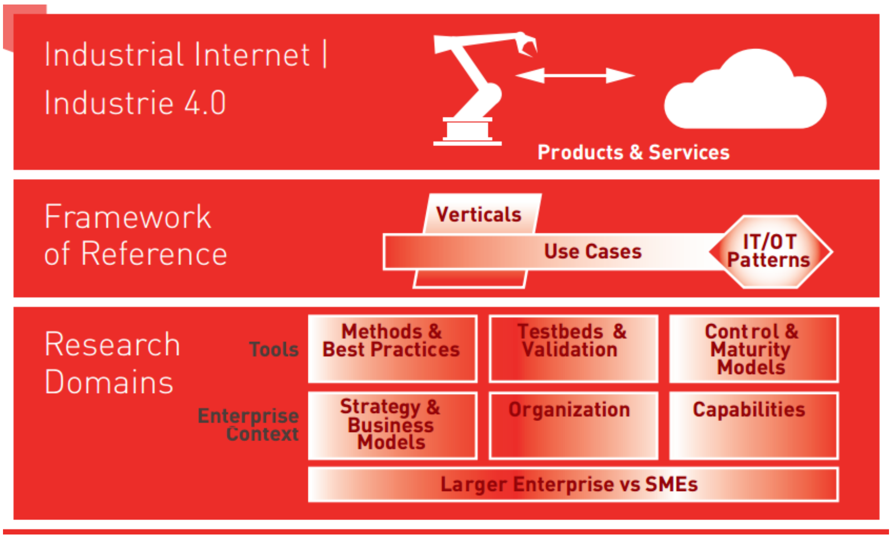

Figure 14: Overview of IIoT research group, German Country Team IIC

The framework of reference for the research group is defined as the intersection of industry verticals, use cases, and supporting IT/OT architecture patterns. The research domain includes methods and best practices for the IIoT (such as this work), IIoT testbeds and validations, control and maturity models for the IIoT, strategy and business models for the IIoT, IIoT organizations, and IIoT capabilities.

The research group also focuses on so-called transfer projects, which are designed to ensure the transfer of knowhow and best practices from large enterprises to small and medium-size enterprises (SMEs). To ensure efficient transfer models, the group has developed the concept of micro testbeds based on the IIC testbed model but with a specific focus on SMEs.

The concepts developed as part of this thesis were greatly enhanced and validated through the interactions and collaborations within this group.

1.9 Copyrights

This thesis is based on the Ignite project management methodology. Ignite is open source, based on the Creative Commons Attribution 3.0 Unported License (CC BY 3.0): https://creativecommons.org/licenses/by/3.0/. The Ignite project can be found at https://projects.eclipse.org/projects/iot.ignite.

The IgniteWorx framework defined as the key artifact of this thesis has been contributed to the Ignite methodology, again under CC BY 3.0.

Ignite is used by IIC's IIoT Project Toolkit (IPT), including the IPT online version. IIC has permitted the examination of two simulated projects from the IPT online version. This is done in Iteration II of this thesis.

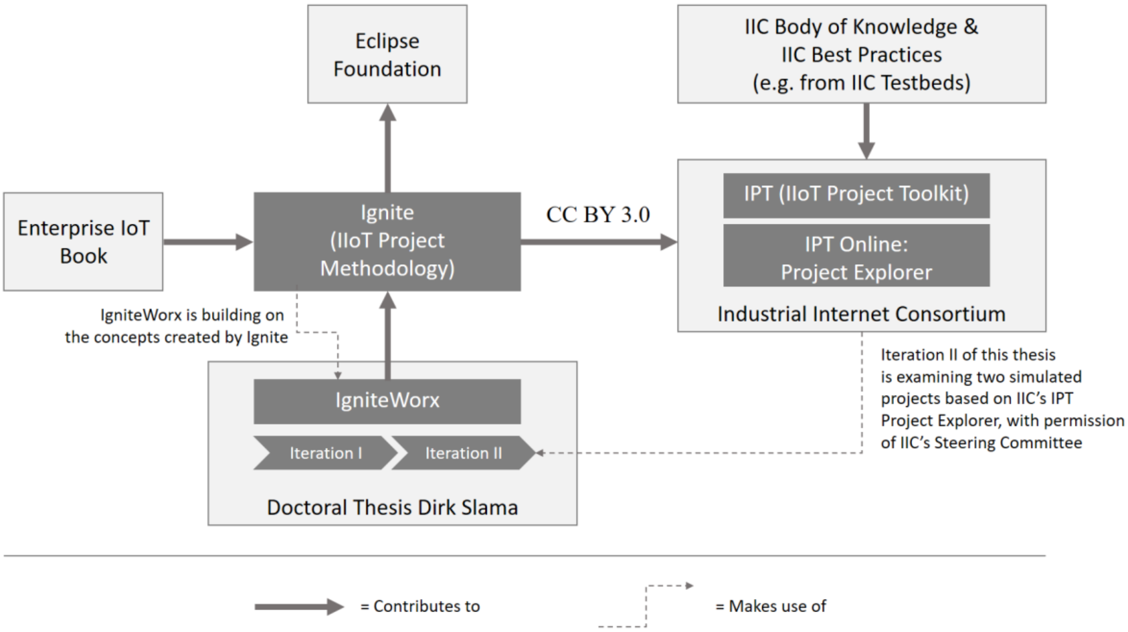

Figure 15: Dependencies and licensing of related artifacts

Also, the following icons have been sourced for this thesis from https://www.flaticon.com via Creative Commons BY 3.0: robotic arm, car parts, people, camera, cloud, artificial intelligence, traffic light, puzzle piece, jigsaw, market, CPU, router, flats, settings gears, and user silhouette.

2 Iteration I: Solution Objectives

To define the solution objectives, this section first looks at the solution focus, followed by an overview of reusable artifacts to support the objectives, requirements for the new artifact, an example, and concrete constraints.

2.1 Support for IIoT Project Setup Phase

This thesis assumes that finding the right project setup has a significant impact on the IIoT project's success. This starts with the project plan but does not end there.

Wang and Gibson Jr (2010) provide a study of pre-project planning and project success, concluding that "how well preproject planning is conducted has a great impact on project outcome." Dvir et al. (2003) offer an empirical analysis of the relationship between project planning and project success, concluding, "No effort should be spared in the initial stage of a project to properly define the project goals and its deliverables requirements." Serrador (2012) looks at the importance of the planning phase to project success, stating, "A summary of the available studies shows unexpectedly consistent empirical results for the correlation of planning and success."

In addition to the definition of the project plan, a number of other success factors can be identified in the early phase of a project, which will also be subsumed under the concept of a project setup as it is used in this thesis.

Baker et al. (1997) identifies "adequate project team capability" as another project success factor. Since the project team onboarding is usually done in the early phase of a project, it is also subsumed under project setup. Team capabilities are not only limited to individual skills and capabilities. Thomas et al. (2008) looks at the development of an effective project by combining planning and team building. The project team organization also plays an important role because the way in which organizations are structured has a strong impact on any systems they create, according to Conway`s Law (Brooks, 1975). Consequently, this should also be considered under the project setup.

Westerveld (2003) identifies the impact that contractors have on the success of a project. Again, choice of contracting partners is typically done early in the project phase, so it is considered part of the project setup. This is a broader topic, which is also related to the decision for in-house implementation resources versus outsourcing to an external systems integrator. Another key factor that has a potentially huge impact on the project in this context is the make-versus-buy decision (Gassmann et al., 2018): how much of the solution should be custom built versus how much should be based on standard components.

As can be seen, there are a number of activities and decisions in the early project phase that shape the project and determine its outcome. These are collectively referred to in this thesis as project setup.

Again, the main objective of this thesis is to develop a system-supported methodology (i.e., a new artifact) that supports project managers in finding a suitable setup for their IIoT projects.

2.2 New Artifact: IgniteWorx

For the scope of this thesis, it is assumed that the problem to be solved is to help improve the setup phase of IIoT projects. The previous examples provide a rough guideline on what can be subsumed under the concept of a project setup; these include activities and decisions in the early project phase, which are shaping the project in a significant way. In the following sections, these general concepts must also be specifically applied to IIoT.

Furthermore, since this thesis uses design science research methodology as the foundation with the main goal of creating a new artifact, this artifact needs a name. The name chosen is IgniteWorx, whereby "Ignite" refers to the original Ignite methodology, while "Worx" indicates that this thesis considers a concrete system to support the methodology.

2.3 IgniteWorx Focus

The scope of a holistic IIoT methodology must be fairly broad, from initial business model design to implementation, operations, and optimization of the solution. This is also supported by the Ignite methodology. IgniteWorx takes a more focused approach where the emphasis is on the project setup phase, which usually comes after the initial business model design and proof-of-concept (PoC) phase and before the actual project ramp-up phase (see Figure 16).

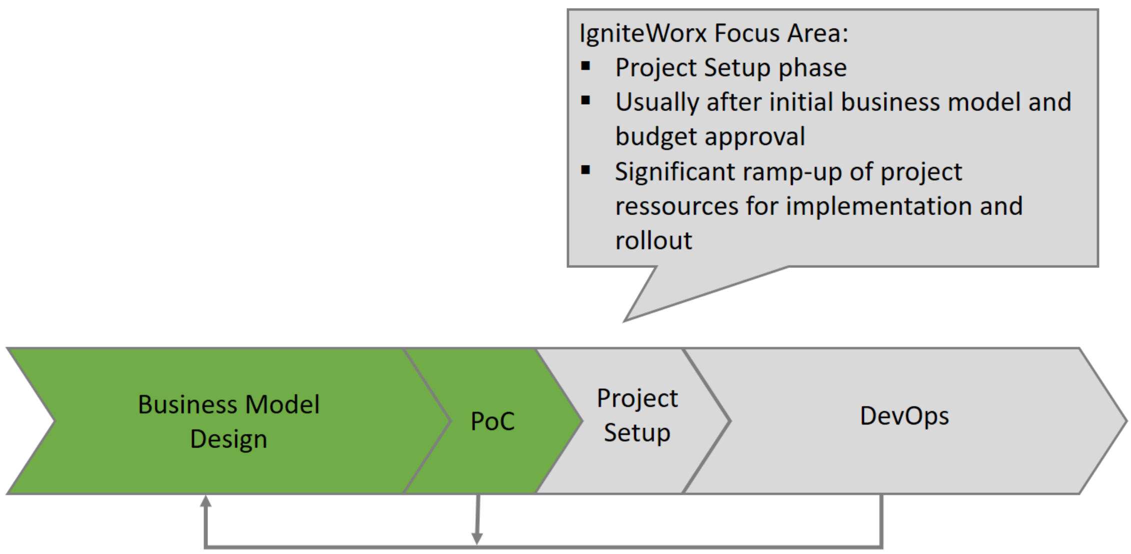

Figure 16: IgniteWorx Focus Area

The business model design and PoC phase of a project can be carried out with a relatively small team, according to Casadesus-Masanell and Ricart (2011). Once the business model design has been stabilized, the actual IIoT implementation project starts. Team sizes for IIoT projects vary depending on the actual requirements. However, since IIoT projects by definition have an industrial context, they usually have to deal with complex technical environments, scalability and security, challenging regulatory requirements, etc. The project setup phase ensures that the implementation project takes the necessary shape to deliver on the requirements defined in the business model design. Any mistakes made during the project setup phase are likely to have a ripple effect through the implementation phase, with potentially huge costs. This is why IgniteWorx focuses on this critical phase of an IIoT project.

Figure 17 provides a slightly exaggerated view of a typical project situation: A business development team has been working with various management stakeholders to convince them about an IIoT solution. The results are some presentation slides and a high-level project proposal, including budget figures. After approval, this information is the foundation for the now officially founded project team, which is tasked with implementing the solution and building up the required operations infrastructure and team.

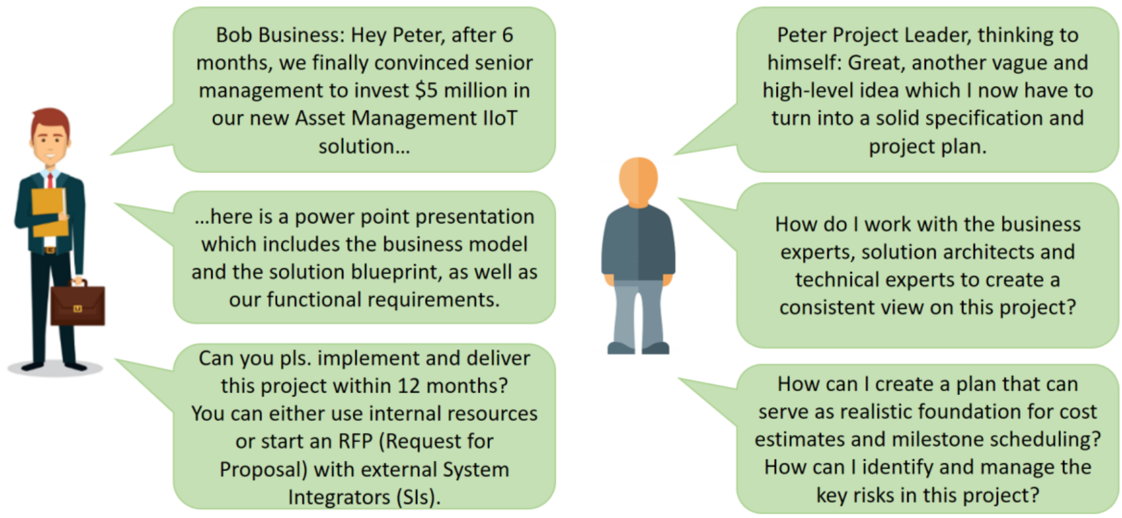

Figure 17: Typical scenario (IIC, 2018b)

The main focus of IgniteWorx as it is defined in this thesis is to support the project manager in this kind of situation. IgniteWorx should build on, more or less, proven existing artifacts and provide IIoT project managers with recommendations for the ideal setup of their IIoT projects. Ideally, the output of IgniteWorx can be used as the foundation for creating a detailed project plan, or, alternatively, an RFP (request for proposal) document for the solution tender process.

2.4 Reuse of Existing Artifacts

A key objective of the IgniteWorx project is to reuse existing artifacts where it is applicable. In the following, two main artifacts with high reuse potential are described: the ignite project assessment tool and the IIC IPT framework. Both are specifically designed to support in the setup phase of an IIoT project, as per the definition provided in 2.1.

2.4.1 Ignite Project Assessment Tool

The Ignite project assessment tool (IPAT) was developed as part of the Ignite methodology (Slama, Puhlmann et al., 2015). IPAT supports a detailed assessment of an IIoT project based on different dimensions grouped into five main areas:

- assets and devices
- communication and connectivity
- backend services
- standards and regulatory compliance
- project environment

These areas cover the most relevant parts of a typical IIoT project assessment. The dimensions of the assets and devices consider general aspects (e.g., number of assets and value of individual asset), required processing power (e.g., for local business logic and event processing), other hardware requirements (e.g., power supply and environment), and, finally, lifecycle management (e.g., project lifetime of the assets in the field). Ignite specifically looks at the requirements for assets deployed in the field and not only sensors and devices.

The next area is communication and connectivity, which is required to connect assets in the field with a backend solution. The dimensions in this area cover local communication and connectivity (e.g., specific technologies as well as bandwidth and latency) and remote communication and connectivity (from a similar perspective).

The backend services area analyses the general application strategy, as well as data management and analytics (including data volumes, variety, and velocity).

The standards and regulatory compliance area includes dimensions for region-specific, industry-specific, and technology-specific standards and regulatory requirements.

Finally, the project environment includes dimensions for project timeline, budget, and technical and functional skills.

Version 1.0 of Ignite IPAT has approximately fifty dimensions. Figure 18 provides an overview of the intended use cases for the Ignite IPAT toolkit. They include project self-assessments, project comparison, and decision-making in projects (e.g., technology selection).

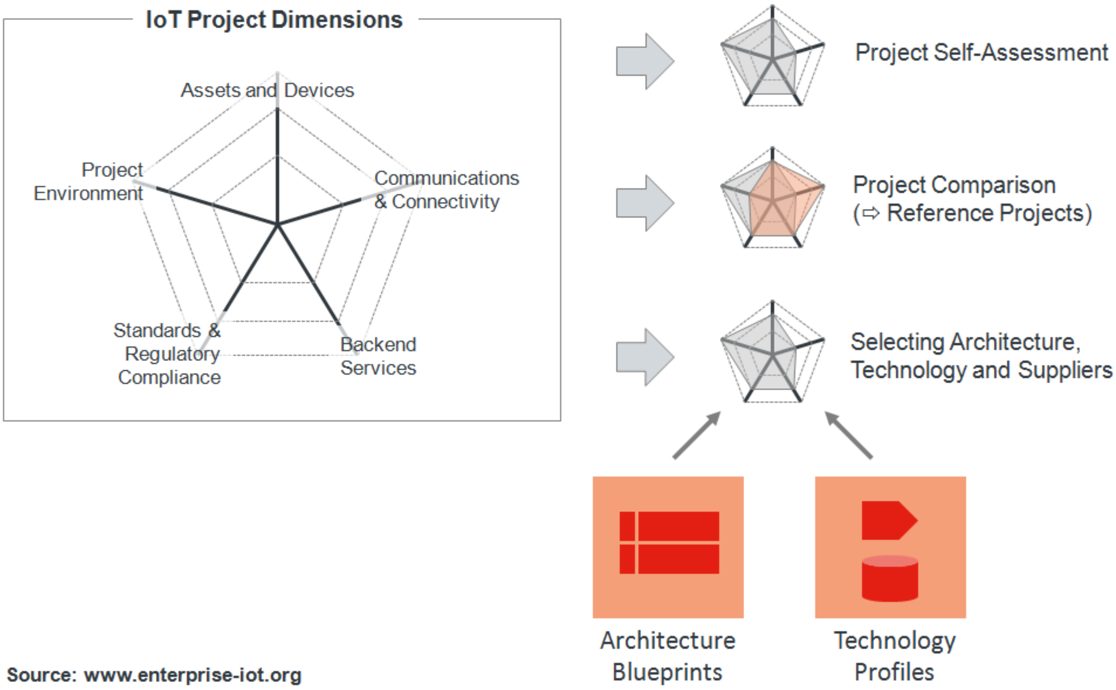

Figure 18: Using the Ignite project assessment tool

2.4.2 IIC's IPT Framework Based on Ignite

The IIC has adopted Ignite IPAT in its own IIoT project toolkit (IPT) by licensing it through the Creative Commons Attribution 3.0 Unported License (CC BY 3.0).

IIC's IPT also focuses on the IIoT project setup phase. As described in Figure 19, IPT is positioned after the business model phase and before the detailed architecture design and planning. In this case, it is assumed that the latter will be based on IIC's industrial Internet reference architecture (IIRA).

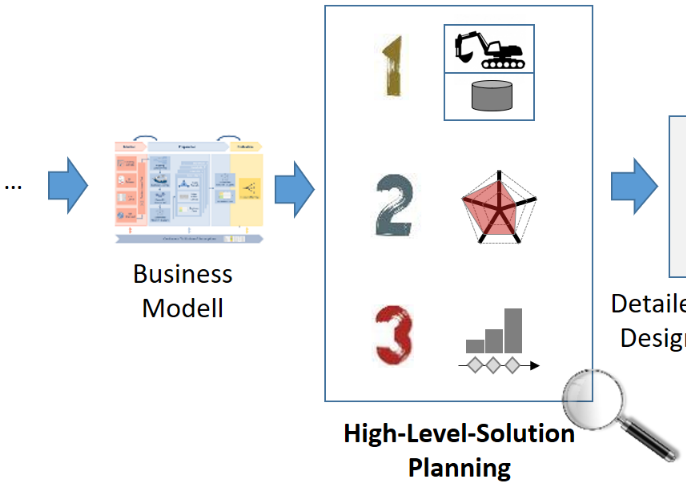

Figure 19: IIC IPT overview

IIC's IPT contains three main elements: solution sketch, self-assessment, and milestone plan. The solution sketch is a tool used to create a first, high-level solution design using a standardized solution canvas as well as icons for typical solution elements (assets, sensors, connectivity, etc.). The self-assessment tool is based on Ignite IPAT and allows for a structured project assessment following the same principles. Finally, the last element supports creation of a high-level milestone plan based on the results of the solution sketch and self-assessment.

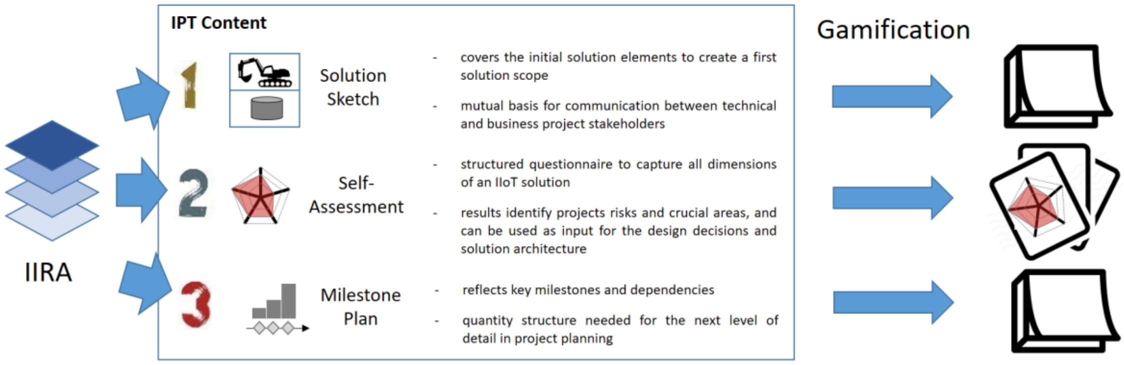

Figure 20: IIC IPT elements

IIC's IPT uses concepts of gamification (Deterding et al., 2011) to make the toolkit usable for project managers in on-site workshops. It was first productized in 2016 and provided to all members of IIC and Platform Industrie 4.0 who attended the joint IIC/Platform I4.0 meeting in Walldorf that year. Figure 21 shows a picture of IIC's IPT.

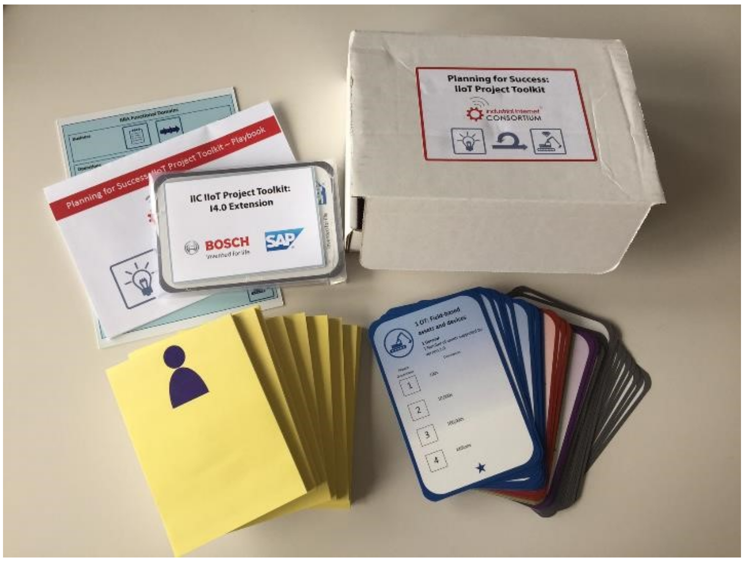

Figure 21: Gamified IPT toolkit from IIC

2.5 IgniteWorx: Requirements for New Artifacts

IgniteWorx is the new artifact created in this thesis and defines a system-supported methodology to support project managers in the creation of an IIoT project setup. This setup should contain recommendations for all relevant aspects of the target solution. The exact definition of an IIoT project setup is given in the design and development phase of this thesis (see section 3.1.2). It should follow the guidelines outlined in 2.1.

To create the recommendations for the IIoT project setup, the IgniteWorx system should combine the existing Ignite dimensions for project assessments with an IIoT knowledge base. One example of such a knowledge base is the body of knowledge created by the IIC. Together, the individual project assessment plus the IIoT knowledge base should provide concrete recommendations for the ideal setup of the IIoT project. To achieve this, IgniteWorx must intelligently match project assessment results with entities from the knowledge base to make meaningful recommendations.

Figure 22: Requirements for new artifact—IgniteWorx

2.6 Example

A concrete example for an approach based on a similar idea is COCOMO (constructive cost model) or COCOMO II (Boehm et al., 1995), especially if combined with function point analysis (Heemstra and Kusters, 1991).

The basic idea of COCOMO is to create a mapping between different software metrics and the estimated costs of the project. Because basic COCOMO is using lines of code of mainly procedural programming languages, other approaches such as weighted micro function points (WMFP) introduce metrics more suitable to modern software projects (ProjectCodeMeter).

COCOMO 81 uses a database with 61 projects, and COCOMO II uses a database with 163 projects. An assessment of the current project (e.g., using function point analysis with a mapping to resulting lines of code) can now be compared to the historic data in the reference database to derive an estimate for the development cost of the new project. Figure 23 provides an overview of the approach.

Figure 23: COCOMO + FPA for software project cost estimation

Because of the code-centric approach taken by COCOMO and similar models, this approach has also been criticized (Mukhopadhyay et al., 1992). However, methodically, it is a good example to motivate the general idea of performing a structured project assessment in combination with a knowledge base as the foundation for project-specific recommendations.

2.7 Constraints and Assumed System Evolution

Before looking at the system design, it is important to get a better understanding of some of the main constraints IgniteWorx will have to address. These constraints are expected to be found in two main areas:

- Reference Data: the availability of real-world project data
- Validation: the availability of end-user feedback on the quality of the recommendations

In the initial phase of the system evolution, neither project reference data nor end-user validation data will be available.

One key issue regarding the reference data is that this data is potentially highly proprietary: large enterprises are unlikely to make such detailed data about their IIoT projects publically available, especially not if this data could be used to derive conclusions about project cost estimates, project risks, etc. This is an issue which an IgniteWorx design will have to address.

A second issue is that if IgniteWorx is focusing on the project initiation phase, the user feedback will be most likely from users who are exactly in this phase of an IIoT project. This means, that the users can not provide feedback from the perspective of a completed project, which obviously would be more valuable.

These are two key issues which the IgniteWorx system design will have to take into consideration. Figure 24 provides an overview of the expected system evolution and the user data that should be available during these phases.

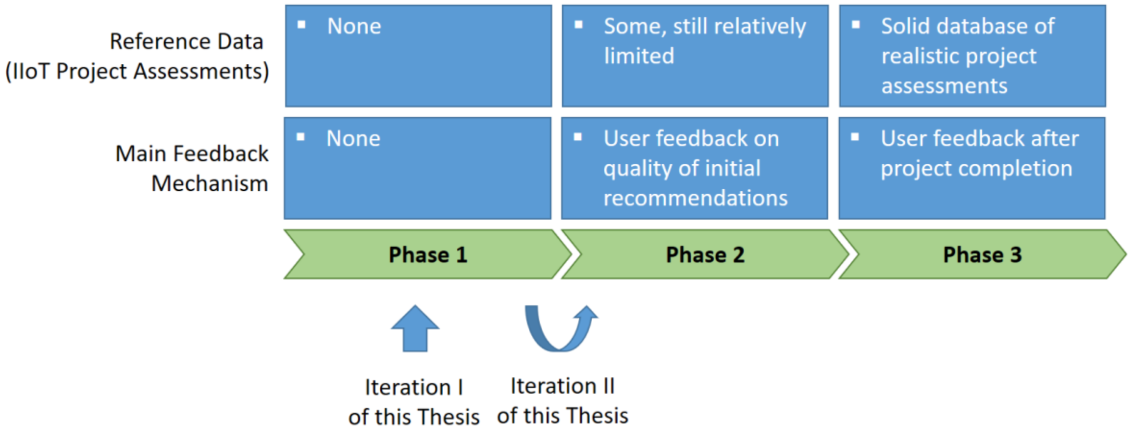

Figure 24: Expected project phases

From a system design point of view, a key question is if the system can be self-learning at all or whether an explicit knowledge modeling approach will be required (at least during phase 1 and probably also phase 2).

Iteration I of this thesis mainly supports phase 1 of the expected system evolution. Iteration II of this thesis supports the transition to phase 2 by introducing a concept for user feedback management.

3 Design and Development I

The design and development phase of iteration I of this thesis focuses on two key aspects: content and system architecture. The content section emphasizes the key content inputs and outputs of the IgniteWorx solution. The system architecture section makes a concrete proposal for how a system architecture could support the management of the inputs and automatic creation of the outputs described in the content section. It should be noted that the actual technical implementation is not in the scope of this thesis.

3.1 Content

This thesis proposes three main content artifacts for the IgniteWorx system: Ignite dimensions, IgniteWorx result sets, and IgniteWorx rules. Figure 25 provides an overview. The Ignite dimensions (1) have been developed as part of the Ignite methodology and should be reused by IgniteWorx to serve as the data for guiding the user through the structured project assessment. IgniteWorx then proposes to add result sets (2), each of which should be used to cluster related results, from which one or more should be chosen by the system based on the answers given in the project assessment. The selection of the results from the different result sets are based on the IgniteWorx rules (3), which create a mapping between Ignite dimensions and individual results in the sets. The following sections provide a detailed description of these artifacts as well as an explanation of how they were derived.

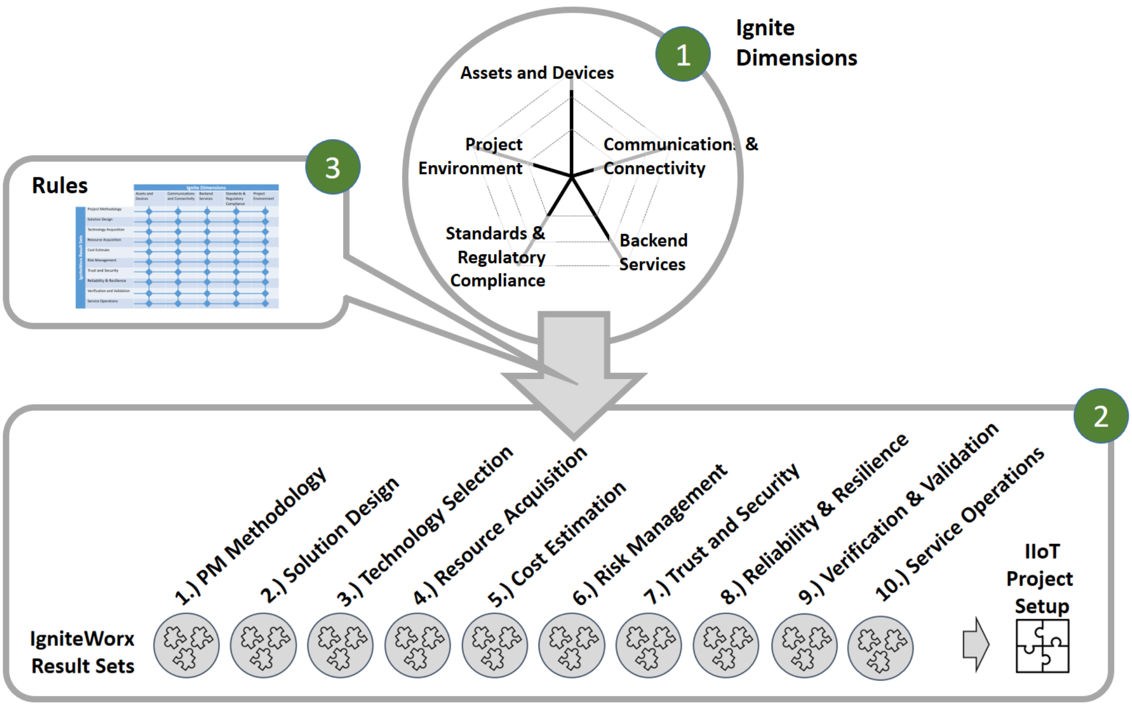

Figure 25: IgniteWorx content structure

3.1.1 Ignite Project Dimensions

As mentioned in section 1.4.2.1, the Ignite methodology (Slama, Puhlmann et al., 2015) includes project dimensions developed as part of the project assessment tool (IPAT). IPAT supports a detailed assessment of an IIoT project based on these dimensions, grouped into five main areas: assets and devices, communication and connectivity, backend services, standards and regulatory compliance, and project environment. Figure 26 provides an overview of the different dimensions of Ignite IPAT.

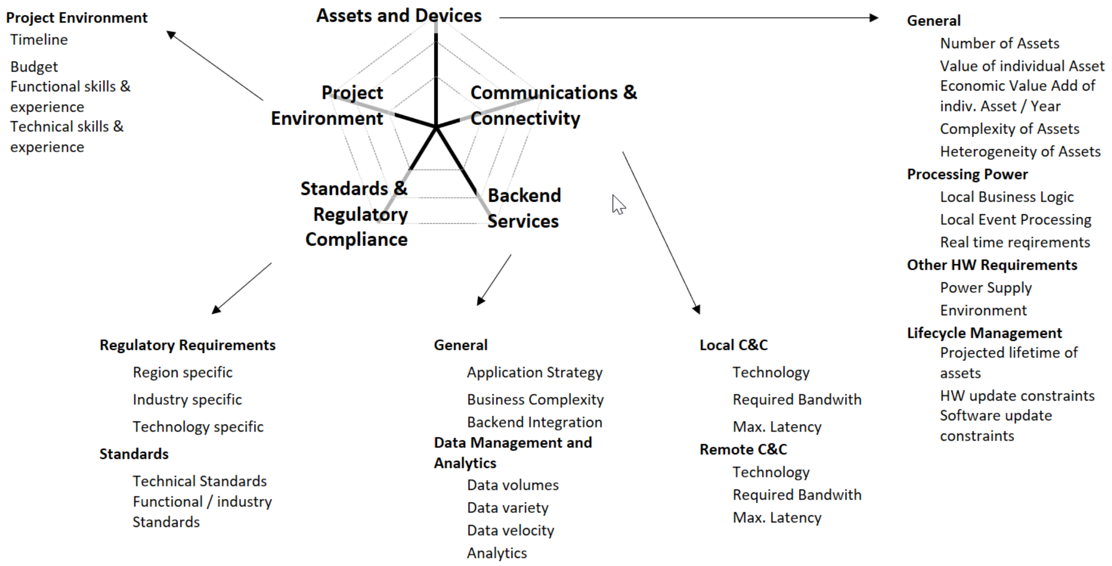

Figure 26: Ignite project assessment tool

One intended use of Ignite IPAT is to allow IIoT project managers to conduct project self-assessments, allowing them to create a holistic view of their project and the technical requirements derived from the initial business model design.

By using a standardized model, Ignite IPAT also supports the comparison of projects in a relatively straightforward manner. It is important to note here that each dimension of Ignite IPAT provides exactly four options. The intention is that option 1 is least critical, while option 4 indicates the highest possible level of criticality. For example, the dimension "number of assets" in Ignite IPAT looks as follows:

Dimension	Number of Assets
Description	Please indicate the number of assets that will be supported by version 1.0 of your solution in this category.
Options	1) 100s 2) 10,000s 3) 100,000s 4) Millions

Version 1.0 of Ignite IPAT has approximately fifty project dimensions, which are all structured alike. By always using four options per dimension, Ignite IPAT aims to make processing the answers provided by end users easier and more flexible.

For example, Ignite IPAT is designed to make it possible to link different technology choices to different dimensions and the four options they each provide, thus enabling a mapping between project assessment and technology recommendations. As can be seen in the following, this structure makes it easy in general to create rules that map Ignite dimensions to matchings results.

3.1.2 IgniteWorx Result Sets for IIoT Project Setup

The goal of IgniteWorx is to provide actionable recommendations for IIoT project managers, based on the individual IIoT project assessment (using the aforementioned Ignite project dimensions).

For these recommendations to be as meaningful as possible for the IIoT project manager, the IgniteWorx design must make assumptions about which areas would be most relevant for an IIoT project manager. To facilitate this, the IgniteWorx design proposes the creation of result sets that can then be used to help group together different topics from different areas. As such, an IgniteWorx result set represents a cluster of possible results from which the IgniteWorx system can then pick the best matching elements and present them to the user.

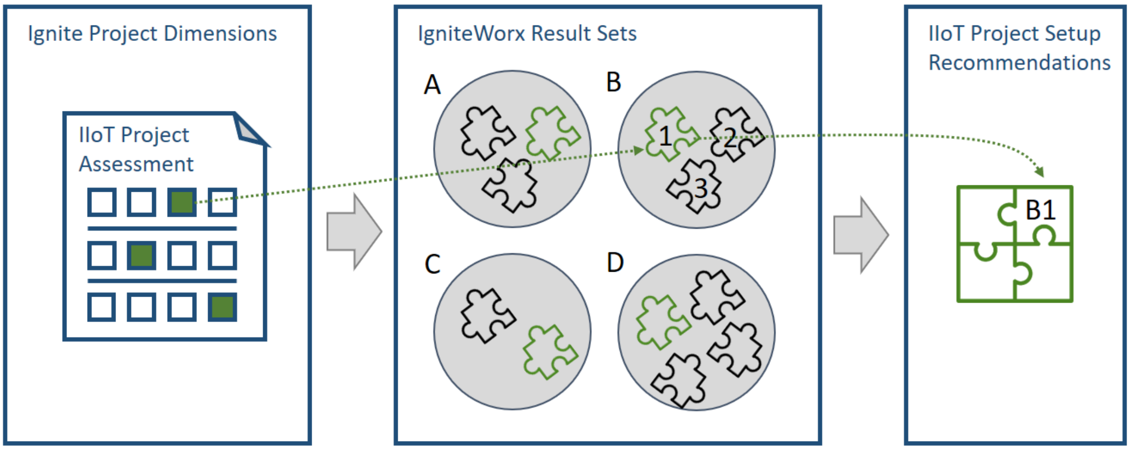

Figure 27: IgniteWorx result sets

Figure 27 provides a schematic overview of how results sets are designed to work. In this example, four sets are defined: A, B, C, and D. Each set in this example contains a number of potential results. For example, result set A contains three potential results, while result set D contains 4. As can also be seen in the overview, the idea is now to take the results of an IIoT project assessment (based on Ignite project dimensions), and map the assessment results to different results in each set. The best matching results are then presented as the recommendations to the user, including B1 in this example.

Ideally, the combination of all matching results presented to the user would create a recommendation for a complete IIoT project setup. However, the key question now is what actually constitutes an IIoT project setup. This must be understood before a proposal can be made that describes how the concrete IgniteWorx results sets are to be defined.

Today, there is no lack of generic or even software-focused project management frameworks. For example, in his paper "Standards and Excellence in Project Management—In Who Do We Trust?" Grau (2013) identifies de-facto standards (e.g. PMBOK, PRINCE 2, ICB 3.0), de jure standards (e.g. ISO 21.500, ISO 10.006, DIN 69900/69901), special standards (e.g. V-Model XT, Scrum, VDA 4.3), and maturity models (e.g. CMMI, SPICE, OPM 3).

However, none of these frameworks is IoT/IIoT-specific. Based on broader literature and Internet research following the guidance from Budgen and Brereton (2006), the following inputs have been considered for the construction of the idealized IIoT project setup for IgniteWorx:

- PMI's PMBOK: The Project Management Body of Knowledge (PMBOK) of the Project Management Institute (PMI) is one of the widely used frameworks, according to Grau (2013). Also, it is very well documented and easily accessible, which is why it was chosen as the representative from the de-facto standards.
- V-Model XT: The V-Model XT was chosen from the group of special standards because it is often used for combined hardware/software projects, which is also a key element of IIoT projects (Rausch et al., 2005).
- Automotive SPICE: SPICE (Software Process Improvement and Capability Determination) or ISO/IEC 15504 is one of the frameworks in the maturity models, according to Grau (2013). Here, Automotive SPICE was chosen because automotive is an interesting asset category within IoT (see introduction).
- Ignite: Since Ignite was chosen as one of the foundations for IgniteWorx, it is also considered here.

3.1.2.1 PMI's PMBOK

PMI is an international nonprofit organization for project management. According to the 2017 annual report (PMI, 2017), PMI has more than 500,000 members and over 400 staff, working on standards and best practices related to project management. A key document of PMI is the PMBOK, the Project Management Body of Knowledge (Project Management Institute, 1987). The PMBOK defines a number of critical standards and guidelines for project management, e.g., the critical path method (CMP) and work breakdown structure (WBS). In addition, the PMBOK defines process groups and knowledge areas. Process groups are defined as initiating, planning, executing, monitoring and controlling, and closing. The ten knowledge areas cover topics such as project integration management, project scope, cost, quality, human resources, communications, risk, procurement, and stakeholders. Together, the process groups and the knowledge areas form a matrix structure. Both process groups and knowledge areas refer to processes that can be mapped to the cells in this matrix (Project Management Institute, 1987).

The concept of PMBOK knowledge areas is close to the basic idea of IgniteWorx result sets since both concepts are designed to help group together different aspects of a project setup in a logical way and, as such, provide a high-level structure for a complex problem domain. In the following, each of the ten PMBOK knowledge areas are examined for suitability for IgniteWorx result sets from the point of view of an IIoT project (Table 7).

PMBOK Knowledge Area	**IIoT-specific aspects**
Project Integration Management	IIoT projects must combine different disciplines, including embedded hardware and software, telecommunications infrastructure, and enterprise application integration. Because of this, multiple project workstreams with complex dependencies can be expected (Slama, Puhlmann et al., 2015). Project integration management, according to the PMBOK, should be helpful to improve the alignment between these different activities.
Project Scope Management	Given the potentially high level of complexity in an IIoT project, project scope management should be helpful.
Project Schedule Management	Given the complex dependencies between different IIoT project workstreams mentioned above, an efficient project schedule management seems key.
Project Cost Management	Project cost management for IIoT needs to combine many different cost factors, from hardware and software development to costs for telecommunications infrastructure used in production.
Project Quality Management	Given that an IIoT solution will potentially support mission critical systems and processes, efficient quality management seems to be essential.
Project Resource Management	The abovementioned complexity will require many different skills in an IIoT project, which would require efficient project resource management.

Project Communications Management	The internal complexity and also the potential dependencies with many external stakeholders in an IIoT project would require efficient project communications.
Project Risk Management	Again, the high level of complexity potentially found in an IIoT project would require efficient risk management, taking IIoT-specific aspects into consideration.
Project Procurement Management	If an IIoT projects combines hardware, software, and telecommunications infrastructure, this requires efficient procurement management for these different solution elements.
Project Stakeholder Management	Especially if the IIoT project has transformational character in a large organization, efficient project stakeholder management seems essential (Porter and Heppelmann, 2014a).

Table 7: IIoT-specific aspects of PMI PMBOK knowledge areas

The initial analysis summarized in Table 7 indicates that the PMBOK knowledge areas could provide a good starting point for the definition of the IgniteWorx results sets since each of the ten areas seems to also be important for an IIoT project. A key question to answer in the following section is how to ensure that the IgniteWorx result sets are kept sufficiently IIoT-specific since it would not make sense to recreate the work done by PMI on a generic project management level.

3.1.2.2 V-Model XT

One of the first descriptions of a V-Model for software engineering was by Rook (1986). In 1997, the German government made use of its own V-Model mandatory for IT projects done for the government (Bund, 1997). Other governments also promote the use of a V-Model in certain areas. For example, DOT (2007) describes the use of a V-Model for intelligent transportation systems in the United States.

The basic idea of the V-Model is to extend the traditional development phases—as found, for example, in the waterfall model—with additional phases for quality assurance. Usually, this is done by mapping each development phase onto a matching quality assurance phase, which results in a V-like visualization (see Figure 28). This built-in support for validation and verification makes it interesting for safety-critical applications.

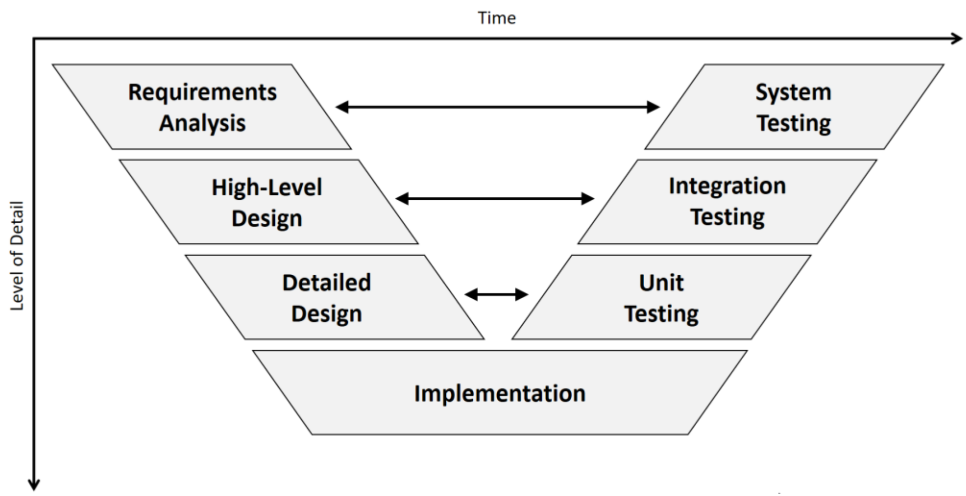

Figure 28: Basic visualization of a V-Model

The proximity to traditional, nonagile development processes also draws some criticism. For example, Balaji and Murugaiyan (2012) compares waterfall, V-Model, and agile and comes to the—maybe not so surprising—conclusion that the V-Model is too rigid to support agile projects.

In 2005, the V-Model XT was introduced with the goal to replace the traditional V-Model (ITZBund, 2005). A key element of the V-Model XT is the ability to customize it so that it can better support different kinds of project settings. The "XT" actually stands for "extreme tailoring."

What makes the V-Model XT interesting from the IgniteWorx perspective is that it was specifically designed with the integration of different stakeholder perspectives in a project in mind. In particular, V-Model XT includes both the acquirer and supplier perspectives.

Figure 29 provides an example of the interfaces between supplier and acquirer according to the V-Model XT. Because XT is designed to be extensible, this example could be modified based on different parameters, e.g., project type, number of suppliers, etc. For example, Rausch et al. (2005) describes how the V-Model XT can be used to support a model-driven development approach and generate development documents in compliance with the V-Model XT.

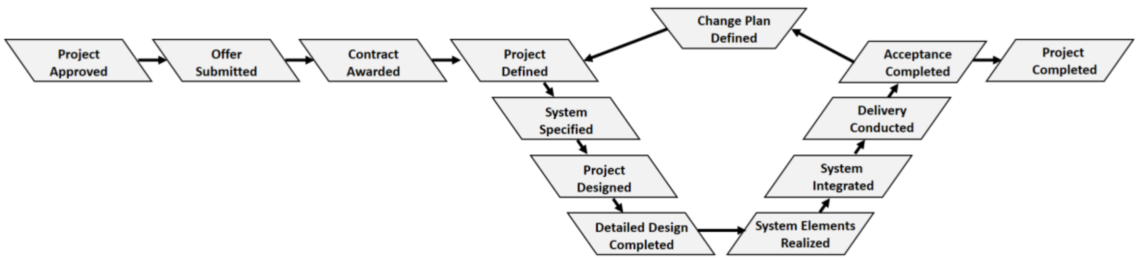

Figure 29: Supplier/acquirer interface according to V-Model XT

IgniteWorx focuses on the project setup. During this phase, supplier selection usually takes place, including complex tender processes. IgniteWorx has the goal to support these processes, e.g., by supporting the creation of better-quality request for proposal (RFP) documents and requirement specifications. As such, the V-Model XT perspective is an important one in the creation of IgniteWorx.

3.1.2.3 Automotive SPICE

Automotive SPICE is an initiative of the Automotive SIG and the Quality Management Center of the German Association of Automotive Industry (VDA QMC, 2017). It is based on ISO/IEC 15504, software process improvement, and capability determination (SPICE).

As defined in VDA QMC (2017), automotive SPICE defines a process reference model and a process assessment model. The process assessment model is designed to support the assessment of the process capability for the development of embedded automotive systems. It was developed in accordance with the requirements of ISO/IEC 33004. The process reference model groups the main processes supported by automotive SPICE into different primary and secondary categories, as shown in Figure 30.

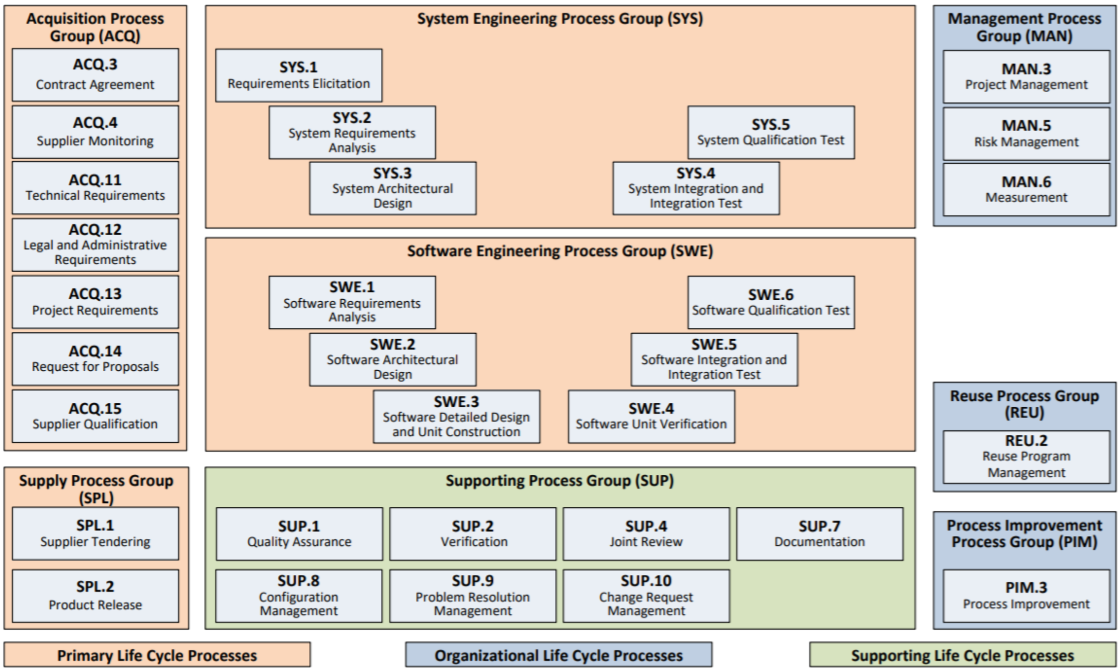

Figure 30: Automotive SPICE process reference model (VDA QMC, 2017)

The main process categories include acquisition and supply processes, system engineering and software engineering, management and support processes, and processes to ensure reuse and process improvement.

Automotive SPICE has been extended in a number of directions. For example, Messnarz et al. (2012) report about experiences with trial assessments combining automotive

SPICE and functional safety standards (IEC 61508, ISO 26262). Kreiner et al. (2013) writes about adding Six Sigma to support a holistic view for process improvements.

Since embedded development is potentially a key part of an IoT/IIoT project, automotive SPICE provides valuable input for this perspective in the development of IgniteWorx.

3.1.2.4 Ignite

The Ignite project management methodology (introduced in section 1.4.2.1, Ignite), is the main methodology artifact created as part of the Enterprise IoT book (Slama, Puhlmann et al., 2015), providing a holistic view on IoT project management.

Ignite examines the entire project lifecycle, including the planning ("plan"), implementation ("build"), and operations ("run") of an IoT solution. Furthermore, Ignite combines the perspective of the asset design and manufacturing with that of the IoT solution lifecycle, as shown in Figure 31.

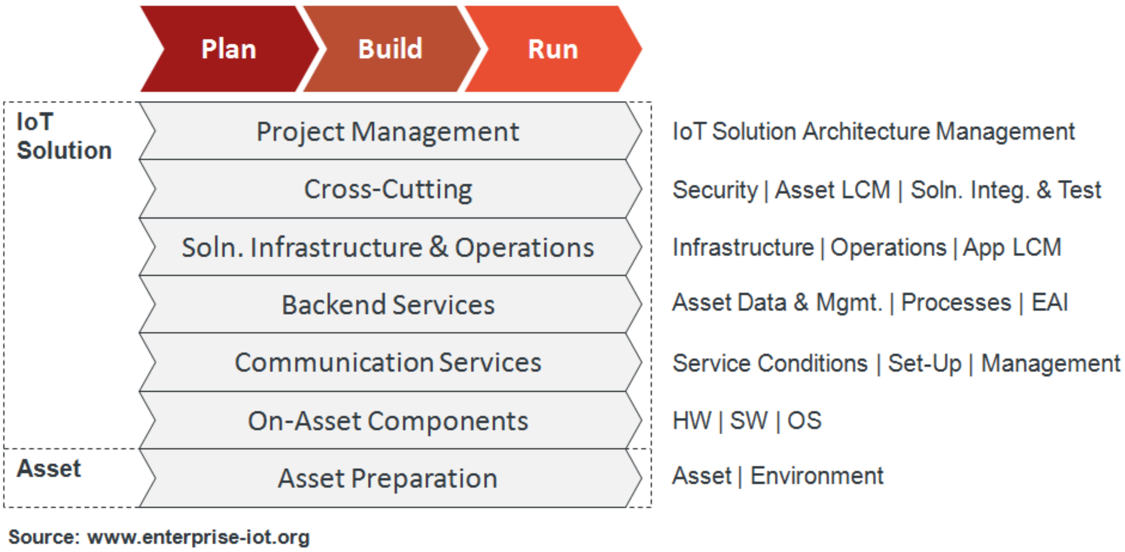

Figure 31: IIoT project workstreams proposed by Ignite (Slama, Puhlmann et al., 2015)

Ignite breaks down the main elements of an IoT project into a set of workstreams. These match the typical architecture of an IoT solution because how organizations are structured has a strong impact on any systems they create, according to Conway`s Law, as described in Brooks (1975).

The main workstreams described in Ignite include project management, cross-cutting functions, solution infrastructure and operations, backend services, communication services, on-asset components, and asset preparation.

The goal of Ignite is not to provide detailed descriptions of each of these individual workstreams but rather a holistic overview of the main workstreams involved in an IoT project and how they must be synchronized. For example, the Ignite on-asset component workstream would have a strong correlation to automotive SPICE (as described in the previous section) since the latter deals with embedded hard- and software, which is a typical on-asset component. However, automotive SPICE mainly focuses on exactly this and is not looking at other required aspects of an IoT project, e.g., the connectivity of the on-asset component to the backend, or the backend solution itself. However, in IoT solutions, this is essential. In many cases, design decisions must be made from such a holistic perspective, e.g., when deciding how to distribute data and services between the asset and the backend (Slama, Puhlmann et al., 2015).

3.1.2.5 Proposed Structure for IgniteWorx Result Sets

The analysis of the different frameworks available today has led to the conclusion that there is no single framework that provides a truly holistic view on the ideal setup of an IoT/IIoT project. However, each of the frameworks previously analyzed can provide significant input for such an approach. Consequently, it is recommended that the design of the IgniteWorx results sets takes the following into consideration from the frameworks analyzed above:

- PMI's PMBOK: The ten PMBOK knowledge areas are a blueprint for the results set concept. PMI introduced this concept to provide a structuring mechanism for the different aspects of a generic project. This thesis proposes to apply this basic idea to the area of IoT/IIoT. Also, similar to the PMBOK approach, it is recommended to keep the initial number of IgniteWorx result sets to 10, in order to provide a good balance between completeness and manageability.
- V-Model XT: The support from the V-Model XT for a clear (and customizable) definition of the interfaces between acquired and supplier is an important inspiration for IgniteWorx to consider since the latter focuses on the project setup phase, which is essential in shaping the future interfaces between acquirer and suppliers.
- Automotive SPICE: The automotive SPICE process model provides a solid blueprint for the required activities of embedded systems development, potentially a key activity in an IoT project.
- Ignite: The Ignite methodology takes a more holistic view of the different workstreams required by an IoT project, including on-asset components, communication, and backend services. This is valuable structural input for the ten results sets defined in the following.

Considering these overall design decisions, the following table provides a high-level overview of the proposed IgniteWorx result sets, including a short description and some background information about why this particular result set is recommended.

IgniteWorx Result Set	Description/Goals	Motivation
A: Project Management Methodology	Which project management methodology is most suitable for the IoT project?	The choice of project management methodology (e.g., agile versus waterfall) has a strong impact on the project and must be carefully considered during project setup. See Ignite.
B: Solution Design	Which design patterns would best support the IoT project?	Reuse of proven design patterns can help to significantly reduce project risks. For example, automotive SPICE includes dedicated process definitions for solution design and reuse.
C: Technology Selection	Which technologies would be most suitable for the IoT project?	Since IoT is still an emerging paradigm, technology selection is a critical success factor. See also ACQ in automotive SPICE.
D: Resource Acquisition	What types of resources would the project need?	Identified by both PMI and automotive SPICE as essential. Must be tailored to the needs of an IoT project.
E: Cost Estimate	What needs to be considered for the cost estimate?	Essential part of the acquirer/supplier relation, as defined by V-Model XT.
F: Risk Management	Which are the most likely risks this IoT project is facing?	Key element of PMI PMBOK; must be tailored for needs of IoT projects.
G: Trust and Security	Which aspects need to be considered from a trust and security perspective?	Key element of IIC's Security Maturity Model (IIC, 2018c); also essential in automotive SPICE.
H: Reliability and Resilience	Which aspects need to be considered from a reliability and resilience perspective?	Very important aspect for mission-critical IoT solutions; must be considered from the beginning (Taft, 2017).
I: Verification and Validation	Which aspects must be considered from a verification and validation perspective?	See SYS and SUP.2 in automotive SPICE.

J: Service Operations	Which aspects must be considered from a service operations perspective (e.g., call center and field services)?	Aftermarket services must already be considered during solution design (Porter and Heppelmann, 2014a).

These ten results sets are the first design proposal, based on the analysis of the frameworks from the previous section, plus additional inputs (see individual references). Figure 32 provides an overview of how the proposed IgniteWorx results sets map to the main Ignite workstreams.

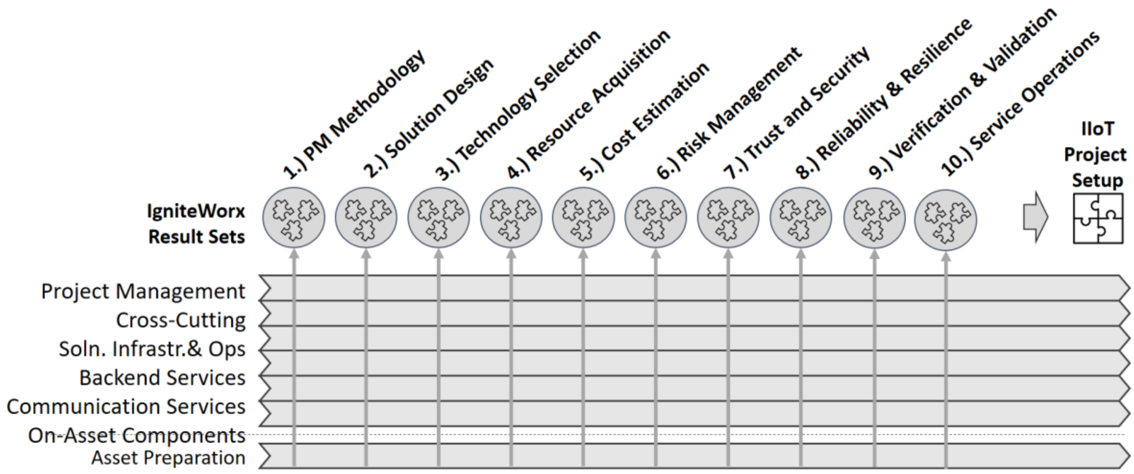

Figure 32: Proposed project setup for IgniteWorx result sets

Since the result sets are a key artifact of this research project, they are discussed in more detail in section 3.2. Also, following the design sciences approach described by Peffers et al. (2007) and used as the foundation for this thesis, this initial design proposal must be evaluated during the evaluation phase of this research project. This is documented in section 5.7.2.

3.1.3 IgniteWorx Rules

The third main artifact of IgniteWorx are the IgniteWorx rules required to identify the best matching project setup for an IIoT project, based on the results of the project assessment. This means the IgniteWorx rules are the link between the Ignite project dimensions and the IgniteWorx result sets.

In the following, the matching mechanisms in general are discussed, followed by a concrete example, and a discussion of the matching mechanics.

3.1.3.1 Mappings Based on IgniteWorx Rules

The IgniteWorx rules provide a mapping between two perspectives. The first perspective is defined by the Ignite dimensions, the foundation of the project assessments. In Ignite, project dimensions are clustered into five categories: assets and devices, communications and connectivity, backend services, standards and regulatory compliance, and project environment. Each category contains a number of project dimensions used to assess a specific part of the IIoT project. There are approximately fifty project dimensions in Ignite 1.0.

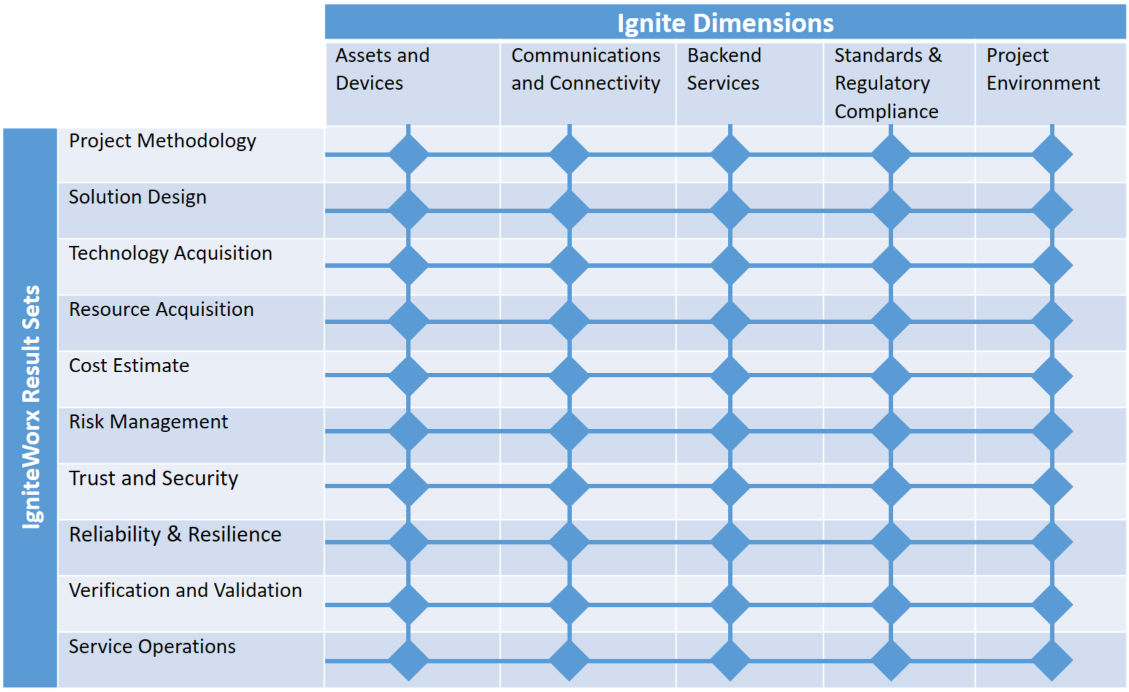

Figure 33: IgniteWorx rules mappings

The second perspective is represented by the IgniteWorx result sets, as defined in the previous section. There are ten result sets in the initial proposal. Figure 33 provides an overview of the mapping between these two perspectives. In the current setting, there are potentially 10x50=500 intersections in the resulting matrix. This means that there are potentially 500 rule definitions required for an implementation of IgniteWorx. This thesis only provides a set of selected examples for these. To efficiently manage such a large amount of content, an online version of IgniteWorx is required, as is shown, for example, in section 4 of this thesis.

3.1.3.2 Example

In the following, project management has been chosen as one of the ten IgniteWorx result sets to help illustrate the concept. It should be noted that section 3.2.1 provides a more detailed analysis of this topic. For the purpose of this initial discussion, it is assumed that the result set has four potential results:

- Agile: To be recommended for small IoT project teams with solid technical and domain-specific/functional skills. Also only to be recommended if there are no serious limitations with respect to software updates.

- Scaled Agile: To be recommended for large projects with a high level of business complexity and no stable requirements.
- Waterfall: Large project, complex stakeholder environment, development of stable requirements is a key goal.
- V-Model: Large project, regulatory requirements imply a strong focus on verification and validation.

Figure 34 provides an overview for how such a structure can look. In this case, it is a matrix with two dimensions: on the X-axis are different options from the project management result set; on the Y-axis are different examples from the Ignite project dimensions (not all have been listed for this specific example).

			Result Set: Project Management			
			Agile	Scaled Agile	Waterfall	V-Model
Project Dimensions	Assets and Devices	Number of Assets			⇧	⇧
		Economic Value Add of Asset			⇧	⇧
		Complexity of Asset		⇧		
		Heterogeneity of Asset		⇧		
		HW Update Constraints			⇧	⇧
		SW Update Constraints	⇩	⇩	⇧	⇧
	Backend Services	Application Complexity		⇧		
		Business Complexity		⇧		
		Backend Integration		⇧		
	Project Environ.	Timeline		⇧		
		Technical Skills	⇧	⇧		
		Functional Skills	⇧	⇧		

Figure 34: Example of rules mappings

In the cells that represent the mapping between a result set element and a project dimension, an arrow goes either up or down. Each arrow represents an instance of an IgniteWorx Rule. An up arrow indicates that the matching is relevant if the dimension has been rated as high for the project at hand. A down arrow indicates the matching is relevant if the dimensions has been rated as low. Each rule is structured as follows:

```
IF (Dimension IS HIGH / IS LOW) THEN RELEVANCE OF (Result) INCREASES
```

Or, using an example from Figure 34:

```
IF (Complexity of Asset IS HIGH) THEN RELEVANCE OF (Scaled Agile) INCREASES
```

3.1.3.3 Matching Algorithm

The matching algorithm should use the rule definitions as follows:

- For each element in each result set, the algorithm must iterate over all related project dimensions to pick the best matching result for a given project assessment.

- Since project dimensions are structured so that each dimension offers exactly four options, it is easy to determine if a project is allocating a high or low relevance to a particular dimension.
- This information can be used to check for each dimension:
    - If a project has indicated that the currently evaluated dimension is ranked as high for the project, *and* there is a rule that indicates a matching result is relevant if this is the case, then the relevance of the result is increased.
    - For low, the same instruction must be repeated with the inverse logic.
- For each result, the number of matching dimensions with a high relevance are counted; the one with the highest count in a given result set should be recommended.

This basic algorithm should be straightforward to implement. However, it should be anticipated that manual creation of the rules will require significant thought and experience, as well as working to balance out the number of rules assigned to different result set elements to create relevant results. Such work requires good online tool support, which is why this thesis does not provide a full set of IgniteWorx rule definitions but rather implements examples, such as the one presented in section 4.1.

3.2 Content Details: Result Sets

Section 3.1 introduces the three key artifacts of IgniteWorx: Ignite dimensions, results sets, and rules. Based on the research described in section 3.1.2, a proposal for an initial setup including ten result sets is made in section 3.1.2.5 (“Proposed Structure for IgniteWorx Result Sets”).

This section looks at each of the proposed ten result sets in detail. The goal is to specify them to a level of detail where an implementation of IgniteWorx can use them as a starting point for bootstrapping the initial content of the system. The proposal in this thesis should be sufficient to provide concrete guidance for the initial setup of an IgniteWorx system; ideally, it can be combined with an existing IoT knowledge base such as described in the example in section 4.1.

The approach taken in this section is based on the following methods:

- Literature research, based on the proposal from Budgen and Brereton (2006), who describe a process for “Performing Systematic Literature Reviews in Software Engineering”
- Reviews of initial research results with subject matter experts, following guidance provided by US General Accounting Office (2006), Dick (2002), and McNamara (2009) (Two subject matter experts were selected for the review of this section based on their general experience in project management, as well as their experience with IoT projects.)

- Use of the material as foundation for a lecture and review with university students at the IIC Winter School 2019, hosted by Ferdinand Steinbeis Institute

The assumption is not that the results presented in this section are already validated according to scientific standards. As outlined in section 2.7 ("Constraints and Assumed System Evolution"), the assumption is rather that a system like IgniteWorx will evolve over time, incorporating feedback from real-world projects and expert users for validation purposes. In the scope of this thesis, a number of measures are described to support this process:

- Section 5 ("Evaluation") describes a cross-case study based on four concrete projects, which is designed to evaluate the initial version of IgniteWorx, including the result sets concept. This is part of iteration I of the research process followed by this thesis.
- Iteration II proposes an extension to the initial IgniteWorx system, which specifically allows to capture user feedback on the recommendations created by the system, including results sets, individual results, and the rules used to derive the concrete recommendations.

Hence, the content described in this section is intended as helping to bootstrap an IgniteWorx system, which then undergoes continuous content review and optimization by end users and systems administrators.

To support this process of bootstrapping the results set definitions of an IgniteWorx system, this section uses the following structure for each result set subsection:

- Overview: The overview provides a short summary of the state-of-the-art research and industry practices for the given practice area of a result set.
- IoT Perspective: This part builds on the general practices and evaluates them from the point of view of an IoT project. The goal is to identify concrete, IoT-specific practices that can feed into the definition of result candidates.
- Result Candidates: This part takes the input from the overview and the IoT perspective and defines candidates for actual results in a given result set. As discussed above, these are candidate recommendations that must then be further refined and reviewed throughout the evolution of an IgniteWorx instance.
- Matching Considerations: As outlined in section 3.1.3.3 ("Matching Algorithm"), IgniteWorx requires a matching algorithm that maps the different Ignite dimensions to potentially relevant results from the result sets. This algorithm is based on a set of rules that describes relevant matches. Each result set proposal includes a discussion of considerations for such matches.

The following section applies the above structure to each of the ten initial result sets proposed for IgniteWorx: project management methodology, solution design, technology selection, resource acquisition, cost estimate, risk management, trust and security, reliability and resilience, verification and validation, and service operations.

3.2.1 A: Project Management Methodology

The first result set in IgniteWorx is designed to support the selection of a suitable project management methodology (PMM) that best fits the specific characteristics of the IoT project being considered. This is important because the choice of a suitable PMM can potentially have a big impact on the project's success. For example, in "The impact of project methodologies on project success in different contexts," Joslin and Müller (2014) conclude that "the application of a PMM account for 22.3% of the variation in project success."

3.2.1.1 Overview

Decades of research have been devoted to creating new PMMs and comparing existing ones. The number of generic and IT-specific PMMs and variations can be overwhelming. Empirical studies like that of White and Fortune (2002) have been designed to "capture the 'real world' experiences of people active in project management." Rehman and Hussain (2007) provide a comparative approach to PMMs.

As discussed in section 3.1.2, Grau (2013) identifies de-facto standards (e.g. PMBOK, PRINCE 2, ICB 3.0), de jure standards (e.g. ISO 21.500, ISO 10.006, DIN 69900/69901), special standards (e.g. V-Model XT, Scrum, VDA 4.3), and maturity models (e.g. CMMI, SPICE, OPM 3).

In the IT and software world, a number of important PMMs have been established over the last several years (Mahalakshmi and Sundararajan (2013); Rehman and Hussain (2007)):

- Waterfall: Especially large, multi-stakeholder projects typically rely on elaborate and detailed specifications that lend themselves to waterfall-like project structures. West et al. (2011) even claim that "Water-scrum-fall" is still the reality of agile for most organizations today.
- V-Model: As introduced in section 3.1.2.2, V-Model is often used in projects with high regulatory requirements because it combines a waterfall-like approach with a built-in verification and validation approach. V-Model XT allows for easier customization to fit specific project needs.
- Agile: Agile software development allows requirements and the corresponding implementation to evolve over time, supported by a collaborative effort of self-organizing and cross-functional teams. A popular example of an agile PMM is SCRUM, which is specifically designed for smaller teams that work on time-boxed iterations called "sprints."

- Scaled Agile: There are a number of PMMs specifically designed to apply the concepts of agile development to larger teams or even multiple teams. Examples include scaled agile framework (SAFe), large-scale scrum (LeSS), scrum of scrums, and Scrum@Scale.
- Iterative/Agile V-Model: Practitioners like Sami (2018) propose combining V-Model and agile (at least the iterative aspects of it) to fulfill the needs of both verification/validation and agile project requirements.
- DevOps: Especially in large organizations, development and operations teams are still often separate organizations, potentially causing huge problems and inefficiencies. Building on Lean and agile practices, DevOps addresses this issue by combining end-to-end automation in software development and delivery (Ebert et al., 2016b).
- In-house project management methods and ad hoc PMMs: According to White and Fortune (2002), the number of projects that only use a very limited number of the available PMM tools or customized in-house best practices should not be underestimated.

3.2.1.2 IoT Perspective

As was discussed in section 1.4.2 ("Lack of established IIoT Project Management Methodologies"), there are currently no well-established PMMs with a specific focus on IoT. Ignite (Slama, Puhlmann et al., 2015) and the IoT methodology (Collins, 2014) could be potential starting points.

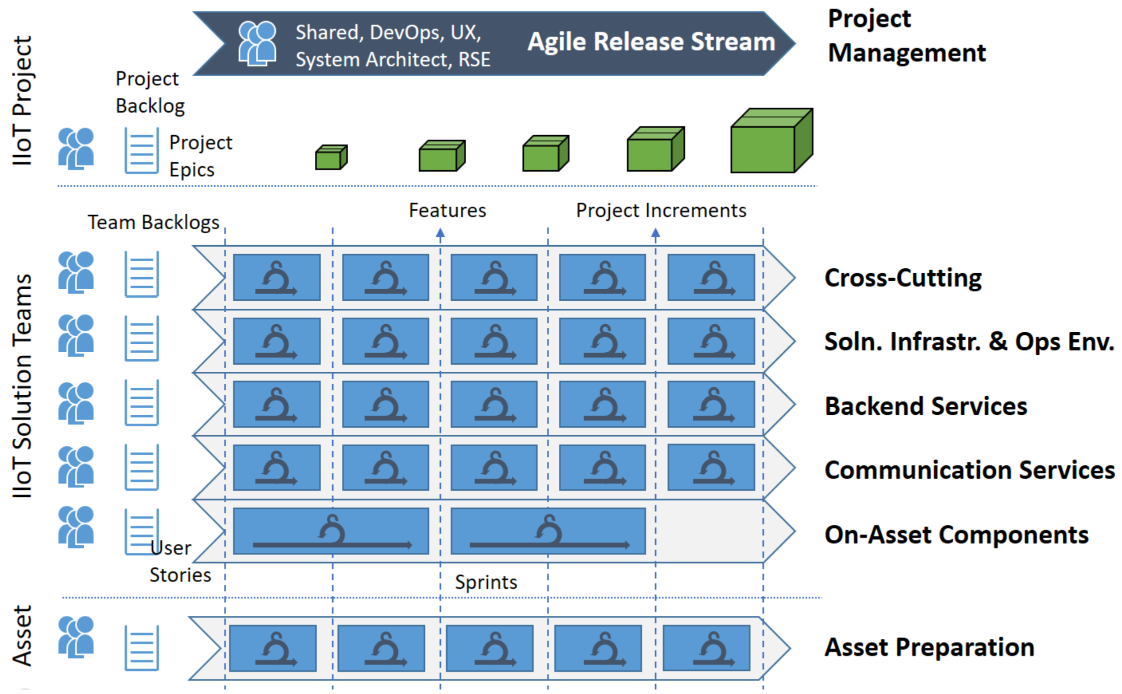

Figure 35: Applying Ignite concepts to SAFe (Slama, Durand et al., 2015)

Slama, Durand et al. (2015) describe how key concepts of Ignite could be applied to SAFe (Leffingwell, 2016) as an example of scaled-agile PMMs. As shown in Figure 35, the basic idea is to apply the IoT workstreams identified by Ignite to teams in SAFe. Sprint durations can be varied, depending on whether a team is working on hardware or software. The SAFe agile release train concept ensures that the outputs from the Ignite workstreams are integrated in regular intervals.

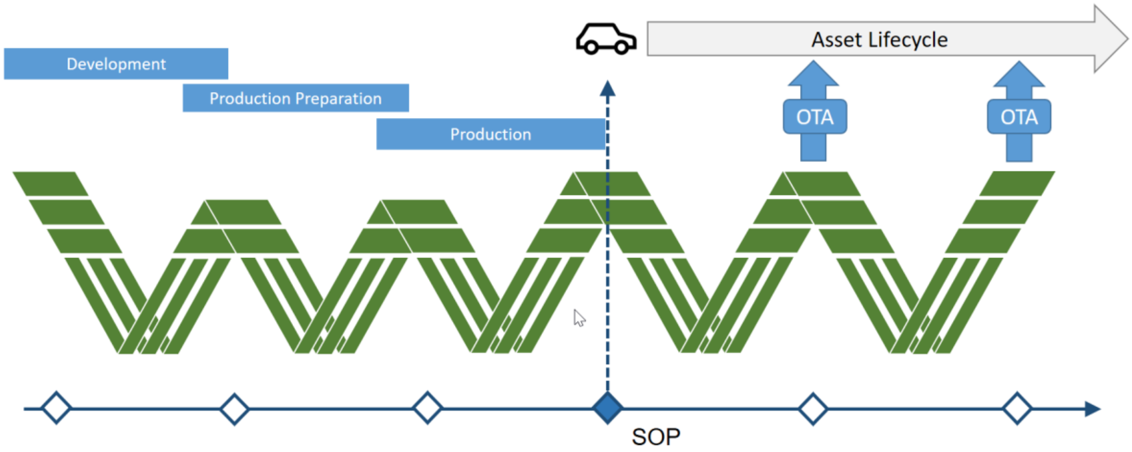

Figure 36: Iterative V-Model and IoT DevOps

Based on the analysis from section 1.2, many IoT projects, and especially IIoT projects with a more industrial focus, will have quite significant requirements for verification and validation, which would lend itself more toward a V-Model-based approach. This could be addressed by the proposal from Sami (2018) and others to combine the V-Model with the incremental approach of agile PMMs. However, this would still not address a second, important specific of IoT projects: In IoT, one has to assume that there will be appliances or assets in the field, running embedded and connected software. This poses a challenge from a DevOps point of view since the DevOps concepts can address traditional development projects, including on-premise deployments as well as cloud deployments. However, to also support DevOps for assets in the field, the project must support over-the-air updates (OTA). The OTA infrastructure and processes must be integrated, both with the verification and validation processes as well as the iterative development. In particular in industries like automotive, management of larger test fleets in this context, with more regular updates as well as serial production with fewer updates, must be considered. The overall idea for combining V-Model, iterative development, and OTA into one integrated solution is depicted in Figure 36. This is a new concept that has not been validated beyond the initial description of the idea in this thesis.

3.2.1.3 Result Candidates and Matching Considerations

Table 8 provides an overview of potential candidates for the first result set—project management methodology. These candidates are derived from the research presented in the previous two subsections and summarized here for completeness. As described in the introduction of this section, they are neither validated nor complete. The intention is that they can help in serving to bootstrap the content of an actual IgniteWorx instance, e.g., in combination with an existing IoT knowledge base.

Result Set A: Project Management Methodology		
#	Result Name	Description and Matching Considerations
A.1	Ad-Hoc/IoT	This could be the minimal definition of a company-specific PMM, with IoT specifics taken into consideration (e.g., extending standard testing procedures to include field tests for IoT appliances).
A.2	SCRUM/IoT	For small IoT project teams, a standard agile PMM could be used as the foundation, potentially combined with elements from an IoT methodology like the IoT methodology (Collins, 2014).
A.3	Scaled Agile/IoT	For larger IoT projects, an approach as described by Slama, Durand et al. (2015) could be recommended, e.g., combining SAFe and IoT, as described in Figure 35.
A.4	Waterfall/IoT	For large IoT projects with many complex stakeholder dependencies, the traditional waterfall approach might be the best solution since it will eventually require the development of stable requirements as the main contract among all stakeholders.
A.5	V-Model/IoT	For larger IoT projects with strong regulatory requirements and stable requirements, V-Model (or V-Model XT) with extensions for IoT could be recommended.
A.6	Agile V-Model/IoT	Similar to A.4, but for projects with less stable, continually evolving requirements: combined V-Model with an iterative approach.

Table 8: Result Set A—project management methodology

3.2.2 B: Solution Design

The second result set in IgniteWorx is designed to support the selection of a suitable approach for defining the IoT solution design. This assumes that IgniteWorx is typically used after creation of the initial business architecture but before the start of the implementation phase (see section 2.1, “Support for IIoT Project Setup Phase”).

3.2.2.1 Overview

There are number of different methodologies explicitly supporting solution design, including rational unified process (RUP and RUP-SOA), Microsoft solutions framework (MSF), IBM's service-oriented modeling and architecture methodology (SOMA), and model-based system architecting and software engineering (MBASE). Reyes-Delgado et al. (2016) provide a good overview of strengths and weaknesses of the different approaches.

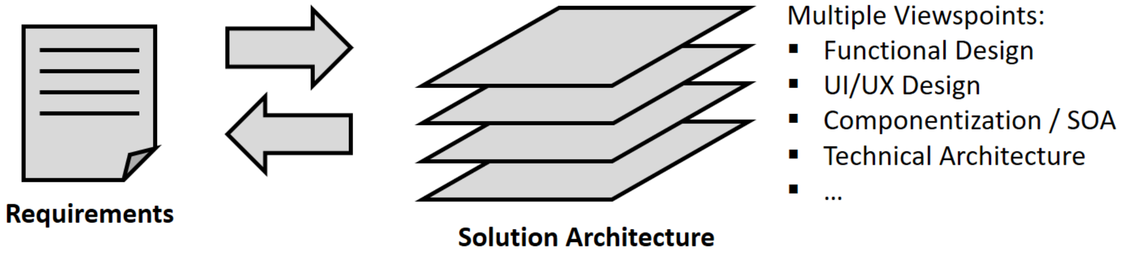

Figure 37: Aspects of Solution Design

There seems to be no industry-wide consensus of what exactly encompasses solution design. However, most approaches agree that solution design has to support the mapping of the higher-level business requirements to a holistic solution architecture which then serves as the foundation for the detailed software design and implementation. Depending on the methdology or framework chosen, the process and the details of the solution architecture differ. Based on Reyes-Delgado et al. (2016) and others, typical viewpoints of the solution architecture include functional design, UI/UX design, componentization/SOA and technical architecture, as shown in Figure 37.

A good example for a holistic, multi-view architecture development method is provided by the open group architecture framework (TOGAF): TOGAF defines the so-called architecture development method (ADM). TOGAF'S ADM can also be combined with TOGAF's SOA reference architecture (Alwadain et al., 2013). Given that many IoT applications must be seen in the context of an enterprise's existing application landscape, this could be an interesting starting point.

Figure 38 provides an overview of TOGAF ADM. At the center of the process are requirements, which are seen as driving the entire process. Key artifacts of TOGAF ADM include business architecture, information systems architecture, and technology architecture. These are supported by processes like migration planning, architecture change management, and so on.

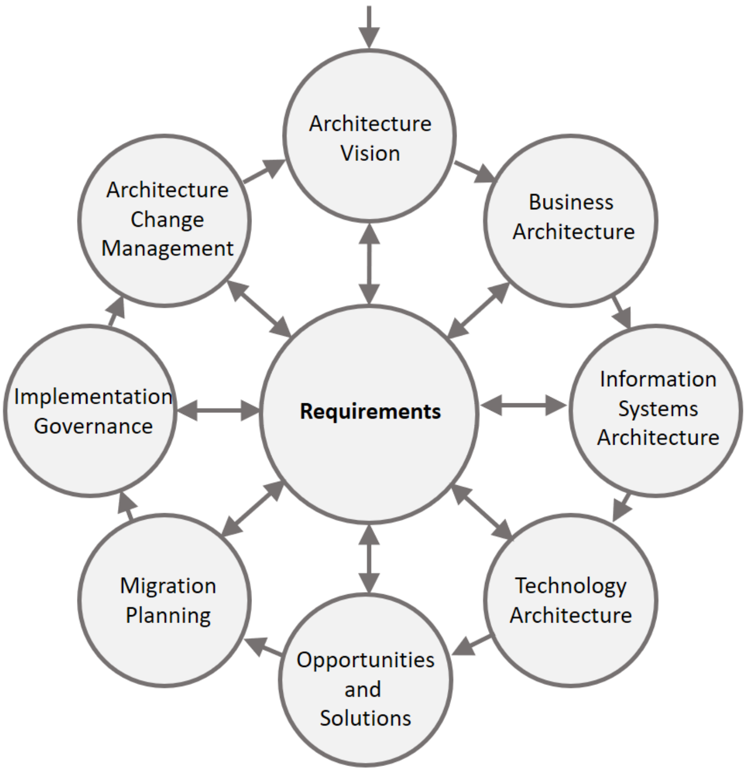

Figure 38: Architecture development method according to TOGAF 9

In McSweeney (2014), a structured approach to solution design is described, which is derived from TOGAF. This approach defines the following viewpoints for a holistic solution design, with requiremnts at the center:

- Business view: context (business environment, resources, skills, products, services, value chain, processes), purpose, characteristics
- Functional view: context (related systems, operational processes), stakeholders, characteristics
- Data view: context, entities, roles, interfaces, characteristics
- Technical view: context, structure (system structure, hardware/software/data/integration-infrastructure), operation, development, characteristics
- Implementation view: context, artifacts/products, execution, characteristics
- Management and operations view: context, operational processes, support, operations, characteristics

An important concept that helps to align the different viewpoints typically found in solution design is called capability modeling. For example, Loucopoulos et al. (2015), describe how capability modeling can help align the business and user context with the technology, service, and operational viewpoints. An emerging discipline is capability-driven development (CDD). See Bña (2015) for an overview. España et al. (2014) provides a case study that combines CDD and SOA. Use of capabilities for IT/business alignment using SOA is also supported by TOGAF 9 (see TOGAF (2009)).

Service-oriented architectures (SOA) are a well-defined and widely used concept for solution design (e.g. Krafzig et al. (2005)). SOA-specific solution design methods like SOMA are well documented (Arsanjani et al., 2008). Micro-services are a widely established concept for supporting the implementation viewpoint (Newman, 2015).

3.2.2.2 IoT Perspective

As was discussed in the previous section, many solution design approaches rely on solution architecture frameworks with multiple viewpoints to capture the various aspects of the solution design. A number of corresponding IoT and IIoT reference architectures have been defined and published by academics, nonprofit organizations, and companies offering IoT technologies.

For example, the IIC has published the IIRA (Industrial Internet Reference Architecture), which defines four viewpoints for an IIoT solution (Lin et al., 2015): business viewpoint, usage viewpoint, functional viewpoint, and implementation viewpoint.

- The business viewpoint describes the business vision and objectives for the IIoT solution
- The usage viewpoint describes the expected usage of the IIoT solution from the point of view of human or machine users
- The functional viewpoint describes the structure, interfaces, and interrelations between the components of the solution, as well as interactions with external components
- The technical viewpoint describes the technologies and system components required for implementing the solution. It also defines a number of IIoT architecture patterns, including gateway-mediated edge connectivity, layered databus architecture pattern and three-tier architecture

On a similar level of abstraction, the Internet of Things—Architecture (IoT-A) defines an IoT domain model (Bauer et al., 2013), which includes an IoT information model and an IoT functional model (which includes an IoT communication model and an IoT trust, security, and privacy model).

Commercial IoT vendors such as Microsoft have also defined IoT reference architectures. These tend to focus only on the technology viewpoint and only from the point of view of the particular vendor. For example, the Microsoft Azure IoT Reference Architecture (Microsoft, 2018) uses a Lambda architecture pattern as the foundation, including real-time and batch processing of IoT data. This pattern is then mapped to a number of different, Microsoft-specific technologies for IoT device management, stream processing, machine learning, etc.

A number of papers and blog entries describe generic IoT design patterns. S. Qanbari et al. (2016) provides some edge-computing focused patterns. Reinfurt et al. (2016) offer three patterns, including "device shadow" and "device gateway." Javeri (2018) provides an overview of event-driven architecture patterns for IoT.

IoT-specific reference architectures and supporting design patterns are providing a structure for capturing the various aspects of IoT solution design. However, they typically don`t provide a process for actually deriving the solution design from the requirements. However, there are at least two methodologies which are supporting an IoT-specific solution design process.

First, the IoT methodology (IOTM) should be considered (Collins, 2014). IOTM suggests an IoT lifecycle based on three phases: brainstorming, build, and tune. See section 1.4.2.2 ("The IoT Methodology") for a summary. IOTM proposes to summarize the solution design on a single canvas, including areas for things, end points, middleware, data model, connectivity, IoT services, apps, and users. Figure 39 provides an example of an IOTM canvas.

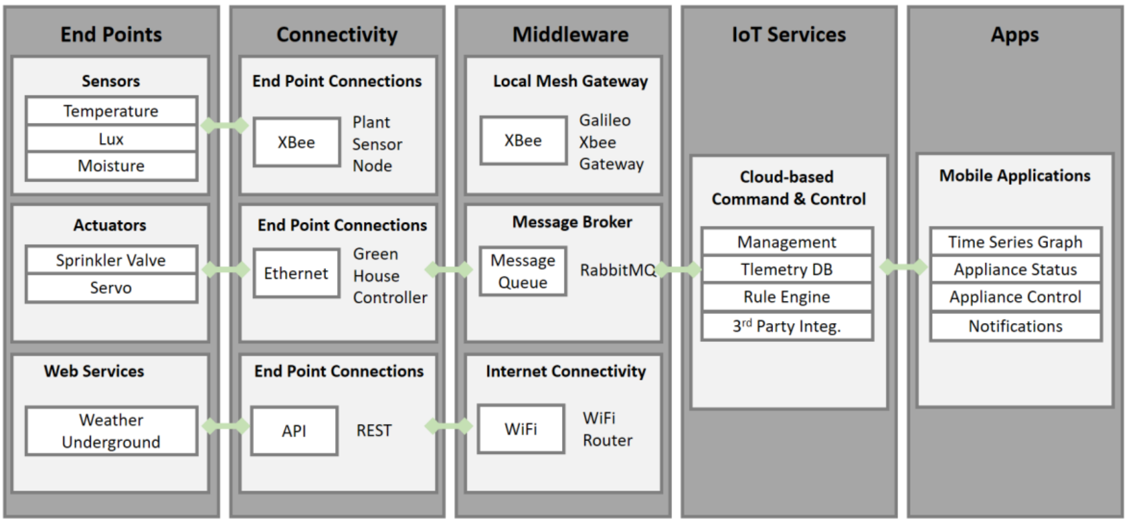

Figure 39: IoT canvas mapping example according to Collins (2014)

Second, the Ignite methodology defines an approach for IoT solution design, based on three phases (Slama, Puhlmann et al., 2015), as shown in Figure 40:

- Analysis, Projections, and Planning: This phase creates a number of artifacts designed to refine the business requirements, validate them on-site, create a high-level solution sketch, analyze the nonfunctional requirements using Ignite dimensions, create a forecast based on a quantitative model (e.g., for number of assets to be supported), and, finally, offer a high-level milestone plan.
- Functional Design: The functional design combines traditional elements of an IT solution design (e.g., process maps, use cases, UI mockups, data-centric domain models, and SOA landscape) with a more IoT-centric perspective (using a so-called asset integration architecture).

- Technical Design: The technical design combines software architecture with technical architecture and the design of the IoT-specific hardware (sensors, gateways, antennas, etc.).

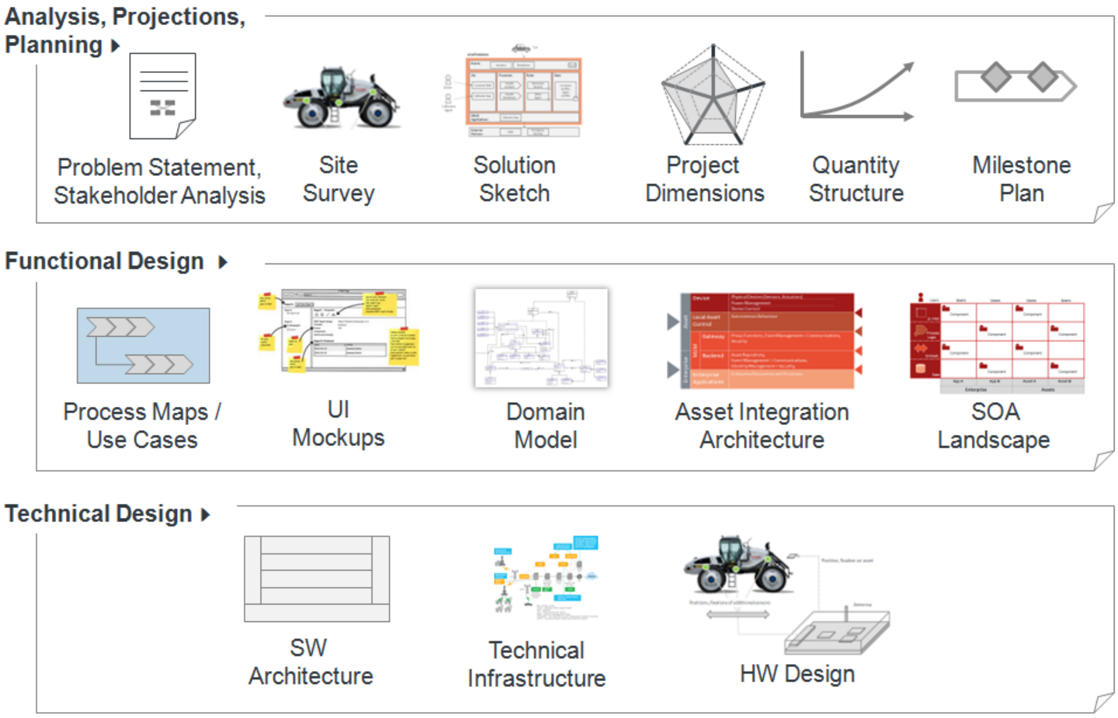

Figure 40: Ignite approach for IIoT solution design

Third, an approach should be considered that combines capability-driven design (CDD), service-oriented architecture (SOA), and IoT. The combination of capability modeling and SOA is described, for example, in TOGAF 9 (TOGAF, 2009). Adding the IoT perspective could result in an approach as described in Figure 41; requirements are derived from the IoT business model architecture (IIC, 2018b). The CDD approach is used to map requirements to capabilities and eventually to support IoT services (IoT functional architecture). Considering the IoT-specific distributed architecture, functional IoT services are mapped to technical components either in the IoT backend or the fog/edge tier. Making this differentiation is important. IoT has—like any distributed system architecture—typical constraints related to bandwidth, latency, etc. However, while most modern distributed architectures for Internet-based or enterprise applications assume relatively high bandwidths and low latencies, IoT systems must be designed based on the assumption that bandwidths can be low and latencies can be high at times or that the asset might even be offline for extended periods of time (e.g., a vehicle driving through a tunnel or in a garage). Taking these IoT-specific design challenges into consideration would be supported by the CDD/SOA for IoT approach described in Figure 41. Again, this is not a scientifically validated approach but only a proposal to be further validated using the long-term validation features built into IgniteWorx (see especially Iteration II, section 6).

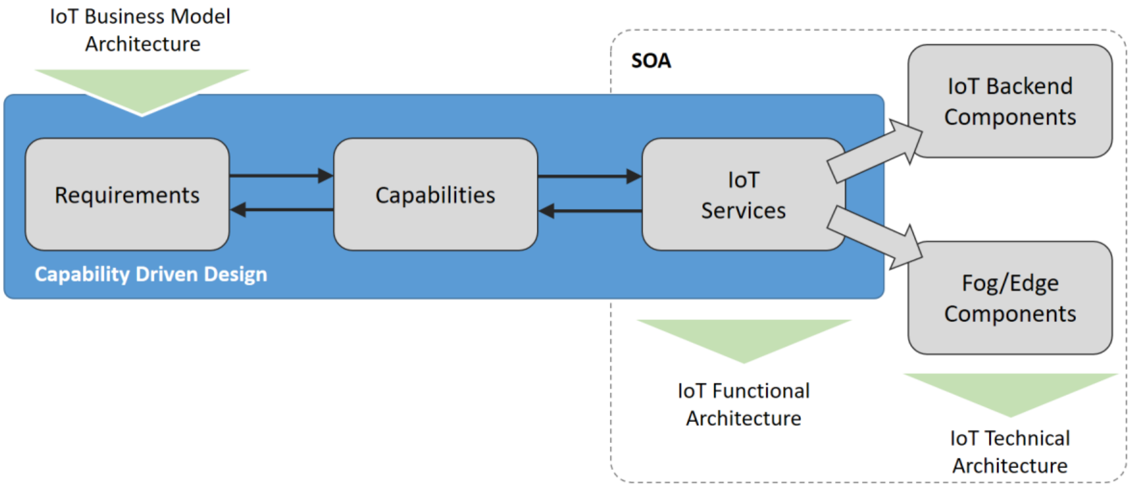

Figure 41: Combining CDD and SOA for IoT solution design

3.2.2.3 Result Candidates and Matching Considerations

Result Set B—Solution Design is actually broken down into two categories:

- Solution Design Methods: Depending on the IoT project characteristics, one of these methods should be chosen.
- Architecture Design Patterns: One of these patterns from the technical viewpoint could also be recommended as the technical target architecture. This could be input to the solution design method. For the moment, mainly the IIC patterns have been included because they seem to be the ones with the strongest industrial validation (through the IIC testbed program), while the other patterns listed above have a more academic background.

Table 9 provides an overview of potential candidates for Result Set B—Solution Design.

Result Set B: Solution Design		
#	Result Name	Description and Matching Considerations
Solution Design Methods		
B.1.i	Lightweight IoT Solution Design Method	A lightweight IoT solution design method, such as IOTM (Collins, 2014), should be recommended for small IoT projects with a very limited functional complexity and small team size, as well as IoT pilot projects and explorative projects done in the context of startups.
B.1.ii	Comprehensive IoT Solution Design Method	A comprehensive IoT solution design method such as the one provided by Ignite (Slama, Puhlmann et al., 2015) should be recommended for enterprise IoT

		projects with a moderate functional complexity and medium-size organizations.
B.1.iii	Enterprise IoT Solution Design Method	For larger enterprise IoT projects with a higher degree of functional complexity and a sizeable team, an enterprise IoT solution design method should be applied. An example could be the DCC/SOA/IoT approach summarized above (see Figure 41).
Architecture Design Patterns (Technical Viewpoint)		
B.2.i	Gateway-Mediated Edge Connectivity (IIC)	This pattern could be recommended to IoT projects with high processing requirements in the fog/edge tier. This could be indicated, for example, by the Ignite dimensions "assets and devices ▸ processing power ▸ local business logic" and " ▸ local event processing."
B.2.ii	Layered Databus Architecture Pattern (IIC)	This pattern could be recommended to projects with a high level of heterogeneity of the asset in the field, as indicated by the Ignite dimension "assets and devices ▸ general ▸ heterogeneity of the asset."
B.2.iii	Three-Tier Architecture (IIC)	This pattern could be recommended to projects with a high level of complexity in the asset ("assets and devices ▸ general ▸ complexity of the asset"), as well as a high level of business complexity in the backend ("backend services ▸ general ▸ business complexity").
B.2.iv	Lambda Architecture	The Lambda architecture pattern (Hasani et al., 2014) supports backend data management for both batch and real-time analytics, a requirement often found in IoT projects (Din and Rubio, 2016).

Table 9: Result Set B—solution design

It should be noted that the functional complexity required to recommend a suitable solution design method could be mapped to the Ignite dimension "backend services ▸ general ▸ business complexity. For "team size," no such dimension currently exists in the Ignite dimensions. This would be a requirement for the next version of Ignite, as described in the appendix (see section 12.1).

3.2.3 C: Technology Selection

The third result set in IgniteWorx is designed to support the IoT project team in the technology selection process. This is a process that must be closely aligned with the solution design process (see section 3.2.2) and the resource acquisition process (see section 3.2.4). In case of an IoT turnkey solution or an IoT COTS solution (as described in section 3.2.4), the selection of the underlying technology will be typically done by the external supplier.

3.2.3.1 Overview

Technology selection is a strategically important process, which unsurprisingly has been widely researched. For example, Durrani et al. (1998) describe a formalized approach to technology acquisition in the context of product development, including a staged process for technology identification, a methodology for technology acquisition, and a decision-making process for sourcing the technology.

Especially in large, global corporations, the technology selection process for an individual project often must be seen in the context of the larger organization. D. Granstrand et al. (1992) look at external technology acquisition in large, multi-technology corporations. For the project manager, it can be challenging to manage these additional, project-external dependencies. Being able to communicate and justify technology selection decisions by using a formal selection process is often very important in larger organizations. Chan et al. (2000) describe a formal evaluation methodology for technology selection. Shehabuddeen et al. (2006) report on the challenges involved in operationalizing a technology selection framework.

3.2.3.2 IoT Perspective

Depending on the required capabilities and other factors identified in the solution design process, technology selection in IoT can include a multitude of different technology segments. Figure 42 provides an overview based on the work done by Slama, Puhlmann et al. (2015).

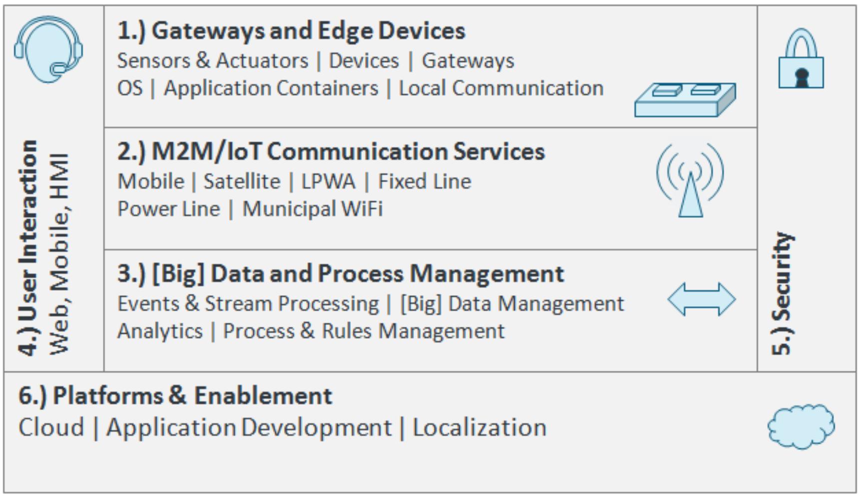

Figure 42: IoT Technology stack (Slama, Puhlmann et al., 2015)

The first technology area described here relates to gateways and edge devices. This can include different standardized or custom sensors and actuators, devices, and gateways. Also, it usually includes embedded operating systems and application container platforms. Local communication services can be required, e.g., to enable wireless communication from the sensors to the gateways.

The second technology area includes wide-area communication networks. These can be the normal mobile phone networks, satellite networks, or specialized IoT networks that have been optimized for low battery consumption to enable portable IoT devices.

The third technology area includes data and process management, such as data analytics, artificial intelligence and machine learning, event processing, and stream analytics. These traditional backend technologies can now also often be found on embedded appliances in the field (fog/edge computing).

Next are technologies for user interaction, which can play a big role in IoT. This includes not only web and mobile application frameworks but can also include hardware and software for human-machine interaction (HMI), which is built directly into the IoT appliances.

Security often also requires use of advanced technologies, e.g., for encryption, certificate management, etc. In IoT, hardware security also often has to be considered to prevent tampering with IoT appliances, e.g., through technologies like trusted platform module (TPM) or similar approaches.

Finally, many IoT projects also use standardized platforms. These can be normal application platforms (cloud or on-premise) without any IoT-specific features. However, projects can also choose specialized IoT platforms that provide additional features like built-in device management, device registries, over-the-air updates (OTA), etc. Many platforms also support specialized IoT communication middleware, supporting a variety of specialized IoT messaging protocols. Furthermore, some advanced IoT architecture patterns such as the aforementioned layered databus architecture pattern can benefit from advanced communication middleware, e.g., based on standards like the data distribution service (Pardo-Castellote, 2003).

Another important enabling technology for many IoT projects consist of localization technologies, which enable 2D or even 3D localization of IoT devices. These typically vary widely in cost and localization precision.

Not shown in Figure 42 but also often very important can be specialized IoT testing platforms, designed to support testing of complex IoT setups in the lab before performing more expansive field tests.

3.2.3.3 Result Candidates and Matching Considerations

The following describes candidates for a result set for technology selection. It is clustered into areas modeled after the overview described in Figure 42. An IgniteWorx implementation could choose to recommend one result from each area or only the best matching results. Also, the list below does not go down to the level of individual products, which would be difficult in a research project like this.

Result Set C: Technology Selection		
#	Result Name	Description and Matching Considerations
Gateways and Edge Devices		
C.1.i	Off-the-Shelf Gateways and Sensors	Should be considered if an off-the-shelf solution is available and fits the project's needs.
C.1.ii	Custom Gateways and Sensors, Externally Sourced	This will better support specialized requirements but also drive project costs up. External sourcing should be considered if no suitable internal manufacturing capabilities are available.
C.1.iii	Custom Gateways and Sensors, Self-Manufactured	This approach will usually only be chosen for highly standardized, large-volume IoT appliances.
C.1.iv	Gateway-less Sensors	This approach can be chosen if sensors can communicate with existing infrastructure, e.g., mobile phones or WiFi.
M2M / IoT Communication Services		

C.2.i	Public WiFi Infrastructure	Often chosen in smart home IoT projects or similar use cases (Li et al., 2011).
C.2.ii	Mobile network (2G, 3G, 4G, 5G)	Applicable for IoT solutions where the standard mobile networks are sufficient in terms of cost, availability, and battery consumption of the IoT appliances.
C.2.iii	Mobile Virtual Network Operator (MVNO)	Can be chosen to optimize global coverage and performance of the mobile network for the IoT solution (Byun et al., 2017).
C.2.iv	Specialized IoT Network (LPWAN)	Specialized IoT networks, such as LoRA, NB-IoT, Cat-M, Cat-NB, and Wi-SUN, or proprietary solutions such as Sigfox are usually chosen for low-cost, low-battery-consumption IoT solutions. Global availability can be an issue (Bardyn et al., 2016).
Data and Process Management		
C.3.i	IoT Analytics Tool	Used to support aggregated analytics of IoT data, e.g., for IoT usage reports, real-time dashboards, asset performance statistics, and threshold visualization (Marjani et al., 2017).
C.3.ii	AI/ML Tool	Used for advanced IoT uses cases, often involving predictive or preventive maintenance (Mahdavinejad et al., 2018).
C.3.iii	Event Processing Tool	Used to support use cases that require real-time reactions to IoT events (Zhang et al., 2014).
C.3.iv	Stream Analytics Tool	Used to identify patterns in IoT data in real time, e.g., for anomaly detection (Yang, 2017). Often requires support from specialized time series databases for efficient data management.
C.3.v	BPM or Case Management	Used to manage the events from IoT devices on the process level, e.g., to coordinate customer interactions (Meyer et al., 2013).
User Interaction		
C.4.i	Integrated Web and Mobile Development Framework	Required for most IoT applications that have direct interaction with human users. IoT UI frameworks also support mapping of IoT device data to specialized UI

		widgets, e.g., to visualize device event data and time series data.
C.4.ii	HMI Development Tool	Usually only required if custom IoT appliances are designed and manufactured (Nuamah and Seong, 2017).
Security		
C.5.i	Basic IoT Security Suite	Almost all IoT solutions will require at least basic security, including encryption and certificate management (Babar et al., 2011).
C.5.ii	Advanced IoT Security Suite	Advanced IoT security solutions will include support for dealing with DDoS attacks, virus detection, automatic network vulnerability checks, etc.
C.5.iii	Hardware Security	Especially for projects with custom hardware development and high security requirements; specialized hardware protection will be required, e.g., through standards like TPM (Kinney, 2006).
Platforms & Enablement		
C.6.i	Standard DevOps Platform	Basic requirement for nearly any IT project today is a platform that supports development and operations in an integrated fashion (Ebert et al., 2016a).
C.6.ii	IoT DevOps Platform	Many large cloud platforms today also offer IoT management services. Alternatively, IoT platforms can be deployed and operated on-premise or on top of basic Cloud-PaaS offerings (Truong and Dustdar, 2015).
C.6.iii	IoT DevOps Platform with OTA	Especially for IoT solutions with complex embedded applications deployed in the field, it could make sense to integrate the OTA (over-the-air) capabilities with the DevOps platform. This is especially true for solutions that are extensively tested in the field. Managing a test fleet that must be frequently updated during the development and validation cycle requires OTA capabilities integrated into the development cycle (Shavit et al., 2007).
C.6.iv	Localization Technology	Some IoT solutions require localization to support features such as geo-fencing or location-based services. For outdoor usage, this can usually be

		achieved through GPS systems. For indoor, a number of different technologies exist, including triangulation using a variety of frequencies and protocols (UWB, BLE, RFID, WiFi), as well as inertial approaches (Zafari et al., 2017).

Table 10: Result Set C—technology selection

3.2.4 D: Resource Acquisition

The fourth result set in IgniteWorx is designed to support resource acquisition by ensuring that, during project setup, the right decisions are made to initiate and support this process.

3.2.4.1 Overview

Managing resource acquisitions as a well-defined process can be a key success criteria for any project. This includes finding the right resource acquisition strategy, as well as implementing it in a way that satisfies all project stakeholders, including the business and technology stakeholders on the project, senior management, legal, and procurement levels.

From a strategic point of view, a key decision for many projects is the make-versus-buy decision. Especially in IT projects, this can be a complex decision process. Schwartz and Zozaya-Gorostiza (2003) look at different models to deal with uncertainty in information technology and propose the real-options approach to decide between acquisition and development projects.

Jones et al. (2001) study determinants and performance impacts of external technology acquisition. A key aspect here is the lifecycle perspective: in the early stages of the lifecycle, companies often compete to develop technical standards for an industry, while during later stages, they are more concerned with producing the standard at the lowest cost. Another key issue identified is intellectual property protection.

Once the make-versus-buy question has been answered, the actual acquisition process starts. If the decision was for an internal solution, available staff with the required skills must be identified and onboarded. If the decision was for an external solution, the next question usually is whether an common-off-the-shelf (COTS) solution should be acquired—and potentially customized—or whether the solution should be based on a custom development project. Badampudi et al. (2016) discuss "Software component decision-making: In-house, OSS, COTS or outsourcing." In either case, an external solution requires running a procurement process, following the corporate procurement procedures and standards.

At the center of such an IT solution procurement process is usually a so-called request of proposal document (RFP, or sometimes request for quotation, RFQ), followed by a vendor selection process.

Web resources like Miller (2010) provide some guidance on this process:

- Pre-RFP Planning: It is key to already involve all stakeholders during the preparation of the RFP, including business and technology, purchasing, finance, and legal experts.
- RFP Document: This document must not only include a very clear set of comprehensive requirements but also additional information like timelines and key milestones.
- RFP Process: Identifying and getting potentially suitable suppliers to answer the RFP requires careful preparation and guidance of the vendors.
- Vendor Evaluation: Especially for complex IT solutions, a solid due diligence process is required. A clear decision and governance model should be the foundation. For contract negotiations, specialized procurement experts should be involved.
- Engagement Model: Especially for custom development projects, a decision must also be made on the engagement model, e.g., fixed price, time and material, or even agile fixed price (Fewell, 2011).

As can be seen in the following, for an IoT project, the resource acquisition process might involve multiple components and specialized vendors if the project is not delivered as a turnkey solution.

3.2.4.2 IoT Perspective

Considering the discussions in sections 3.1.2.5 ("Proposed Structure for IgniteWorx Result Sets") and 3.2.3 ("C: Technology Selection"), it can be seen that resource acquisition for an IoT project often includes the following areas:

1. Project Office: Project management, requirements management, solution architecture management, management of the sourcing process, legal support, etc.
2. Embedded Component Design and Development: Embedded hardware design, embedded software design, and development. Fowler et al. (2019) provide a study of the build-versus-buy decision in developing embedded systems.
3. Embedded Component Production: Internal manufacturing or external sourcing of embedded hardware components according to design.
4. Telecommunications Services: Usually wide-area network-based services, e.g., from mobile network operators (MNO) or mobile virtual network operators (MVNO).

5. Application Software: The software that implements the business logic of the new IoT solution.
    a. Can be on-premise or a SaaS (software as a service) in a public or private cloud. Hughes (2018) provides a discussion on key differences, benefits, and risks of on-premise versus cloud.
    b. Can be COTS or custom development. Ochs et al. (2000) discusses the COTS acquisition process, including definition and application experience. Shahzad et al. (2017) provides a discussion on "Build Software or Buy" from the perspective of large-scale software.
6. IoT Platform: Specialized platform with IoT-specific platform features such as device management and over-the-air (OTA) updates. Can be on-premise or as PaaS (platform as a service) in the cloud.
7. Application Platform: Provides common services such as application server, DBMS, messaging infrastructure, etc. Can be on-premise or PaaS.
8. Application Infrastructure: Physical computing resources, data storage, and networking infrastructure. Can be on-premise or cloud-based infrastructure as a service (IaaS).
9. Solution Operations: This includes the technical operations (e.g., application server availability, database and network administration, etc.), as well as support for the business functionality (e.g., capturing and fixing problems with the functional behavior of the system), including field services (see discussion in section 3.2.10).

For each of these different sourcing areas, the project team must select the best matching approach during the project setup phase. Making the sourcing decision for each individual area usually depends on the overall project strategy. For the purpose of discussion in this context, three scenarios are identified:

- [A] Turnkey solution: In this scenario, the customer would usually only retain a small project office with a focus on requirements management and managing the acquisition process. All other project aspects would be externally sourced.
- [B] Internal development, acquire & integrate: Custom solution development combined with externally sourcing as many off-the-shelf components and infrastructure services as possible.
- [C] Internal development, deep vertical integration: Full control over most aspects of the solution, including on-premise hosting, device manufacturing, etc.

Based on these three scenarios, Table 11 provides a matching to the eight sourcing areas identified above. These matchings are based on some general assumptions and must be further validated, e.g., by the approach outlined in section 6 ("Iteration II: ").

Sourcing Area	A: IoT turnkey solution	B: Internal IoT solution development, acquire & integrate	C: Internal IoT solution development, deep vertical integration
Project Office	Internal	Internal	Internal
Embedded Component Design and Development	External	Internal	Internal
Embedded Component Production	External	External	Internal
Telecommunications Services	External (MNO)	External (MNO)	External (MVNO)
Application Software	External (COTS or custom development)	Internal, custom development	Internal, custom development
IoT Platform	External (IoT-PaaS)	External (IoT-PaaS)	Custom solution, on premise
Application Platform	External (Paas)	External (Paas)	Own data center
Application Infrastructure	External (Iaas)	External (Paas)	Own data center
Solution Operations	External	Internal	Internal

Table 11: Resource acquisition for different project scenarios

As can be seen in this example, all three project scenarios would be similar in that they retain the project office in house. However, only B and C would do the embedded component design internally. Only C would self-manufacture the embedded hardware components. All three would rely on a global carrier for communications services. D might consider the MVNO approach, which potentially provides more control over the service quality in different regions. B and C would do customer application development, while A could choose either COTS or custom development. A and B would run the application on a public cloud stack, while C will run it on premise. Finally, solution operations would only be external in scenario A.

3.2.4.3 Result Candidates and Matching Considerations

The following result candidates are based on the definitions in Table 11.

Result Set D: Resource Acquisition		
#	Result Name	Description and Matching Considerations
D.1	IoT turnkey solution	An IoT turnkey solution approach might be selected in case time to market is important and suitable internal resources to support this requirement are not available. In this case, IP protection and long-term availability of internal solution knowhow must be at least partially forfeited. The match between the final solution and the initial requirements might also not always be the highest.
D.2	Internal IoT solution development, acquire & integrate	This approach will typically be chosen to achieve the best possible match between requirements and end result (Shahzad et al., 2017). Also, IP protection and internal skill building can play an important role (Jones et al., 2001).
D.3	Internal IoT solution development, deep vertical integration	This approach is taken if the company requires a high level of control over the entire solution creation process or if resources are available that must be utilized. Another decision factor (at least for on premise versus cloud) can be data ownership. A typical example of a project with such a high level of vertical integration would be an OEM (e.g., a car manufacturer) who develops an IoT service for a mass-manufactured product.

Table 12: Result Set D—resource acquisition

3.2.5 E: Cost Estimation

The fifth result set in IgniteWorx is designed to support the cost estimation for the complete IoT solution. This can be required, for example, to create a new budget, to validate a budget that was defined before, or to validate external offers for the complete IoT solution or parts of it (see section 3.2.4).

3.2.5.1 Overview

Cost estimation in general is a critical but also difficult task. Often, the cost estimates only address specific parts of the total cost, which is referred to as total cost of ownership

(TCO). Gartner defines TCO as follows (Gartner Group, 2019): “For IT, TCO includes hardware and software acquisition, management and support, communications, end-user expenses and the opportunity cost of downtime, training and other productivity losses.”

Specifically the operations cost often seems to be neglected. Zarnekow and Brenner (2005) provide a multi-case study looking at the distribution of the cost over the application lifecycle: “The results show the central importance of recurring costs for production and further development. For a production time of 5 years these costs amounted to 79% of all life cycle costs, whereas only 21% of the costs were incurred during the planning and initial development stages.”

However, in order to acquire funding for the initial planning and development, it is still important to be able to provide efficient cost estimation techniques for this phase. Menzies et al. (2014) describe current best practices in software development cost estimation as a set of parametric/regression-based techniques, including COCOMO, PRICE-S, SEER-SEM, and SLIM. The general idea is to use a database of past projects to derive an estimate for the current project. Different methods for adjusting to the need of local projects are described, e.g., by limiting the selected past projects to those most similar to the current project. The following problems with these best practices are described: poor modeling assumptions, models with superfluous attributes, noise and multiple correlations, and inadequate training or testing data.

In addition to these parametric/regression-based techniques, there are techniques that look more at the required functionality and its associated costs. One example is function points, which estimate based on the number of functional features, e.g., the number and complexity of screens, the number and complexity of database queries, and so on (Heemstra and Kusters, 1991).

3.2.5.2 IoT Perspective

Following the pattern established in section 3.2.4.2, the TCO discussion for an IoT solution must look at all relevant areas. However, this work excludes opportunity costs of downtimes and other productivity losses since they would be highly context specific.

An important cost factor of most IoT solutions is likely going to be software development. In the case of an IoT solution, and important aspect that will have to be considered here is the typically distributed nature of IoT software components. This distributed nature can significantly increase complexity and consequently also development, test, and validation costs. Dash and Acharya (2011) propose a cost estimation approach for distributed systems using a synthesized use-case point model. Winne and Beikirch (2013) extend COCOMO to include parameters required for effort estimation for distributed embedded systems.

For cloud-based IoT solutions, the cloud operations costs are another important factor. Kalmar and Kertesz (2017) compare IoT cloud operations costs for some of the leading public IoT cloud vendors.

From the point of view of embedded development cost estimation, it is questionable whether the aforementioned software development cost estimation methods are applicable without adjustment due to the specific nature of embedded systems. Debardelaben et al. (1997) go as far as to propose incorporating cost modeling directly into the embedded-system design.

Giannopoulos (2006) provides an interesting case study on embedded hardware and software development cost estimations, based on experience with the Ford Motor Company. For the embedded software cost estimation, he follows the established pattern of use-case-based effort estimates. For the hardware cost estimates, he differentiates between design and development. For the design, he develops a metric based on the following complexity drivers:

1. type of components
2. number of components
3. memory type
4. memory size
5. number of interfaces
6. type of interfaces
7. functionality class
8. distributed functionality
9. test/acceptance criteria

In addition, he cites environmental factors such as:

1. Degrees of contact prevention and guarding against foreign matter (e.g., protection of persons from touching voltage-carrying parts)
2. Degrees of water protection

Especially for mission-critical embedded components, additional costs like certification must be considered. Pop et al. (2013) provide insights on certification costs for different types of embedded platforms.

From a hardware manufacturing point of view, Giannopoulos (2006) describes a model that considers material costs, labor costs, and machine costs, as well as scrap and overhead.

Finally, an IoT TCO calculation must also consider the operations side. Cisco / Jasper (2016) provide an example of the potential distribution of the operational costs of an IIoT solution for monitoring of heavy equipment (see).

Cost Factor	Percent	Assumptions
Network Communication	33–50%	▪ Monthly device/subscription fee ▪ Monthly device/overage fee
Administrative Labor	20–50%	▪ Every deployed device requires at least 15 interactions/year. ▪ Each interaction takes at least 5 minutes.
Technical Support	10–33%	▪ 10% of deployed devices require support. ▪ T1 MTTR: 25 minutes ▪ T2 MTTR: 3 to 5 hours

Table 13: Distribution of operational costs of IIoT solution (Cisco / Jasper, 2016)

It should be noted that the proposal from Cisco / Jasper (2016) does not include operational costs such as application runtime costs, application maintenance, etc. However, it still helps in completing the picture since it at least shows an example of three essential IIoT operations cost drivers. Also, as can be recalled, the work from Zarnekow and Brenner (2005) shows the relationship between initial investment costs and operational costs of a typical IT solution. It is important to keep both sides in mind from a TCO point of view.

3.2.5.3 Result Candidates and Matching Considerations

While the different cost drivers of an IoT TCO calculation are now well understood, the question remains: How much does it actually cost to develop an IoT solution? And what can be recommended to users of IgniteWorx?

Unfortunately, there seems to be very little research available providing concrete cost estimates. However, Klubnikin (2016) provides some numbers:

- MVP: “If you calculate and add up the costs of IoT components (including hardware, infrastructure, mobile or wearable applications and certificates), you won’t arrive at a sum smaller than $50 thousand. That’s how much an MVP version of an IoT solution costs.”
- Custom EKG tracker: “The price of building a custom EKG tracker which analyzes the electrical signals of a human body and visualizes sensor data via a mobile app is estimated at only $300 thousand—but there are hidden costs you might overlook.”
- Complex home automation system: “A complex Home Automation system which uses machine learning algorithms to identify and remember a home owner’s face and automatically adjusts its settings based on the person’s preferences may cost up to $5 million (hardware and software costs included).”

However, it is not clear how these numbers are calculated and what the described projects really entail. Also, it seems that the numbers do not differentiate between who is actually doing the development: a start-up is typically operating under a completely different cost framework than a large corporate organization, which usually has much higher base costs and overheads.

Consequently, the approach described here does not aim to provide predictions for the actual cost of a project. Instead, Table 14 defines a set of project categories. In the following, these project categories are then mapped against cost regions.

Result Set E: Cost Estimation			
#	Result Name	Example	Description and Matching Considerations
E.1	Small-scale IoT Pilot	Humidity monitoring for one particular greenhouse	Small ad-hoc solution / pilot project, without real productization requirements
E.2	Small-scale IoT MVP	Start-up building humidity monitoring for greenhouses as a standard product	Single sensor, WiFi, multi-tenant cloud solution, basic mobile app; fully productized; single market (e.g., EU)
E.3	Industrial IoT Monitoring Solution	Predictive maintenance solution for hydraulic pump	Like E.2, plus multi-sensor, advanced analytics
E.4	IoT Appliance	Connected solution for washing machine or breath analyzer for asthma patients	Custom ECU; OTA; GSM-based connectivity; sold in >100 countries
E.5	Complex IoT Actuator	Automated manufacturing robot	Hard real time; sensor-data fusion; tightly regulated
E.6	Extremely Complex IoT Actuator	Autonomous car; plane; warship; rocket	Huge costs because of extreme technical complexity and physical constraints

Table 14: Result Set E—cost estimation

Taking the project scenarios from Table 14 as the starting point, Figure 43 matches them against a potential cost range, using a log 10 scale. This not a scientifically proven IoT cost framework but rather the starting point for discussions and further evaluations, according to section 6 (Iteration II:). An IgniteWorx user can use the feedback regarding his or her project's positioning in the cost scale from Figure 43 to validate his or her own assumptions and start a more detailed cost estimation process using one of the techniques described in this section.

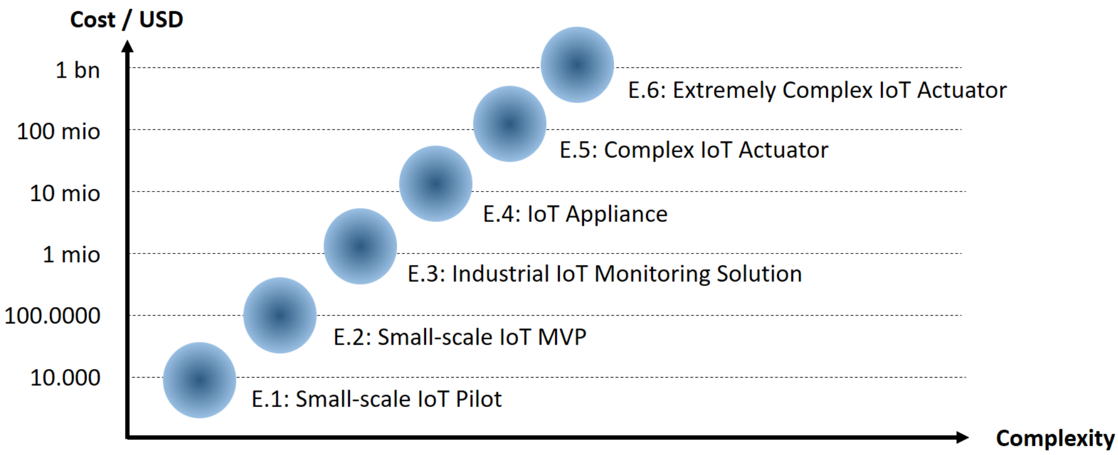

Figure 43: Cost versus complexity for different IoT solution scenarios

3.2.6 F: Risk Management

The sixth result set in IgniteWorx is designed to support the setup of a risk management approach that best matches the project profile.

3.2.6.1 Overview

Depending on the perspective, risk can have different meanings in the context of IoT:

- Industry Risks: These are risks in a particular industry (e.g., manufacturing, energy, or aviation) that are addressed using the new capabilities enabled by IoT.
- Project Risks: These are risks inherent to the project that must deliver the IoT solution.
- Security Risks: These are risks related to the security of the IoT solution, which is a result of the project.

The focus of this section is on the management of project risks, i.e., the risks that a project manager usually has to deal with as part of delivering the project on time, on budget, and according to spec.

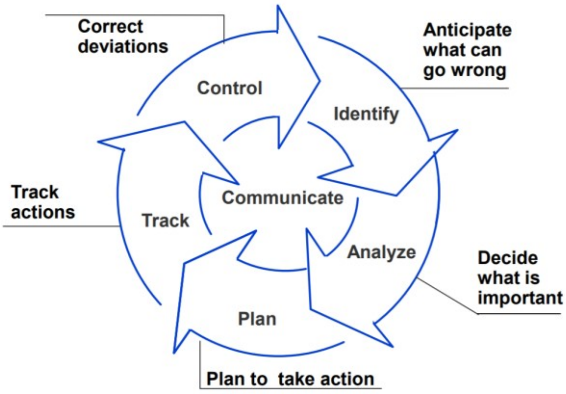

Figure 44: PMI practice standard for project risk management

There are a number of tools, frameworks, and standards available for risk managers. For example, PMI defines a *PMI Practice Standard for Project Risk Management* (PMI, 2009). This document outlines the principles of effective risk management, as summarized by Figure 44:

- plan risk management
- identify risks
- perform qualitative risk analysis
- perform quantitative risk analysis
- plan risk responses
- monitor and control risks

There are a number of other project risk management frameworks, e.g., PRAM for risk analysis and management (Chapman, 1997) or FMECA for failure mode, effects, and criticality analysis (Bowles, 1998). Hillson (2003) describes the use of a risk breakdown structure (RBS) in project management. Yeo and Ren (2009) describe a risk management capability maturity model for complex product systems projects.

There are also a number of established standards in this space, including the ISO 31000 family of standards (Purdy, 2010), ISO 27000 (Disterer, 2013), and NIST (SP800, 30).

COBIT is a framework for IT governance and management in large corporations (Haes et al., 2013). In COBIT 5, two frameworks—Val IT and Risk IT—are merged to provide a holistic view on IT risk management governance.

3.2.6.2 IoT Perspective

Again, a holistic risk management in the context of an IoT project must address all the dimensions of the IoT project, beyond the well-established concepts of software project risk management. Still, since software development is likely a good part of each IoT project, it is a good starting point for the discussion.

Kwak and Stoddard (2004) provide a comprehensive overview of project risk management lessons learned from software development projects. They build on the work of (Boehm, 1991) and others. Boehm (1991) identifies the following ten main risks in software projects:

- Personnel shortfalls
- Unrealistic schedules and budgets
- Developing the wrong functions and properties
- Developing the wrong user interface
- Gold plating (adding more functionality/features than is necessary)
- Continuing stream of requirements changes
- Shortfalls in externally furnished components
- Shortfalls in externally performed tasks
- Real-time performance shortfalls
- Straining computer-science capabilities

Jones (1998) adds the following risks for software projects:

- Risks associated with inaccurate estimating and schedule planning
- Risks associated with incorrect and optimistic status reporting
- Risks associated with external pressures that damage software projects

Again, IoT projects often must take the hardware development perspective into consideration as well. From the point of view of IoT hardware development, Sigfox (2018) proposes the following risk areas:

- Electronic design
- Electronic hardware
- Casing
- Battery
- Radio frequency communications
- Antenna design
- Embedded software development
- Platform development
- Mobile app development

Sigfox (2018) specifically highlights the inherent risks of antenna design for projects using specialized IoT networks (such as Sigfox, NB-IoT, LoRa, etc.) since these are not yet widely established communications technologies.

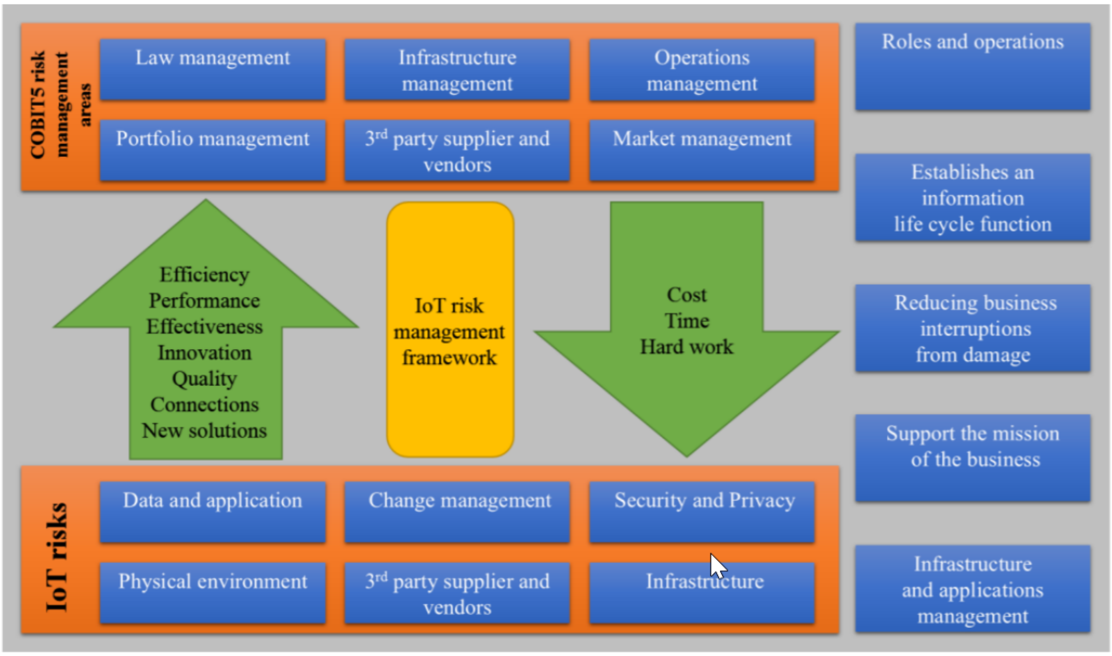

Figure 45: Proposed COBIT 5 IoT risk management framework (Latifi and Zarrabi, 2017)

Finally, Latifi and Zarrabi (2017) propose a COBIT 5-based framework for IoT risk management. The proposal identifies six main IoT risks and then maps them to existing COBIT 5 functions:

- Data and Application: Use the COBIT 5 information lifecycle function to ensure data is secured and available. Data performance can be efficiently measured.
- Security and Privacy: Align business goals and related security risks.
- Physical Environment: Physical security measures help in reducing business interruptions.
- Infrastructure: Ensure security awareness, secure development, and establish security architecture.
- Change Management: Assess, prioritize, and authorize changes with all required risk-impact information available.
- Third-Party Suppliers and Vendors: Use twenty-two existing COBIT 5 mitigation actions to maximize efficiency and minimize audit findings.

3.2.6.3 Result Candidates and Matching Considerations

Table 15 summarizes the recommendations for setting up IoT risk management during the project initiation phase.

Result Set F: Risk Management			
#	Result Name	Example	Description and Matching Considerations

F.1	Embedded IoT Risk Management	Risk management as part of regular project reporting, e.g., risk list and status included as part of weekly project status update, e.g., using a simple risk breakdown structure (Hillson, 2003).	Lightweight approach, no dedicated resource required; use content from section 3.2.6.2 to populate initial IoT risk breakdown structure; applicable to project categories E.1–E.3 in Figure 43
F.2	Dedicated IoT Risk Management	Dedicated risk management function, using tools like risk sector plots to do detailed risk assessment on key risks, following PRAM, FMEA, ISO 3100, or similar.	Requires dedicated risk management resource; applicable to project categories E.3–E.4.
F.3	Dedicated IoT Risk Management with Custom Methodology	Similar to B but building on a custom methodology like COBIT 5 risk response workflow, with custom IoT features (Latifi and Zarrabi, 2017).	Requires dedicated risk management team, including methodology specialist; applicable to project categories E.5–E.6.

Table 15: Result Set F—risk management

3.2.7 G: Trust and Security

The seventh result set in IgniteWorx is designed to support the setup of a project structure that will support the implementation of appropriate trust and security levels.

3.2.7.1 Overview

The widely cited paper from Grandison and Sloman (2000) does a survey of trust in Internet applications. It defines trust as "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context" (assuming dependability covers reliability and timeliness). A taxonomy of different categories of trust is provided, including:

- Access: A trustor trusts a trustee to use resources that he or she owns or controls.
- Service: The trustor trusts the trustee to provide a service that does not involve access to the trustor's resources.
- Certification: Trust can be based on a criteria relating to the set of certificates presented by the trustee to the trustor.

- Delegation: A trustor trusts a trustee to make decisions on its behalf.
- Infrastructure: The trustor must trust him- or herself and his or her base infrastructure.

An important enabler of trust in this context is information security. There are a number of information security management system standards (ISMSs) out there, which attempt to provide more or less holistic frameworks for managing security in IT systems.

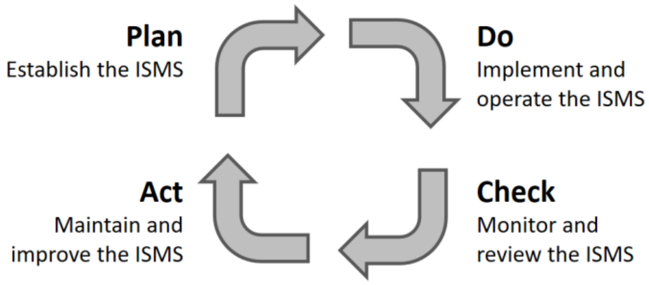

Figure 46: PDCA (“plan-do-check-act”) model for ISMS

Susanto et al. (2011) identifies the “big five” of information security management system standards as follows:

- ISO 27001: This standard introduces a cyclic process model known as PDCA (“plan-do-check-act”), as shown in Figure 46, which is designed to establish, implement, monitor, and improve the effectiveness of an organization’s ISMS.
- BS 7799: Defines a set of best practices for ISMS, which were partly adopted by ISO 27001.
- PCIDSS: The Payment Card Industry Data Security Standard, established by the Payment Card Industry Security Standards Council.
- ITIL: The Information Technology Infrastructure Library (ITIL) defines best practices for IT service management, including security management.
- COBIT: The Control Objectives for Information and Related Technology (COBIT) is an IT governance framework, designed to create an integrated view on control requirements, technical issues, business risks, and security issues.

In the following, these more traditional views on trust and security must be translated to the IoT.

3.2.7.2 IoT Perspective

While security is seen by many as a key element of trust in the IoT, the concept of trust is much broader. For example, IIC and others (2016) identify security, privacy, safety, resilience, and reliability as the key aspects of trustworthiness in the IIoT.

Another good example is provided by the Online Trust Alliance (OTA). OTA has published an IoT trust framework, which consists of four areas (Online Trust Alliance, 2018): security principles; user access and credentials; privacy, disclosures, and transparency; and notifications and related best practices. The IoT trust framework includes the following concrete policy definitions:

- Privacy: Privacy-related policies such as collecting and sharing data from IoT devices must be clearly disclosed; data collection is limited to the data needed to support functionality.
- Disclosures: The IoT solution must include thorough, easily discoverable disclosures covering privacy policies, data collection, and device functionality.
- Updates: Purchasers are informed about device updatability; updates are delivered securely with minimal user intervention or impact.
- Control: Consumers must be provided with choices and controls regarding the data collected by the IoT solution; the solution must support to transfer or wipe the data upon loss or sale.

The European Union Agency for Network and Information Security (ENISA) has proposed a scheme for certification and the development of an associated trust label for IoT devices. In ENISA (2017), a number of required policies are defined:

- Security by design: The security of the whole IoT system must be considered from a consistent and holistic approach during its whole lifecycle.
- Privacy by design: Privacy must be made an integral part of the system.
- Asset management: Asset management procedures and configuration controls for key network and information systems must be ensured.
- Risk and threat identification and assessment: Risks related to the IoT devices must by identified using a defense in-depth approach.

A number of technical measures are also proposed, including:

- hardware security
- trust and integrity management
- strong default security and privacy
- data protection and compliance
- system safety and reliability
- secure software/firmware updates
- authentication
- authorization
- access control—physical and environmental security
- cryptography
- secure and trusted communications
- secure interfaces and network services
- secure input and output handling
- logging
- monitoring and auditing

The IIC security framework (IIC and others, 2016) aims to create a broad industry consensus on how to secure IIoT systems. In addition to security, it highlights safety, reliability, resilience, and privacy as key characteristics of a trustworthy IIoT system. The functional building blocks of the IIC security framework are shown in Figure 47, including the four core security functions: endpoint protection, communications and connectivity protection, security monitoring and analysis, and security configuration management. These are supported by a data protection layer and a systemwide security model and policy layer. The paper stresses the point that IIoT systems are often built with components from different suppliers and that security must be ensured for each of these.

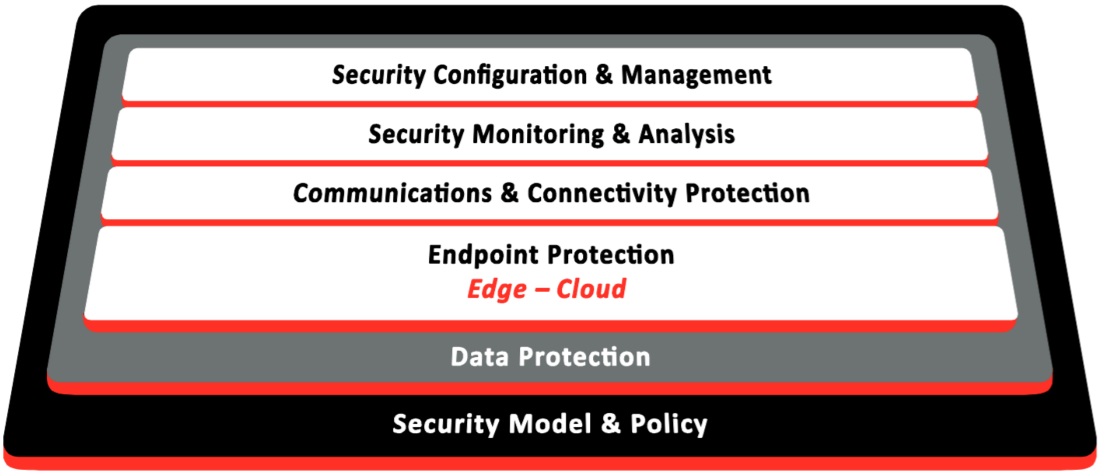

Figure 47: IIC security framework functional building blocks

Of course, the IoT is built on already existing technologies, including standard hardware and software, embedded hardware and software, and communication technologies. A large amount of research on enabling security has been done in the past decades in each of these areas.

Gasser et al. (1989) defines an early digital distributed system security architecture. Meier et al. (2003) describes how to improve web application security. Vai et al. (2016) provides a discussion on how to secure embedded systems.

Detailed research on IoT security—done, for example, by Riahi et al. (2013)—provides a systemic approach for IoT security. Farooq et al. (2015) details a critical analysis of the security concerns of the IoT. Xu et al. (2014) discusses design challenges and opportunities for security in the IoT.

IIC (2018c) defines an IoT security maturity model based on the plan-do-check-act (PDCA) cycle (see Figure 46). IoT security maturity is described in three dimensions: security governance, security enablement, and security hardening. The model defines five maturity levels for each dimension:

1. Level 0 (none): No common understanding of how the security practice is applied and no related requirements are implemented.

2. Level 1 (minimum): The minimum requirements of the security practice are implemented. There are no assurance activities for the security practice implementation.
3. Level 2 (ad hoc): The requirements cover main use cases and well-known security incidents in similar environments.
4. Level 3 (consistent): The requirements consider best practices, standards, regulations, classifications, software, and other tools.
5. Level 4 (formalized): : A well-established process forms the basis for practice implementation, providing continuous support and security enhancements.

3.2.7.3 Result Candidates and Matching Considerations

Table 16 proposes two main areas for the trust and security result set: trust policies and target security maturity.

Result Set G: Trust and Security		
#	Result Name	Description and Matching Considerations
Trust Policies		
G.1.a	Basic	Clearly formulated and publicly available IoT trust policies, e.g., according to the IoT trust framework (Online Trust Alliance, 2018)
G.1.b	Advanced	Dedicated role in the IoT project organization to manage trust policy enforcement during system setup and operations
Target IoT Security Maturity		
G.2.a	L0: None	Acceptable during proof of concept
G.2.b	L1: Minimum	Acceptable for minimum viable product (MVP)
G.2.c	L2: Ad Hoc	Acceptable for non-mission-critical IoT solutions
G.2.d	L3: Consistent	Should be the goal for all IoT solutions with significant criticality
G.2.d	L4: Formalized	Goal for IoT solutions with high mission criticality

Table 16: Result Set G—trust and security

3.2.8 H: Reliability and Resilience

The eighth result set in IgniteWorx is designed to ensure that the IoT project is set up properly with respect to ensuring reliability and resilience.

3.2.8.1 Overview

Reliability and resilience are two related but distinct concepts. Hukerikar and Engelmann (2017) define reliability as "the property of a system that characterizes its probability to have an error or failure." Clark-Ginsberg (2016) provides an example from the energy section: "Reliability can be defined as the ability of the power system to deliver electricity in the quantity and with the quality demanded by users."

Resilience, on the other hand, is concerned with the ability of a system to recover from a failure. Murray et al. (2017) define resilience as the "ability of a system to persevere or work through a major fault in a critical part of the system."

This means that reliability can be described as the end goal, while resilience is one of the key enablers to achieve this goal. Available techniques to ensure resilience include fault tolerance techniques as well as disaster recovery techniques. Hukerikar and Engelmann (2017) summarize the available resilience metrics, including:

- Reliability Metrics: The systems failure frequency
- Availability Metrics: The proportion of time the system provides a correct service, e.g., measured in mean time between failure (MTBF)
- Error and Failure Detection Metrics: For example, the number of failures that were detected and indicated

Literature research reveals a broad spectrum of available techniques and methodologies to ensure high levels of reliability and resilience for IT solutions:

1. Resilience methods within the software development cycle (Murray et al., 2017)
2. Resilience modeling and analysis (Microsoft Trustworthy Computing group, 2013)
3. Resilience testing (Heorhiadi et al., 2016)
4. System redundancies and fail-over techniques (Hukerikar and Engelmann, 2017)
5. Disaster recovery techniques (Wold, 2006)

In the following, the specifics of reliability and resilience are discussed from the point of view of an IoT solution.

3.2.8.2 IoT Perspective

Again, an IoT solution can be treated like a special kind of distributed system, with the added complexity that compute nodes are potentially globally distributed and operated in the field, without easy physical access through maintenance experts.

A good starting point for the following discussion are the the eight fallacies of distributed computing (Deutsch, 1994), which are describing false assumptions about distributed system development. In Table 17, the eight fallacies of distributed computing are looked at from the perspective of IoT. In addition, two new fallacies have been identified, which are specific to IoT solutions.

Fallacies of Distributed Computing	**IoT Perspective**
The network is reliable	Especially for mobile/moving assets and devices in the IoT, this cannot be assumed. For example, a car might drive into a tunnel or a garage with zero network coverage.
Latency is zero Bandwidth is infinite	Many assets in the IoT—like trains, ships, or airplanes—rely on networks with potentially very high latency and low bandwidth, especially if these are satellite-based networks. The same can apply, for example, to an IoT device/asset deployed in a rural area (e.g.. in IoT solutions for agriculture), where no modern cellular networks might be available.
The network is secure	Especially for IoT applications that integrate via the Internet (and not a VPN), this cannot be assumed.
Topology does not change	Again, especially for mobile/moving assets, frequent changes in network topology must be assumed.
There is one administrator	Especially for global IoT solutions, this cannot be assumed.
Transport cost is zero	See discussion on the TCO of IoT solutions in section 3.2.5.2.
The network is homogeneous	Cannot be assumed, especially not in IoT solutions with globally distributed devices and assets
Fallacies of IoT	**IoT Perspective**
All compute nodes can be physically accessed	In an IoT solution, physical access to assets and devices (and the compute nodes running on them) in the field can be very difficult to near impossible. This imposes severe limits on repair and upgrade activities.
All compute nodes can be physically protected	Physical protection of assets and devices in the field can be difficult to near impossible. This can have consequences, for example, from the trust and security point of view. Also, system failures due to severe environmental conditions must be taken into consideration.

Table 17: Fallacies of distributed computing and IoT

Naturally, these IoT fallacies have some consequences for the implementation of a reliability and resilience strategy. In the following, each of the five reliability and resilience techniques from the previous section are looked at from this perspective.

The first technique identified is “Resilience Methods within the Software Development Cycle.” Murray et al. (2017) highlight the importance of designing the software for resilience when the software specifications are being developed. Treat (2015) goes even further, arguing that with distributed microservice architectures, failure is all but guaranteed. This means failure must be embraced from the beginning, by making conscious decisions to anticipate and isolate failure and allow for graceful degradation. The key to being highly available is described as “learning to be partially available.” Some IoT-specific examples are described in the following:

- In an IoT solution, for example, a cloud-based service might collect a number of KPIs from remote devices. The service must anticipate that not all devices will be available all the time. For devices currently not online, the service should still display the latest available set of KPIs, including the information when the KPIs were last updated and the current online status of the device.
- Another example is a smart home appliance that uses cloud-based weather forecast data to optimize room temperature. This application must still provide a basic service, even if the weather data from the external cloud service is temporarily not available.
- The last example is related to server failures: An IoT system should be designed in a way that in case of a server failure, not all IoT devices attempt to reconnect at the same time—this could have an effect similar to a DDoS attack. Instead, the IoT devices could use a random delay before reconnecting to ease the load on the server.

Building resilience methods into the IoT software development cycle requires that all developers are properly trained and best practices are shared. This is something the IoT project manager must consider in his or her project setup.

The second technique is “Resilience Modeling and Analysis” (RMA), as described by Microsoft Trustworthy Computing group (2013). RMA describes a structured methodology for prioritizing engineering investments, with a focus on achieving resilience for cloud-based solutions. It is based on concepts like recovery-oriented computing (Patterson et al., 2002) and failure mode, effects, and criticality analysis (FMECA) (Borgovini et al., 1993).

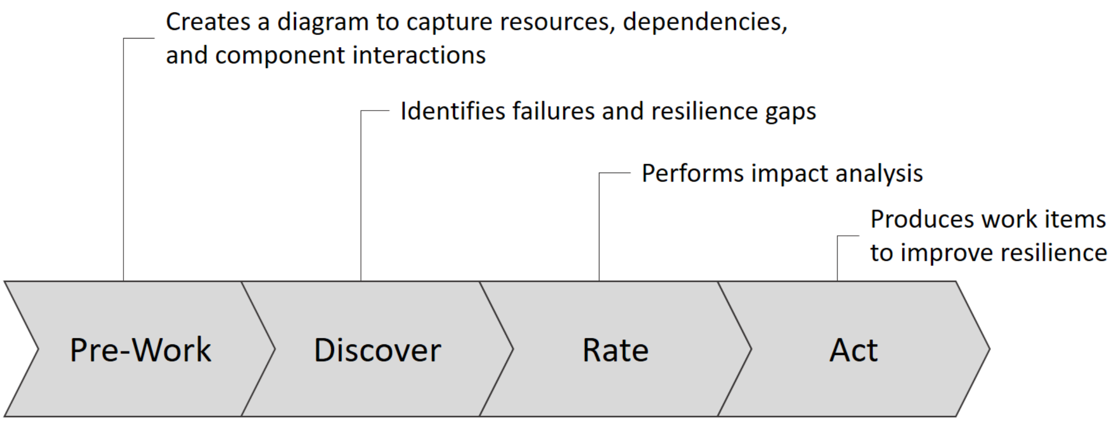

Figure 48: Resilience modeling and analysis (RMA), according to Microsoft Trustworthy Computing group (2013)

Figure 48 provides an overview of the four main phases defined in the RMA approach. The result of the RMA pre-work phase is a detailed architecture diagram, which captures resources, dependencies, and component interactions. For an IoT solution, this would naturally extend the RMA perspective to not only include the cloud architecture but also any kind of edge/fog compute architecture and edge devices, as well as the relationships and interactions between the components in the different layers. Key interactions are covered in an RMA workbook.

The RMA discovery phase identifies potential failures and resilience gaps. This phase analyzes each entry in the RMA workbook (i.e., each component interaction) and adds potential failure scenarios, as well as potential responses to the failure. RMA provides a catalogue of threat categories and root causes, which can be used as a starting point for the detailed discussion.

Next, the RMA rate phase analyzes and records the effects that can result from each of the failure points identified during the RMA discover phase. For each failure point, the following analysis is required:

RMA workbook column	Description	IoT Perspective
Effects	If this failure occurs, how deeply is the system functionality impaired?	For an IoT solution, the potential impact on people in the field relying on the particular functionality must be considered.

Portion Affected	What portion of users or transactions are affected?	Does this apply only to the users of one single IoT asset or device, or does it affect a larger group of users?
Detection	How long does it take until a system or user is notified to take corrective actions?	Does this involve the field support force of the IoT assets/devices?
Resolution	How long does it take to restore the functionality after detection of the failure?	Is physical access by a human operator to the IoT asset/device in the field required?
Likelihood	With which frequency is this failure likely to occur?	Is the frequency dependent on the conditions in the field, e.g., weather?

Table 18: Rating of RMA Workbook entries, including IoT specifics

Finally, the RMA act phase must consider the risk rankings from the RMA rate phase and develop a prioritized roadmap to implement suitable mitigation strategies. The test team can use the entries in the RMA Workbook to define a suitable test strategy.

"Resilience Testing" is the third resilience technique in the list from the previous section. In the blog post entitled "Chaos monkey released into the wild," Bennett and Tseitlin (2012) describe how the video streaming service Netflix started to establish a development model based on the assumption that breakdowns are the norm, rather than the exception. The tooling described in the blog is used by Netflix to randomly take down instances. The important thing is that this is not only done in a test environment but in the real production system (typically during less busy weekdays). Step by step, Netflix has released an entire tool suite dubbed "Simian Army," which is designed to support testing the reliability, security, and resilience of its infrastructure. A new discipline ("Chaos Engineering") seems to be currently emerging in this space (Principles of Chaos Engineering, 2018). Heorhiadi et al. (2016) describe Gremlin, a framework for systematically testing the failure-handling capabilities of microservices by manipulating interservice messages at the network layer. A similar approach could also be applied to resilience testing of components in an IoT solution, especially for messages exchanged between the cloud or enterprise backend and the components in the fog/edge tier.

"System Redundancies and Fail-Over Techniques" are number four on the list of techniques to be considered. For cloud and backend servers, these techniques are well understood today (Infante et al., 2007). For software components deployed in the fog/edge tier, similar approaches can be applied. For example, Microsoft (2014) describes how to enable high availability for edge servers by deploying multiple edge servers in pools in each site. Kim et al. (2017) describe a technique that allows IoT devices to migrate to other local entities when their own entity becomes unavailable, using authentication services as an example. Yun and Nakagawa (2017) provide a comparison between parallel and standby redundant systems.

Finally, “Disaster Recovery Techniques” is the last item on the list of techniques to be considered. Disaster recovery must ensure that IT systems continue to provide their services event after natural disasters (e.g., floods, hurricanes, tornadoes, or earthquakes) or human-induced disasters (infrastructure failure, disastrous IT bugs, or failed change implementations) by building redundancies and recovery capabilities into the systems. As such, it is closely related to the previous technique but usually includes physically redundant backup sites. These can be hot, warm, or cold standby sites (indicating the level of preparedness for taking over operations). ISO/IEC 27031:2011 provides guidelines for information and communication technology readiness for business continuity (which is a superset of disaster recovery—see Sahebjamnia et al. (2015)).

From the IoT perspective, two different aspects must be taken into consideration due to the distributed nature of IoT solutions:

- Disaster strikes in the backend: What impact does a disaster in the backend have on the systems in the field? Even if a backend disaster recovery is executed successfully (e.g., by activating a standby site), how can the IT systems in the field be switched over to work with the new site? How can the systems in the field operate at least partially autonomous during the switchover phase?
- Disaster strikes in the field: How can the system in the field be protected against human impact, either onsite or through malicious attacks from the backend? And how far can system continuity be ensured despite a physical impact, e.g., by hardening the system against environmental factors? Must the systems in the field be equipped with a kind of black box recording, which will work despite a disastrous impact? And how can the backend systems potentially support the identification of assets/IoT devices in the field that are potentially impacted by a disaster and remotely start appropriate support/recovery processes?

3.2.8.3 Result Candidates and Matching Considerations

Table 19 provides an overview of Result Set H, reliability and resilience. This is based on the list of reliability and resilience techniques introduced in the previous sections.

Result Set H: Reliability and Resilience		
#	Result Name	Description and Matching Considerations
H.1	Establish Software Design for Resilience	Requires team training and potentially regular reviews by senior software designers but not necessarily

		dedicated resources to support it. Appropriate for most project types.
H.2	Establish Resilience Modeling and Analysis (RMA)	RMA most likely requires dedicated resources to support it. Appropriate for larger IoT project with a higher level of mission criticality.
H.3	Establish Resilience Testing	Basic resilience testing can be embedded with the test team. Fully fledged resilience testing also within the production system would be a significant investment. Appropriate for larger, mission-critical IoT solutions.
H.4	Establish System Redundancies and Fail-Over Techniques	Significant investment required, both in terms of development resources as well as infrastructure investments. Suitable to highly mission-critical IoT solutions.
H.5	Establish Disaster Recovery Techniques	Implementing disaster recovery beyond what the public cloud vendors are offering out of the box will be a major investment; suitable for very large projects with the highest level of mission criticality.

Table 19: Result Set H—reliability and resilience

3.2.9 I: Verification and Validation

The ninth result set in IgniteWorx is designed to ensure the proper setup of verification and validation during the project initiation.

3.2.9.1 Overview

The objective of verification and validation (V+V in the following) is to ensure the quality of a system, where quality refers to characteristics like functionality, safety, reliability, real-time ability, usability, and reusability (Herrmann, 2001).

Verification is usually seen as addressing the issue of “building the product right”. According to Weck (2009), the following are some of the key aspects of verification:

- Verification is done during development
- The intention is to check if the requirements are met
- For hardware-centric projects, it is typically done in the laboratory
- The focus is on components and subsystems

Validation is usually seen as addressing the issue of “building the right product”. The following are described by Weck (2009) as key aspects of validation:

- Validation is performed during or after the component integration.
- It is typically done in real or simulated mission environments.
- The purpose is to check if the stakeholder intent is met.
- It is done based on the fully integrated and complete system.

Herrmann (2001) differentiates between constructive approaches and analytical measures to support V+V. The goal of constructive approaches is to organize the system development process to minimize quality defects and errors from the very beginning. As an example for an organizational measure, the introduction of the V-Model as the foundation for the development process is cited. Figure 49 shows a proposal for a V-Model with embedded V+V functions, as proposed by WSDOT (2014) and described in more detail in MITRE (2014). As can be seen, the upper parts of the V-Model are assigned to system validation, while the lower parts are assigned to system integration and system verification. This is in line with the above definitions by Weck (2009).

Another key element that is highlighted by Figure 49 is the aspect of traceability. According to Mersky et al. (1993), traceability is the ability to identify the relationships between requirements and the resulting system features. Traceability "facilitates the construction of efficient test plans and permits verification that the resulting test cases have covered the permutations of functional and design requirements/features."

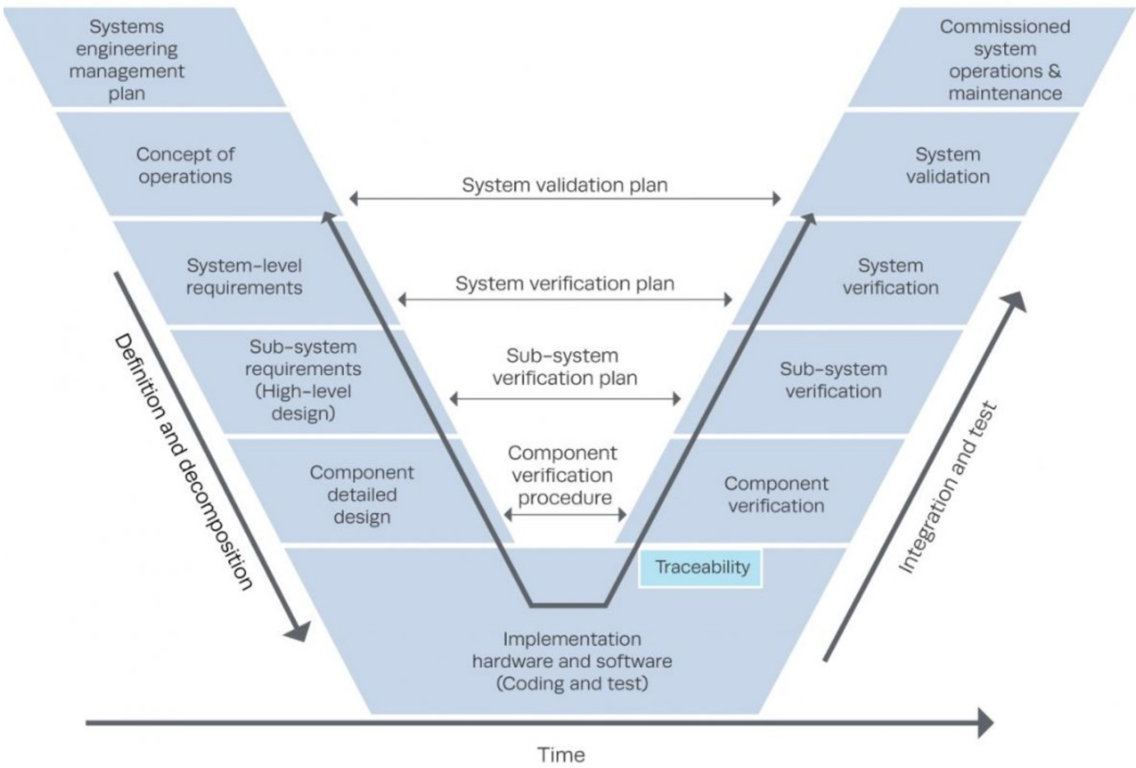

Figure 49: V-Model with verification and validation, according to MITRE (2014)

Testing is one of the important tools in a V+V strategy. According to Herrmann (2001), testing comprises the following main activities:

- Test case determination
- Test data selection
- Expected results prediction
- Test execution
- Monitoring
- Test evaluation

Different types of tests include component testing, integration testing, system testing, and acceptance testing. Bloomfield et al. (2004) differentiates between black box testing (focusing on functional tests) and white box testing (structure-based testing).

In addition to testing, other analytical V+V measures include the following, according to Bloomfield et al. (2004):

- Failure Detection: control flow monitoring, data consistency monitoring
- Static Techniques: inspections, walkthroughs, reviews, and audits
- Usability Validation: expert reviews, usability tests, and follow-up studies based on the installed system
- Efficiency Validation: stress testing, performance testing
- Others: maintainability validation (including regression testing) and portability validation

A number of different guidelines and standards address the topic of creating V+V plans. For example, IEEE Std 1012-1986 specifies the required content for a software verification and validation plan (SVVP). The standard stresses that V+V planning should be thought of as an integral part of project planning. The following steps for V+V planning are defined:

- Definition of V+V scope
- Definition of V+V objectives, including detailed and measurable conditions of satisfaction
- Selection of V+V techniques and tools
- Development of the V+V plan, including
    - V+V organizational structure
    - V+V master schedule, which summarizes the various V+V tasks and their relationships within the overall project environment
    - Required resources
    - Responsibilities
    - Tools, techniques, and methodologies
    - Detailed test plan

Finally, a number of publications stress the importance of ensuring the independence of the V+V resources. For example, Herrmann (2001) points out that test planning and test execution should be done by a qualified and independent team (i.e., not the development team).

According to Mersky et al. (1993), sometimes the V+V effort will be performed by an entirely different organization. This approach is referred to as independent verification and validation (IV+V). Gupta (2018) provides an overview of an elaborate IV+V process, including IV+V planning, review, assessment, and report. An overview of IV+V team roles is also included.

3.2.9.2 IoT Perspective

Many aspects of V+V for IoT are a combination of well-established V+V techniques, combining V+V for traditional enterprise software, wide area networks, and embedded systems. Some of the challenges for IoT V+V include (based on the findings from the previous sections):

- IoT assets and devices will be deployed in the field, with potentially very different environmental factors and different, region-specific regulatory requirements.
- IoT solutions are potentially mission critical.
- IoT solutions are potentially deployed at large scale, making fixes and updates of assets in the field very difficult.
- IoT solutions potentially have a very high level of complexity, especially if they combine many, heterogeneous assets and devices or assets and devices from different manufacturers.
- The solution requirements are potentially unclear and continuously evolving due to this complexity.
- IoT solutions are potentially making use of advanced, not very mature technologies and algorithms.
- IoT solutions are combining technologies from different disciplines, including enterprise technologies, real-time technologies, and advanced data processing technologies.

Under these conditions, different levels of testing will most likely be a key part of any IoT V+V strategy. These can include:

- Functional testing: Testing the basic functionality that is expected from the IoT solution. These functional tests can be complicated if the solution has multiple interaction points. For example, an IoT solution may offer both a web and mobile phone user interface, plus some human-machine interface (HMI) functions embedded directly in the IoT asset of device. Interactions with the one interface can have a direct impact on the other and vice versa.

- Performance testing: This can already be difficult in normal, distributed systems. IoT will most likely add another layer of complexity here, making it even more difficult to identify performance issues within the multiple layers of an IoT system.
- Scalability testing: For an IoT solution, this can mean different things, including testing for large number of users, large number of devices, and high volumes of data intake. Similar to the well-established web test frameworks that can simulate high volumes of web traffic, there are a number of IoT-specific test tools emerging that can simulate, for example, large numbers of IoT assets or devices and their physical movements.
- Field tests: Many conditions that can be found in the field will be difficult to simulate efficiently. These can be environmental conditions but also technical, IoT-specific issues. For example, Slama, Puhlmann et al. (2015) describe problems with too many IoT devices reconnecting automatically after a server failure.
- Device compatibility tests: In case of IoT solutions that support large numbers of devices from different vendors in different versions, this can be particularly difficult. Take, for example, a smart home solution that supports devices from multiple vendors. At least in theory, each update of a supported device must be tested against all versions of the smart home controller and vice versa since there is no guarantee that all customers will always have upgraded to the latest versions in their homes.
- Sensor reliability testing: This must include testing of sensor performance in different environmental situations.
- Security testing: See section 3.2.7.
- Reliability and resilience testing: See section 3.2.8.
- Testing in the production systems: As discussed in section 3.2.8, some Internet companies are constantly running tests in their production systems, simulating different kinds of failure situations in the real production system. For large-scale, mission-critical IoT systems, a similar approach should be considered.

In some IoT projects, there might also be need for more specialized V+V methods. Some examples include:

- In the automotive industry, specialized test techniques including hardware-in-the-loop (HIL), software-in-the-loop (SIL), and model-in-the-loop (MIL) are used to maximize test coverage during the development phases and minimize the more costly detection of failures during field tests (Jaikamal, 2009).
- For systems using artificial intelligence, new V+V methods are emerging. For example, Ramachandran (2005) describes strategies for V+V in neural network-based autonomous systems. This work addresses difficulties involved with

proving that a neural network has learned all scenarios provided to it or how the network would react to unfamiliar data that was not part of the training phase. This is especially relevant for the automotive industry, which is looking at emerging V+V concepts like safety of the intended functionality (SOTIF). These concepts are addressing AI-related issues which are not sufficiently covered by established V+V standards like ISO 26262/ASIL (automotive safety integration level). See Griessnig and Schnellbach (2017) for details.

- For embedded hardware (e.g., ASICs or SOCs), specialized V+V procedures can differentiate between design time verification (before tape-out) and post-manufacturing verification (Mozhikunnath, 2016).
- Some IoT systems are relying on localization data. Testing localization-based functions can also create significant test complexity because the system functionality can potentially depend heavily on environmental conditions (e.g., different metallic shields in a manufacturing site, etc.)

Finally, certification can be an important part of the V+V process. Klubnikin (2016) defines the following categories for certificates required for IoT devices:

- Environment and electrical safety includes, for example, the Restriction of Hazardous Substances Directive and Energy Star compliance, as well as tests for issues like overheating and electric shock.
- Communication protocols: For example, a Bluetooth device must be validated by a certified lab before it can officially claim that it is Bluetooth-compatible.
- Electromagnetic and radio-frequency interference requires proving the device does not interfere with other devices and that it conforms to the electromagnetic radiation exposure standards.

Klubnikin (2016) also provides the following examples for typical certificates required by IoT devices:

- FCC certification: required for electrical devices sold in the United States. Can potentially be more expensive for devices using wireless connectivity.
- UL/CSA certification: required for electrical devices that plug into electrical outlets (Canada and United States)
- CE certification: the European Union analogue of FCC and UL
- RoHS certification: confirms that a product does not contain lead (EU and United States)

3.2.9.3 Result Candidates and Matching Considerations

Table 20 provides an overview of Result Set I, verification and validation. This is based on the V+V considerations discussed in the previous sections.

Result Set I: Verification and Validation

#	Result Name	Description and Matching Considerations
I.1	Informal V+V	Suitable for small and not mission-critical IoT projects. V+V will mainly focus on ad-hoc tests performed by the development team.
I.2	Formal V+V	Independent V+V team, following clearly defined validation master plan, creating formal management report with all findings. Required by IoT solutions with regulatory requirements.
I.3	V+V-centric PMM	Like I.2, but integrated into a project management methodology like V-Model which is explicitly designed to support V+V. Required by mission-critical IoT solutions.
I.4	IV+V	V+V process is run by external provider. Suitable for missing critical systems that require this for approval.
I.5	Dedicated Certification Team	Dedicated team to support all certification requirements. Will probably be required by most commercial IoT solutions, especially if sold in multiple countries.

Table 20: Result Set I—verification and validation

3.2.10J: Service Operations

The tenth and last result set in IgniteWorx is designed to help ensure that after the launch of the IoT solution, a suitable service operations organization will be available.

3.2.10.1 Overview

Extensive literature and Internet research has shown that there is currently no widely accepted and well-defined concept for IoT service operation. However, a number of established or emerging concepts seem to be helpful in this context:

- IT Service Management (ITSM): A set of practices to design, plan, deliver, operate, and manage IT services in an organization. ITIL is a well-established standard in this space (AXELOS, 2019).
- Field Service Management (FSM): Supports the management of service technicians working in the field to support the assets of a company (Gartner Group, 2019).

- Asset Performance Management (APM): An emerging concept that aims to combine the more financial-oriented side of enterprise asset management (EAM) with real-time asset data analytics enabled by the IoT (Miklovic, 2015).

The following section looks at each of these three concepts in more detail before discussing the IoT-specifics.

For ITSM, the discussion is based on the ITIL framework, which is owned by Axelos, a joint venture between the UK government and Capita. ITIL is structured as five core books to cover the full-service lifecycle: service strategy, service design, service transition, service operation, and continual service improvement (Gartner Group, 2019). Figure 50 provides an overview of the ITIL lifecycle for ITSM (BMC, 2016).

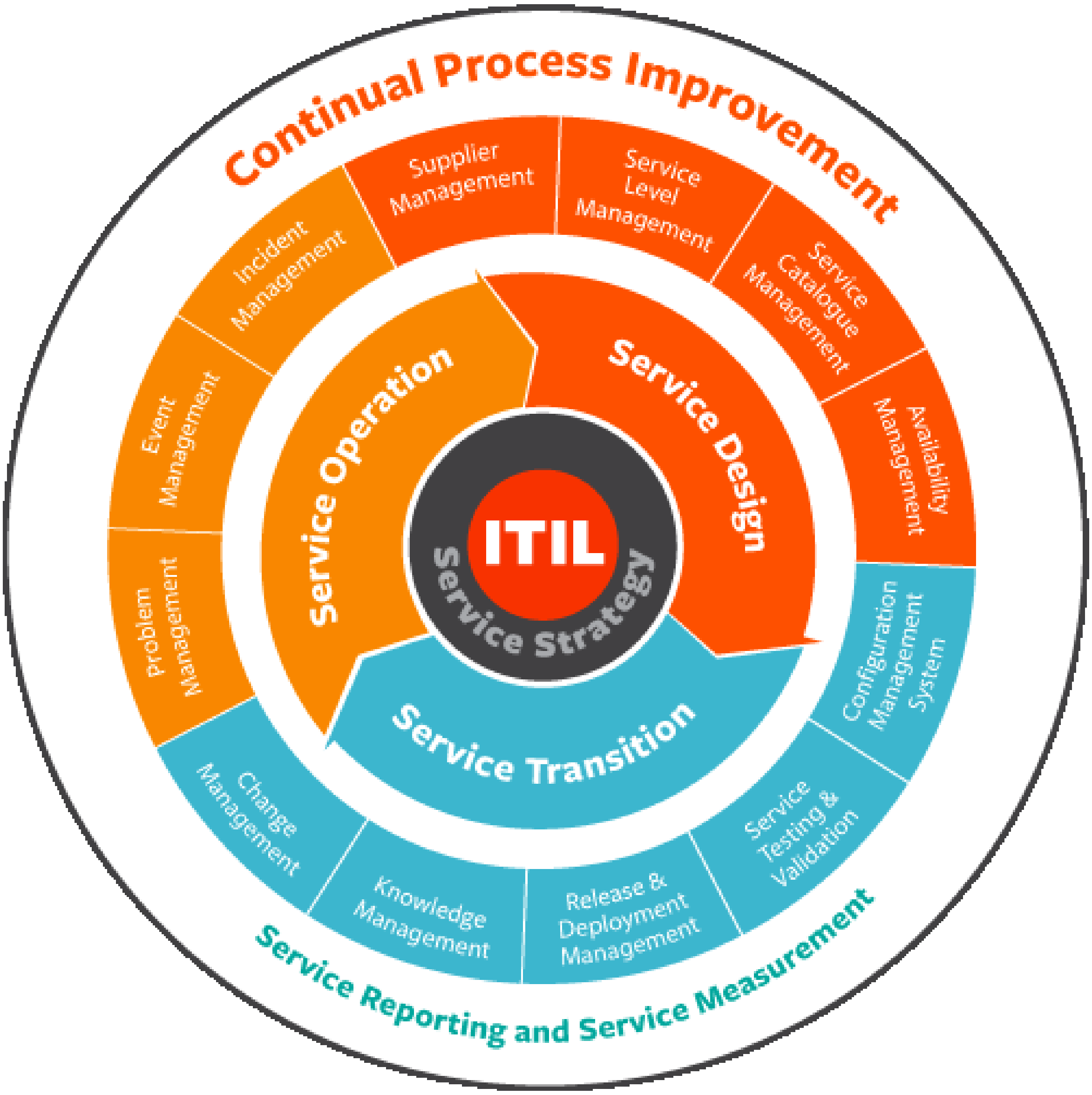

Figure 50: ITIL lifecycle according to BMC (2016)

ITIL is a mature and comprehensive framework that consists of processes, procedures, tasks, and checklists. The five main elements can be described as follows (AXELOS, 2019):

- ITIL Service Strategy: definition of organizational objectives, based on customer needs
- ITIL Service Design: creation of the plan for implementing the service strategy
- ITIL Service Transition: development/improvement of the required capabilities for the introduction of new services
- ITIL Service Operation: management of the provided services
- ITIL Continual Service Improvement: focus on large-scale improvements

The service operations side deals with the day-to-day operations of the IT service organization. ITIL defines five processes and four functions in this context. The four functions are service desk, technical management, application management, and IT operations management. The five service operations processes are (Brahmachary, 2018):

- Access Management: grants authorized users the right to use a service; blocks any access request of nonauthorized users to the service
- Event Management: captures, filters, and categorizes events to decide the appropriate actions to be taken. Events might or might not require an action.
- Incident Management: Incidents are events that have a negative impact on a service or its quality. Incident management helps restore the IT service to working state as quickly as possible.
- Problem Management: deals with identifying and addressing problems at their root. Multiple incidents can relate to the same problem.
- Request Fulfilment: responsible for acknowledging and processing service requests received from users. Usually, these are technical requests, not requests related to the functionality of business applications.

To manage all IT assets and other related data, ITIL foresees the use of a so-called configuration management database (CMDB) as the central repository for this kind of information. However, the complexity of introducing a CMDB should not be underestimated. Rouse (2017) warns that CMDB projects often fail due to stale and unusable data.

While ITSM is focusing on IT-related assets, field service management (FSM) is focusing on enterprise assets, e.g. operational equipment, machines and vehicles. FSM is described by (Gartner Group, 2019) as a practice that “includes the detection of a field service need (through remote monitoring or other means, inspection or a customer detecting a fault), field technician scheduling and optimization, dispatching, parts information delivery to the field, and process support of field technician interactions.”

The market for FSM solutions seems mature, with multiple large market players in it. The typical features of an FSM application are described as follows (Wong):

- Creation of work orders from cases
- Management and monitoring of technicians
- Scheduling and order management
- Vehicle/technician location tracking
- Job status updates
- Route optimization and GPS navigation
- Time tracking and driver logs
- Knowledge and asset repositories
- Parts and inventory management
- Integrated invoicing/payment processing
- Customer portal access
- Regulatory compliance measures

Finally, asset performance management (APM) is focusing on assets in the sense of the assets typically listed in the balance sheet of a company. According to (Gartner Group, 2019), APM encompasses “the capabilities of data capture, integration, visualization and analytics tied together for the explicit purpose of improving the reliability and availability of physical assets”(Gartner Group, 2019) and also explicitly mentions the concepts of condition monitoring and predictive forecasting.

Miklovic (2015) describes APM as a practice that enables enterprises to look at assets beyond their market value or the current depreciated value by looking at the individual performance of an asset in real time. This in turn enables enterprises to decide how they should allocate resources to the asset in the future. APM is described as crossing functional lines by combining the financial and the operational perspectives of an asset.

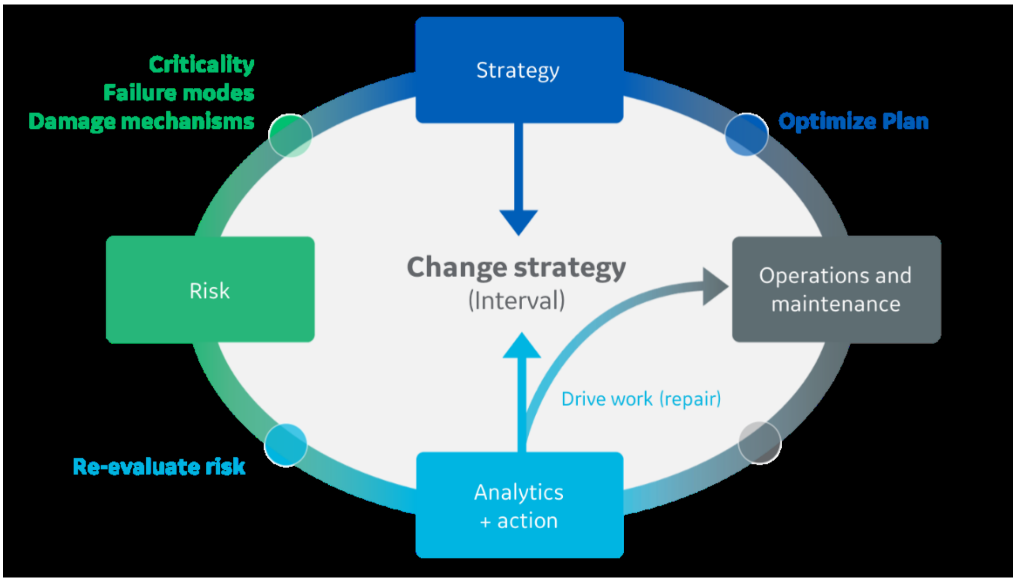

Figure 51: APM lifecycle according to GE (Bailey, 2019)

APM as a new concept was initially popularized by GE, with a strong focus on IIoT-based support for the real-time data analytics part of APM (Bailey, 2019). Figure 51 describes the APM lifecycle according to GE. The figure shows the relationship among strategy, risk, analytics, and actions, as well as operations and maintenance. Bailey (2019) describes APM as a technology that helps answer the following questions:

- How critical is the asset?
- What is the history of the asset, and what is its current health?
- In what ways can this asset fail, how can the risk of failure be mitigated, and at what cost?
- What would be the consequences of the asset's failure on the business?
- What action should be taken now to prevent failure?
- What should be the overall strategy for the asset to optimize business objectives?

The following looks at how IoT is enabling new concepts like APM, as well as changing existing concepts like FSM, by providing a real-time link to enterprise assets. Furthermore, the need for ITSM in IoT solutions is analyzed.

3.2.10.2 IoT Perspective

In the pre-IoT world, one of the biggest problems of most enterprise asset and resource management solutions was the disconnect between data in the repositories and the reality in the physical world (Xu et al., 2002). IoT is changing this by providing real-time insights into the status and performance of assets. Based on input from Martin (2015) and other Internet sources, the following is an overview of areas in FSM where IoT can potentially have a big impact:

- Improved triage: IoT-data can be used to determine the severity and priority of asset-related incidents.

- Faster identification of required parts: Use RFID data for precise identification of assets and key parts deployed in the field.
- Inventory tracking: Use RFID data for creating a precise and real-time inventory update.
- Initiation of automated intelligent dispatch events: Use IoT sensor data to better prioritize incidents and to provide more information for problem resolution.
- Remote monitoring and diagnostics: Use real-time machine data for asset health and performance assessments.

All of this will only be possible if the IoT project is preparing the service operations organization accordingly. This will be one of the big challenges of the IoT project management team. How to do this will depend strongly on a number of different factors, including:

- Is there already an existing organization responsible for FSM?
- If so, how is the organizational relationship between the IoT solution project and the existing FSM organization?
- If not, how far is the IoT solution project empowered to actually set up a new FSM organization to start operating after the start of production?
- Will the focus be mainly on operational FSM topics, or will it also include more strategic topics as covered by the APM perspective?

The analysis in the previous section shows that the FSM and APM perspectives potentially have a significant overlap and will be difficult to fully separate. Consequently, the following discussion subsumes these two perspectives as FSM/APM.

ITSM with a focus on IoT will be referred to as IoT-ITSM in the following discussion. IoT-ITSM relates to all processes related to operating the IoT-solution on the IT level, e.g., ensuring that the on-asset hardware is operating properly, ensuring connectivity to the backend, and keeping the backend applications and databases physically operational. However, it cannot be assumed automatically that the IoT-ITSM operation will have domain knowhow and support capabilities that would typically come from FSM/APM. After all, the skills required to deal with the IP configuration of an IoT gateway or to keep a time series database running are very different than, for example, the skills required to analyze and repair the malfunction of an excavator hydraulic component. Consequently, the IoT project must make a deliberate decision early in the IoT project setup on how to organize IoT-ITSM and FSM/APM.

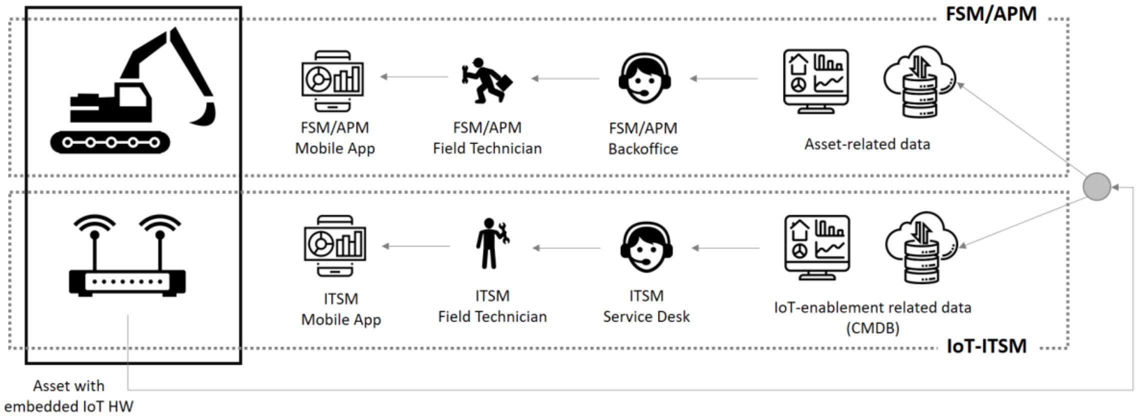

Figure 52: FSM/APM and IoT-ITSM as two separate organizations

Figure 53 shows an example for setting up FSM/APM and IoT-ITSM as two separate organizations. As an example, a simplified monitoring solution for excavators is shown, using some form of gateway or TCU on the excavator. Both the FSM/APM application and the IoT-ITSM application have their own databases, receiving data from the gateway/TCU. The IoT-ITSM solution is using some form of CMDB to store information related to the configuration items that make up the IoT solution (e.g., an inventory of gateways in the field, with related incidents). The FSM/APM solution stores asset-related data from the same source, e.g., performance data from the hydraulics component of the excavator. Both solutions then have their dedicated and specialized staff, which is supporting their respective services.

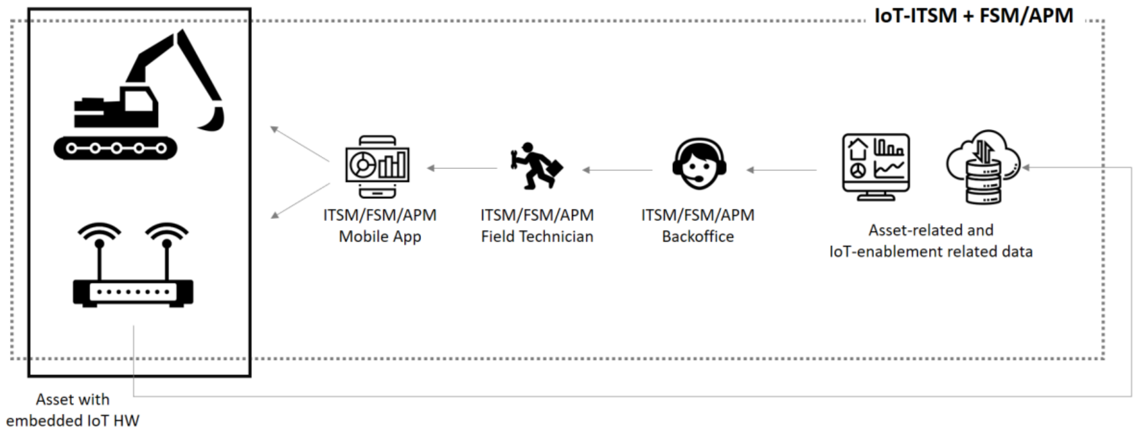

Figure 53: IoT-ITSM and FSM/APM as integrated organization

Figure 53 shows an example of an integrated IoT-ITSM + FSM/APM solution. In this case, only one repository is used, which stores both, asset-related and IoT-enablement related data. The back office is supporting all functions, and so is the field service. Of course, these are only two examples of a potential organizational setup; in reality, many other, potentially hybrid combinations could be possible. However, the examples serve the purpose of highlighting the issue and the choices an IoT project manager must make.

Table 21 discusses examples for required services in IoT-ITSM as well as FSM/APM, using ITIL concepts to help structure the discussion.

ITIL Areas and Processes	IoT-ITSM Examples	FSM/APM Examples
Service Design		
Service-Level Management	SLA for completion time of FOTA updates to entire fleet of assets in the field	SLA for refresh rate of remote asset performance data
Availability Management	Recovery procedure after failed FOTA update	Availability of fleetwide asset performance analytics service
Service Transition		
Knowledge Management	Procedure for configuring IP connectivity of IoT gateway	Procedure for replacing hydraulics components
Release and Deployment Management	Upgrade of time series database server version	Upgrade of business logic for fleet performance analytics application
Service Testing and Validation	Testing of FOTA capabilities	Testing of accuracy of fleet performance analytics algorithms
Configuration Management System	Configuration of VPN for IoT gateways	Physical configuration of excavator hydraulics components
Service Operation		
Event Management	Successful backup of time series database	Availability of new daily time series analytics report (batch-run complete)
Incident Management	Single excavator IoT gateway cannot connect to backend	Breakdown of individual excavator hydraulics component
Problem Management	Backend not available due to server downtime—excavator fleet offline	Fleetwide product recall due to faulty hydraulics components

Table 21: Examples of IoT-ITSM and FSM/APM services, based on ITIL

The examples in Table 21 are not complete, but they are a useful starting point to discuss different aspects that IoT-ITSM and FSM/APM will have to cover. Furthermore, it shows how important a clear understanding of the delineation of these two different concepts is. Finally, it shows that ITIL might well be suitable as a framework to not only manage IoT-ITSM but also FSM/APM, even if they are treated separately.

3.2.10.3 Result Candidates and Matching Considerations

Table 22 shows the results for Result Set J—service operations. In this case, it might be difficult to derive concrete recommendations, so this result set can also serve to initiate a structured discussion on the best way forward for the project.

Result Set J: Service Operations		
#	Result Name	Description and Matching Considerations
J.1	Brownfield: separate IoT-ITSM and FSM/APM solutions	IoT-ITSM will be integrated with an existing ITSM organization, probably using an existing CMDB and up-skilling existing ITSM staff. Similar for FSM/APM, most likely an existing FSM application will be extended to receive and process real-time IoT data, also adding APM capabilities.
J.2	Greenfield: Integrated IoT-ITSM and FSM/APM solution	As depicted in Figure 53, this approach integrates IoT-ITSM and FSM/APM functions in a single application and organization. Most likely only applicable in greenfield-type of situations.
J.3	Hybrid	In the backend, specialized applications/repositories for IoT-ITSM and FSM/APM are used (e.g., CMDB for IoT-ITSM, as well as COTS FSM application). However, service desk/FSM backend and field force are still one organization.
J.4	Fully automated, integrated IoT-ITSM + FSM/APM	All customer support services are fully automated, e.g., by using extensive web-based FAQs and automatic email analytics and auto-response tools for all customer interactions. No call center and field services provided. Requires extremely high level of automation of the IoT solution, both for the IoT-ITSM side as well as the FSM/APM side. Prerequisite for highly scalable solutions with many assets in the field.

Table 22: Result Set J—service operations

3.3 System Architecture

Based on the requirements laid out in the previous sections, IgniteWorx now needs a proposal for the supporting system architecture. To define a suitable architecture, a set of architecture evaluation criteria is examined before looking at the state of science for potential support system categories, as well as related concepts and algorithms. Based on this, an architecture proposal is derived.

3.3.1 Architecture Evaluation Criteria

Evaluating the suitability of available systems and architectures for IgniteWorx requires the definition of a set of architecture evaluation criteria. On the one hand, this is a difficult topic because there are many different potential definitions of what a system architecture actually comprises. On the other hand, there are well-defined software architecture evaluation frameworks available, e.g., the architecture evaluation frameworks developed by the Software Engineering Institute (SEI) (Clements et al., 2002):

- ATAM: Architecture Tradeoff Analysis Method
- SAAM: Software Architecture Analysis Method
- ARID: Active Reviews for Intermediate Designs

ATAM evaluates an architecture for suitability by focusing on the following quality attributes: performance, reliability, availability, security, modifiability, portability, functionality, variability, subsetability, and conceptual integrity. SAAM is focusing on modifiability, e.g., portability, subsetability, and variability, as well as functionality. ARID is focusing on analysis of the suitability of different elements of an architecture from a developer's point of view.

Of course, there are also many other approaches. For example, Mattsson et al. (2006) focus on software architecture evaluation methods for performance, maintainability, testability, and portability. Jackson et al. (2011) focus on usability, sustainability, and maintainability. Bouwers et al. (2009) look at high-level design, modularization, and separation of concerns.

For the purpose of this thesis, ATAM has been identified as the most suitable approach since it seems well established and covers a broad range of pragmatic evaluation criteria. It is applied to the evaluation of a suitable architecture for IgniteWorx as follows:

- Performance: IgniteWorx must support the expected quality of service of a non-mission-critical, web-based application. This means, for example, that response times should be such that users can use the system efficiently in real time. Collaboration by up to twenty-five concurrent users should be supported.

- Reliability: The main aspect of reliability is that project assessments should never be corrupted by the system. This is important because users might put significant work hours into creating an assessment.
- Availability: Regular downtimes for maintenance are acceptable, but they should be announced in advance.
- Security: An IgniteWorx implementation must ensure that IoT project assessments can only be accessed by the author or explicitly authorized users.
- Modifiability: An IgniteWorx system must ensure that all content—including project dimensions, result sets, and rules—can be edited using a web-based online editor, without requiring any coding. It is acceptable for the actual matching algorithm to be hard coded using a suitable programming language.
- Portability: It must be ensured that an operator of IgniteWorx can provide a suitable execution environment for the implementation.
- Functionality: The functionality outlined in sections 2, 3.1, and 3.2 must be fully supported.
- Variability: The architecture will have to support the anticipated evolution of the system from manually provided content and rules to more automated recommendations, as described in section 2.7.
- Subsetability: The main requirement here is that IgniteWorx should focus on project dimensions, result sets, and rules as loosely coupled domains. In particular, reuse of the existing concepts for Ignite dimensions is a key requirement (see section 2.4).
- Conceptual integrity: The architecture must have an underlying concept for unifying the design of the system on all levels.

These evaluation criteria are first used to evaluate the results of the architecture design phase (see evaluation in section 3.3.6). Secondly, they will be used again for an evaluation of a concrete IgniteWorx implementation, based on the results of both research iterations of this thesis (see evaluation in section 9.6.1).

3.3.2 Related System Categories—State of Science

There are a number of different system categories described by state-of-the art research that could potentially support the requirements for an IgniteWorx implementation. In the following, the state of science regarding decision support systems, expert systems, recommender systems, semantic reasoning systems, and survey engines is described before analyzing their suitability for IgniteWorx.

3.3.2.1 Decision Support Systems

Decision support systems (DSS) support decision makers by aggregating and presenting relevant information from a variety of data sources. Spraque (1980) provided one of the first comprehensive description of a DSS: "DSS tends to be aimed at the less well structured, underspecified problem that upper level managers typically face; DSS attempts to combine the use of models or analytic techniques with traditional data access and retrieval functions; DSS specifically focuses on features which make them easy to use by non-computer-proficient people in an interactive mode; and DSS emphasizes flexibility and adaptability to accommodate changes in the environment and the decision making approach of the user."

DSS typically support different types of decision analysis (DA) methods. From an IgniteWorx point of view, one interesting branch of DA methods supported by some DSS are multi-criteria decision analysis (MCDA) methods, which are described in 3.3.3.5.

Holsapple (2000) defines the following five basic DSS approaches as:

- Text-oriented DSS
- Database-oriented DSS
- Spreadsheet-oriented DSS
- Solver-oriented DSS
- Rule-oriented DSS

He also introduces the compound DSS, a hybrid system that combines the characteristics of two or more of the above basic DSS categories.

Burstein and Holsapple (2008) also introduce the intelligent decision support systems (IDSS), which is performing selected cognitive decision-making functions by using artificial intelligence or intelligent agents technologies. Lasi (2012) introduces "Decision Support within Knowledge-based Engineering."

Holsapple (2000) defines the main components of a DSS as follows:

- Inputs: Factors, numbers, and characteristics to analyze
- User knowledge and expertise: Inputs requiring manual analysis by the user
- Outputs: Transformed data from which DSS "decisions" are generated
- Decisions: Results generated by the DSS based on user criteria

It should be noted that some of the research related to DSS seems slightly outdated, e.g., based on the search results from Google Scholar. Arnott and Pervan (2005) state "The analysis of the professional or practical contribution of DSS research shows a field that is facing a crisis of relevance."

3.3.2.2 Expert Systems

In the 1970s, availability of computers with larger memory modules enabled the evolution of artificial intelligence toward knowledge-based expert systems, pioneered by Edward Feigenbaum (Feigenbaum, 1977). An expert system typically combines an inference engine and a knowledge base. The knowledge base represents known facts and rules. The inference engine applies the rules to the facts to derive new facts.

Ogu and Y.A. (2013) provide a good overview of typical expert system architectures, including the required components for user interaction. Figure 54 provides a generalized overview. The so-called expert system shell includes the interface for the end-user, as well as a knowledge base editor for the knowledge engineer, in addition to the inference engine and an explanation system. All these components access a shared repository with the knowledge base and rule base.

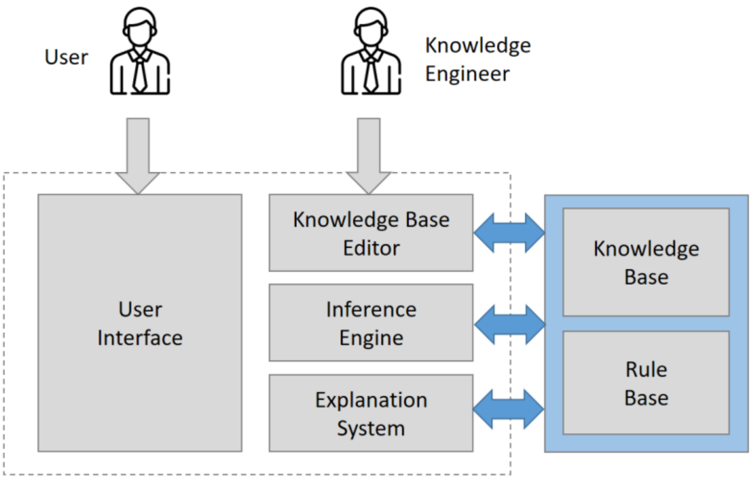

Figure 54:Architecture of an expert system

Expert system shells can also support the question/answer paradigm inherent to the Ignite project dimensions used for the assessment of individual IIoT projects. For example, Sosnin (2011) describes a question-answer shell for personal expert systems. Another example is Choi (2002), who presents a rule-based expert system using an interactive question-and-answer sequence.

3.3.2.3 Recommender Systems

Ricci et al. (2015b) define a recommender system (RS) as "software tools and techniques that provide suggestions for items that are most likely of interest to a particular user." Popular examples of the output or RS are Facebook's "People You Might Know," Netflix's "Other Movies You Might Enjoy," and Amazon's "Customers Who Bought This Item Also Bought…"

Dhillon (2015) defines six classes for RS: collaborative recommender system, content-based recommender system, demographic-based recommender system, utility-based recommender system, knowledge-based recommender system, and hybrid recommender system:

- Collaborative recommender systems "aggregate ratings or recommendations of objects, recognize commonalities between the users on the basis of their ratings, and generate new recommendations based on inter-user comparisons." Algorithms can include graph-based algorithms (e.g., based on neighborhood) or latent factor model (Liang, 2012).

- A content-based recommender system "learns a profile of the new user's interests based on the features present, in objects the user has rated."
- Demographic-based recommender systems make recommendations based on demographic classes, but "the algorithms first need a proper market research in the specified region accompanied with a short survey to gather data for categorization."
- Utility-based recommender systems make suggestions "based on computation of the utility of each object for the user." For example, product availability can be factored into the computation.
- Knowledge-based recommender systems "suggest objects based on inferences about a user's needs and preferences."
- Hybrid recommender systems combine two or more of the above RS categories to improve recommendation results.

Liang (2012) describes the main architectural elements of the RS developed by Hulu, the video streaming company. The system consists of an online and an offline part. The online part includes five main modules: user profile builder (includes the user's historical behaviors), recommendation core (generates raw recommendations), filtering (filter for raw recommendation results, e.g., on past user behavior), ranking (e.g., to increase diversity) and explanation (why was a recommendation made?). The offline system includes data center (including all user behavior), related table generator (for collaborative filtering and content filtering), topic model (a group of shows that have similar content), feedback analyzer (users' reactions to recommendation results), and report generator (e.g., click-through rates and conversion rates). This example shows the high level of complexity inherent to RS implementations.

During the research on this topic, two common problems in RS emerged that also have a high relevance for the design of IgniteWorx: the so-called cold start problem and the product complexity problem.

The term "cold start" was chosen as an analogy to a car engine, which only runs on the optimal level once the engine has warmed up. For RS, this means that the system has to acquire a sufficient amount of metadata before it can make efficient recommendations. There are two main issues described in the literature: product cold start and visitor cold start (see, for example, Volkovs et al. (2017) or Nadimi-Shahraki and Bahadorpour (2014)). The product cold start relates to new products that have no metadata available in the system, like user reviews or any other kind of "likes" from a certain group of users. The user cold start relates to new users who have no history in the system that can be used to derive preferences.

Gope and Jain (2017) provide a survey on solving cold start problem in recommender systems. They differentiate between explicit and implicit solutions. Techniques for explicit solutions include active learning and interviews. Nadimi-Shahraki and Bahadorpour (2014) describe an ask-to-rate technique, in which a new user is asked to rate the selected items until having a sufficient number of rated items. Implicit solutions include adapted filtering strategies and external data collections, e.g., through social networks.

The cold start problem can also be applied to IgniteWorx, using the following mapping:

- User cold start: The "user" in Ignite is an IIoT project. Ignite dimensions are used to implement a kind of "ask-to-rate" strategy, as described by Nadimi-Shahraki and Bahadorpour (2014).
- Product cold start: The products in IgniteWorx are the elements of the result sets. IgniteWorx rules are used to explicitly map relevant "products" to "users" (or, more specifically, IIoT projects to result set elements).

The second RS problem area with a high relevance for IgniteWorx is described by Ricci et al. (2015b): Most RS are designed today to recommend items with a relatively simple structure, e.g. music, movies, books. More complex item types, such as financial investments or travel, are considered to be atypical cases for current RS. Most current RS are designed to treat different configurations as different items. According to Ricci et al. (2015a), "Complex products are typically configurable or offered in several variants. This feature still poses a challenge to recommender systems, which are instead designed to consider different configurations as different items. Identifying the more suitable configuration requires reasoning between the interactions of alternative configurations (classifying and grouping items) and calls for addressing the specificity of the human decision making task generated by the selection of a configuration."

Felfernig et al. (2015) state, "Knowledge-based recommender technologies help to tackle these challenges by exploiting explicit user requirements and deep knowledge about the underlying product domain for the computation of recommendations."

Assuming that the product domain in IgniteWorx is the recommended IIoT project setup, then the solution proposed here is to decomposition the "product" into multiple results sets, according to section 3.1.2.5.

3.3.2.4 Semantic Reasoning Systems

Building on the success of the early World Wide Web, Berners-Lee et al. (2001) introduced the concept of the sematic web, describing how ontologies improve how knowledge can be captured and made more easily accessible. At the core of these ontologies are taxonomies and inference rules. Taxonomies define classes of objects and relations among them. Inference rules enable automated programs to deduce conclusions from the available data, enabling a truly intelligent Internet.

A key technology for adding intelligence to a knowledge base are semantic reasoners (SRs), which are designed to infer logical consequences from a set of asserted facts. At the core of an SR is an inference engine, which is processing inference rules. These rules are usually specified by using an ontology language, such as W3C's Ontology Web Language (OWL); see also McGuinness et al. (2004).

OWL supports most of the key components of an ontology, including classes (as a way of abstraction) and individuals (concrete instances of classes), attributes and properties, and rules and axioms.

Applying an ontology language such as OWL to IgniteWorx, one could describe the different elements from IgniteWorx using the language specific syntax. Take, for example, Ignite dimensions: Each Ignite dimension provides multiple options. Dimensions are grouped into categories. Using OWL abstract syntax, this could look as follows:

```
Namespace(iwx = <http://enterprise-iot.org/IgniteWorx.owl#>)
Ontology( <http://enterprise-iot.org/IgniteWorx.owl#>
 Class(iwx:Category)
 Class(iwx:Dimension partial restriction(partOf someValuesFrom(Category))
 Class(iwx:Option partial restriction(partOf someValuesFrom(Dimension))
 DatatypeProperty (ex:name)
 ObjectProperty(ex:has)
 Individual(type(ex:Category)
 value(ex:has Individual(type(ex:Dimension)
 value(ex:name "Number of Assets"^^xsd:string))))
)
```

In this example, three classes are defined: category, dimension, and option. Dimensions are part of categories, and options are part of dimensions. Next, an instance of a category is created, with a child element dimension called number of assets.

This is, of course, only a simple example. The next step would be to also model concepts such as IIoT project assessment and result sets using OWL. OWL would then also be used to model the relationships among Ignite dimensions, IIoT project assessments, and result sets, using rules.

Potential advantages of using established standard such as OWL include interoperability, as well as the availability of commercially available tools.

Finally, Blomqvist (2014) provides a survey on the use of semantic web technologies for decision support systems, concluding, “Semantic Web technologies can help to solve basic DSS needs such as information interoperability, integration and linking, while additionally potentially continuing to support the development of ‘Intelligent DSS’, but in a new and more open manner than what was traditionally possible with AI technologies.” This should be interesting, given that decision support is also a concept related to many IgniteWorx requirements.

3.3.2.5 Survey Engines

Given that a key part of IgniteWorx is the user friendly capturing of detailed IIoT project assessments, survey engines are also considered as candidates for an IgniteWorx implementation.

Survey Engines are embedded into many modern web services or are even available as standalone services, such as Survey Monkey (Waclawski, 2012). There are also a number of patents defined that deal with reusable online survey engines (Kirkpatrick et al., 2007) or web survey tool builder and result compilers (Fuerst, 2001).

Schlereth and Skiera (2012) describe a dynamic intelligent survey engine (DISE) for capturing user preferences. The requirements for DISE are as follows:

- Broad support for different web-based data collection methods, including number, text, radio buttons, or spectrums
- Definition of quotas for the sampling of respondents
- Multilingual user interface
- Ability to conditionally show questions depending on previous responses
- Ability to create different versions of a survey and to assign respondents randomly to one of the versions
- Ability to integrate survey panel providers

The DISE architecture as shown in Figure 55 includes a communication layer, a process layer, an execution layer, an information layer, and a third-party integration layer. Vertical services include survey construction, data elicitation, data analysis, and conceptual representation.

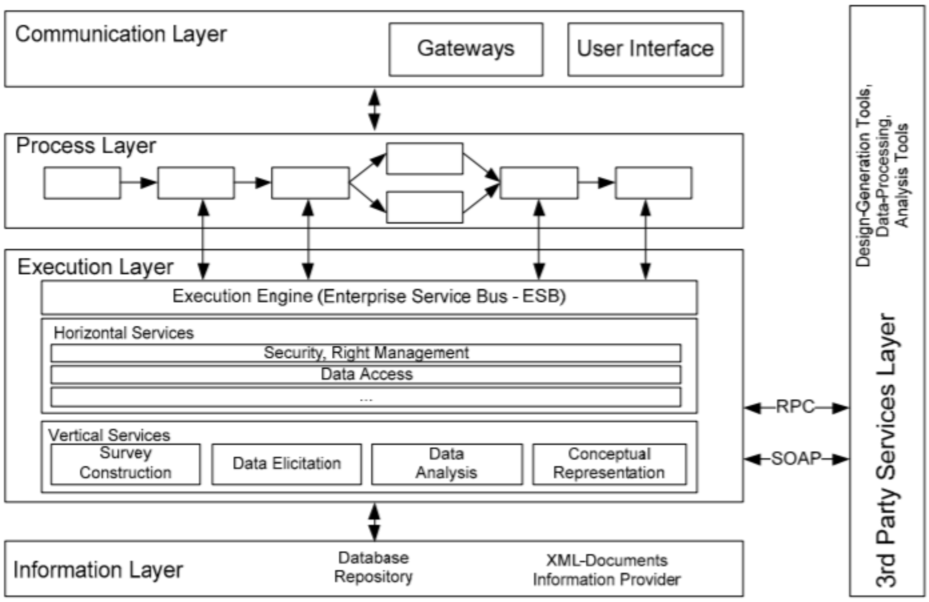

Figure 55: DISE architecture (Schlereth and Skiera, 2012)

The IgniteWorx solution will at least have to support all vertical services describes in DISE to support the user friendly capturing and analysis of IIoT project assessments.

3.3.2.6 Evaluation for IgniteWorx

In the following, a tabular evaluation of the different system categories that have been considered for IgniteWorx is provided.

System Category	Evaluation
Decision Support Systems	The main goal of IgniteWorx is to support IIoT project managers in making the right decisions about their individual project setup. IgniteWorx must support project managers in making some important decisions. However, IgniteWorx is also aiming to provide as much structure around this process as possible; this is what the main artifacts of IgniteWorx are about: dimensions, results sets, and rules. Since DS systems tend to be aimed at less structured, underspecified problems (Spraque, 1980), they are unlikely to be suitable for an IgniteWorx implementation.
Expert Systems	The combination of explicit knowledge base (=> IgniteWorx result sets), rule base (=> IgniteWorx rules), inference engine, and explanation system makes expert systems a very interesting candidate as a blueprint for IgniteWorx. Also, the described user interfaces are a good fit (end-user UI and knowledge editor).

Recommender Systems	The web-centric roots of recommender systems make them attractive for IgniteWorx. Some of the problems that the RS community is dealing with can also be found in the IgniteWorx approach, especially the cold start problem (Gope and Jain, 2017), which seems directly applicable to the recommendations in the IgniteWorx result sets.
Semantic Reasoning Systems	The use of ontologies and open standards such as OWL could potentially be attractive for an IgniteWorx implementation, provided that they allow for easy integration with a web-centric and easy-to-use user interface.
Survey Engines	Most of the requirements described by the DISE example for a flexible survey engine can also be mapped to IgniteWorx, especially for the construction of the IIoT project assessment tool (Schlereth and Skiera, 2012). This means this tool category should also be considered, at least for the assessment module of IgniteWorx.

Table 23: Evaluation of system categories for IgniteWorx

In summary, it seems that IgniteWorx would very well fit a combination of expert system and semantic reasoning system, with additional features borrowed from recommender systems and survey engines.

3.3.3 Related Concepts and Algorithms—State of Science

There are a number of related concepts and algorithms that could be helpful for the IgniteWorx solution design, including analytics hierarchy process (AHP), artificial intelligence, Bayesian networks, decision trees, Markov decision process (MDP), and multi-criteria decision analysis (MCDA). Many of these concepts and algorithms describe the computational foundations for the systems in the previous section or are related in some other way.

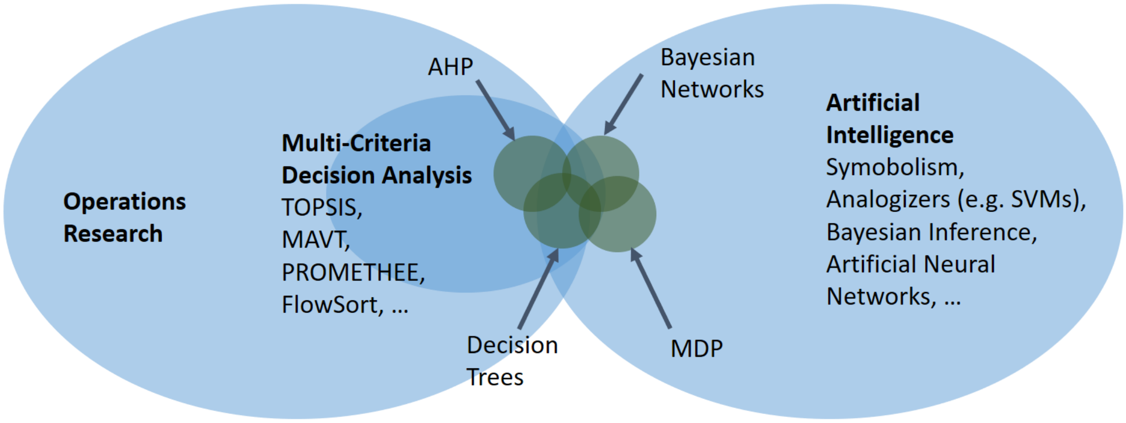

Figure 56: Concepts and algorithms related to IgniteWorx problems

Artificial intelligence and multi-criteria decision analysis are broader frameworks that include some of the other concepts and algorithms described in this section. However, since there seems to be no clear delineation or hierarchy levels between most of the concepts outlined in this section, (Figure 56 attempts to provide an overview of how they relate to each other based on the findings of the research of this thesis), they are presented in alphabetical order.

3.3.3.1 Analytic Hierarchy Process

AHP is an approach for supporting complex decision analysis (see section 3.3.3.5) based on mathematical and psychological concepts. Detailed descriptions of the theory and the underlying the process can be found in Saaty and Saaty (2000).

AHP is designed to help with the decomposition of a complex problem into a hierarchy of sub-problems, which can then be addressed individually. The different elements in an AHP hierarchy can be of different kinds, qualitative or quantitative.

After building the hierarchy, it is systematically evaluated. The different elements in the hierarchy are each compared (two at a time) to understand their impact on the element above them in the hierarchy. By using weights or priorities for individual elements in the hierarchy, the AHP algorithm determines the best matching option.

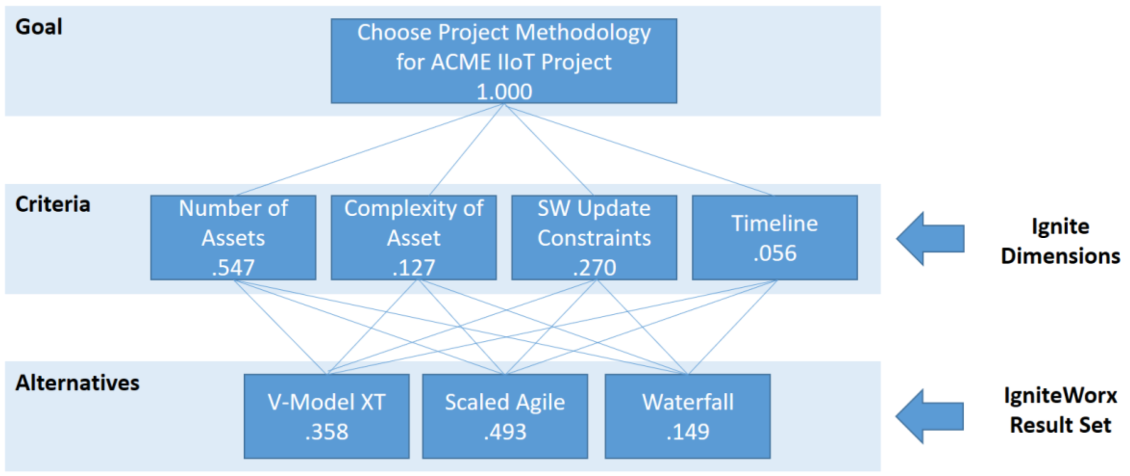

Figure 57: AHP example hierarchy for IgniteWorx

Figure 57 applies the standard example from Saaty and Saaty (2000) to one result set from IgniteWorx, using four sample Ignite dimensions to help choose the best suitable project management methodology for a fictional ACME IIoT project. The example shows that an AHP-inspired algorithm could well work for IgniteWorx.

It seems as if AHP could be incorporated into at least some of the system architecture described in the previous section. For example, Ghodsypour and O'Brien (1998) describe a decision support system for supplier selection using an integrated analytic hierarchy process and linear programming.

However, AHP is not without criticism. Pérez et al. (2006) state that many features of AHP have been criticized, "especially the additive hierarchical composition of convention AHP, which leads to the possibility of occurrence of the Rank Reversal phenomenon (adding an irrelevant alternative may cause a reversal in the ranking at the top)." Whitaker (2007) analyzes criticisms of AHP and "why they often make no sense," claiming that "by correctly structuring and setting the priorities they do give the expected results."

3.3.3.2 Artificial Intelligence

According to introductory texts like those of Poole and Mackworth (2017) or Neapolitan and Jiang (2018), modern AI is aiming to solve problems like reasoning and problem solving, planning, learning, natural language processing, perception, and moving or manipulating objects. Expert systems are combining AI and knowledge representation to support complex decision processes. AI methods include computational intelligence and statistical methods, as well as traditional symbolic AI. Some of the well-established AI approaches include:

1. Symbolism: Using formal logic, e.g., with IF/THEN-like rules. Haugeland (1985) described this as GOFAI ("good old-fashioned artificial intelligence").
2. Bayesian inference: statistical inference based on Bayes' theorem (Lee, 2012)

3. Analogizers: using classification algorithms like support vector machines (SVMs) and nearest-neighbor algorithms for pattern recognition (Theodoridis and Koutroumbas, 2009)
4. Artificial neural networks solve problems by using collections of connected nodes that loosely model the neurons in a biological brain (van Gerven and Bohte, 2018).

From the point of view of IgniteWorx, these different approaches have to be seen in the context of the expected evolution of an IgniteWorx system (see 2.7): During the first two phases, only very limited IIoT project reference data will be available. Only in phase three can it be expected to have a significant number of IIoT project assessments and qualified meta data available (e.g., information about the relevance of previous recommendations). This means that in the initial phases of the system evolution, formal logic seems to be the most likely candidate; this approach will allow us to manually define the rules, which can help determining the best possible project setup for an IIoT project, based on the project characteristics. Statistical inference, analogizers, and neural networks all rely on the availability of significant reference data that can be used for automatic deductions.

3.3.3.3 Bayesian Networks

According to Szocs and others (2008), a Bayesian network (BN) is "a probabilistic graphical model for representing causal relationship among variables. It consists of a set of nodes and directed arcs. The nodes represent variables and the arcs represent the directed causal influences between linked nodes." BNs can also provide the basis for inference and learning algorithms (see discussion of AI in section 3.3.3.2).

BNs can be evaluated using Bayes' theorem (Bayes et al., 1763), e.g., to determine the probability of an event based on previous events and conditions modeled in the BN. Bayes' theorem is defined mathematically by the following equation:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

where A and B are events. P(A|B) and P(B|A) are conditional probabilities, e.g., the probability of event A, in case B is true. P(A) and P(B) are marginal probabilities, i.e., the probabilities of observing events A and B independently (Kendall et al., 1994). Bayesian networks and Bayes' theory can be used in combination to model and evaluate complex causal dependencies and probabilities.

From the point of view of IgniteWorx, a BN is interesting because it could present the probabilistic relationships between elements of an IIoT project assessment and elements of the IgniteWorx results sets.

BNs seem to be commonly applied to solve problems in project management, as well as software project management specifically. For example, Lee et al. (2009) describe an approach for large engineering project risk management using a Bayesian belief network.

A similar approach, but specifically for software projects, is described by Hu et al. (2013). Ancveire et al. (2015) go one step further, applying Bayesian networks to software delivery risk management in agile software development.

DEMELO and SANCHEZ (2008) provide a concrete example for a BN to predict software maintenance project delays, as shown in Figure 58. The example shows how a maintenance project's chances for being finished on time can be derived in a BN from factors such as potential delays in implementation and testing, testing complexity, and also platform and system expertise, as well as risks induced by bringing in new resources. It can be relatively easily imagined how this approach can be mapped to the key artifacts of IgniteWorx. See section 3.3.5.5 for more a concrete mapping.

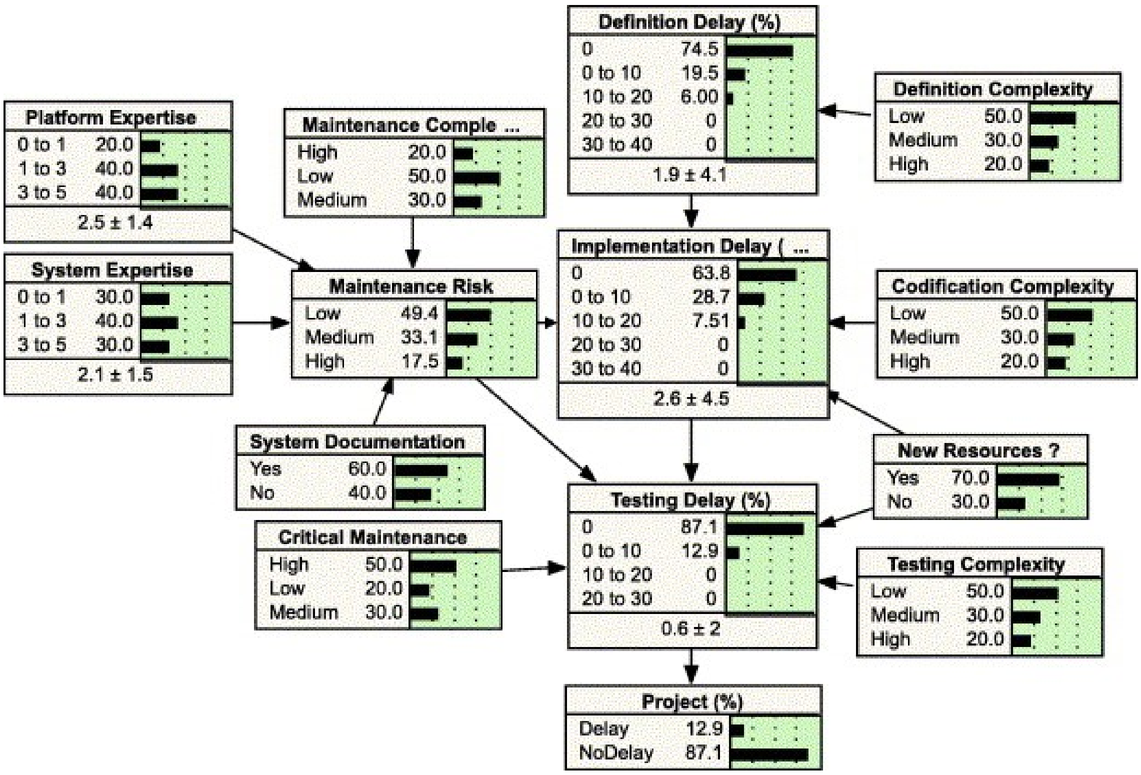

Figure 58: Bayesian network to predict software maintenance project delays (DEMELO and SANCHEZ, 2008)

Another area for the application of BNs in project management is cost estimation. For example, Chulani et al. (1999) describe a Bayesian analysis of empirical software engineering cost models. Khodakarami and Abdi (2014) give a Bayesian networks approach for modeling dependencies between cost items to support project cost risk analysis.

Given that many of these topics are also very close to IgniteWorx (both risk management as well as cost estimation have been identified for the initial IgniteWorx result sets), BNs could be an interesting candidate for the IgniteWorx solution design.

3.3.3.4 Decision Tree Analysis

Decision trees are support tools for decision making, using a tree-like model of decisions and their potential consequences. A decision tree can also be seen as the visualization of an algorithm based on conditional control statements.

Decision trees are commonly used in operations research to help select the best matching business strategy (Wagner, 1975). Another common use is in machine learning (Witten et al., 2016).

Figure 59 provides an example of a decision tree in a project management situation. In this case, the decision tree is designed to help evaluating different options for building or upgrading an existing plant. Decision nodes provide data about concrete decisions. Chance nodes provide data about events with different probabilities. The branch nodes provide the computed value of each branch.

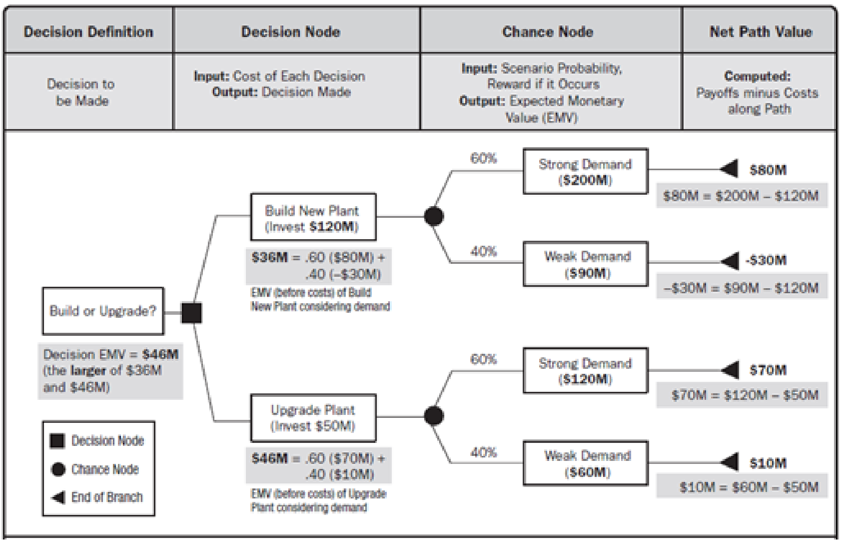

Figure 59: Example of decision tree in project management (Ogunsina, 2013)

The benefit of decision trees from the point of view of IgniteWorx is that they present a way for dealing with causal relationships and probabilities in a relatively simple way. For example, decision trees do not have to deal with the typical graph traversal problems that can be found, for example, in Bayesian networks.

3.3.3.5 Multi-Criteria Decision Analysis

Multi-criteria decision analysis (MCDA) describes a set of methods designed to “evaluate alternatives based on multiple criteria using systematic analyses which overcome the limitations of unstructured individual or group decision-making” (Yatsalo et al., 2015), often supported by decision support systems (DSS, see 3.3.2.1).

Belton and Stewart (2002) describe the main categories of problems that are well handled through MCDA:

- **Screening alternatives**—a process of eliminating those alternatives that do not appear to warrant further analysis
- **Attention**, i.e., selecting a smaller set of alternatives that likely contains the "best"/trade-off alternative
- **Sorting** alternatives into classes/categories (e.g., "unacceptable," "possibly acceptable," "definitely acceptable," etc.)
- **Choice /selection**—finding "the most preferred alternative" from a given set of alternatives
- **Ranking** alternatives (from "best" to "worst" according to a chosen algorithm)
- **Designing** (searching, identifying, creating) a new action/alternative to meet goals

From an IgniteWorx point of view, especially selection and ranking/sorting are the most relevant aspects because the approach requires methods that help to select and rank the best possible matching results from a result set for a given IIoT project, based on the specific project characteristics.

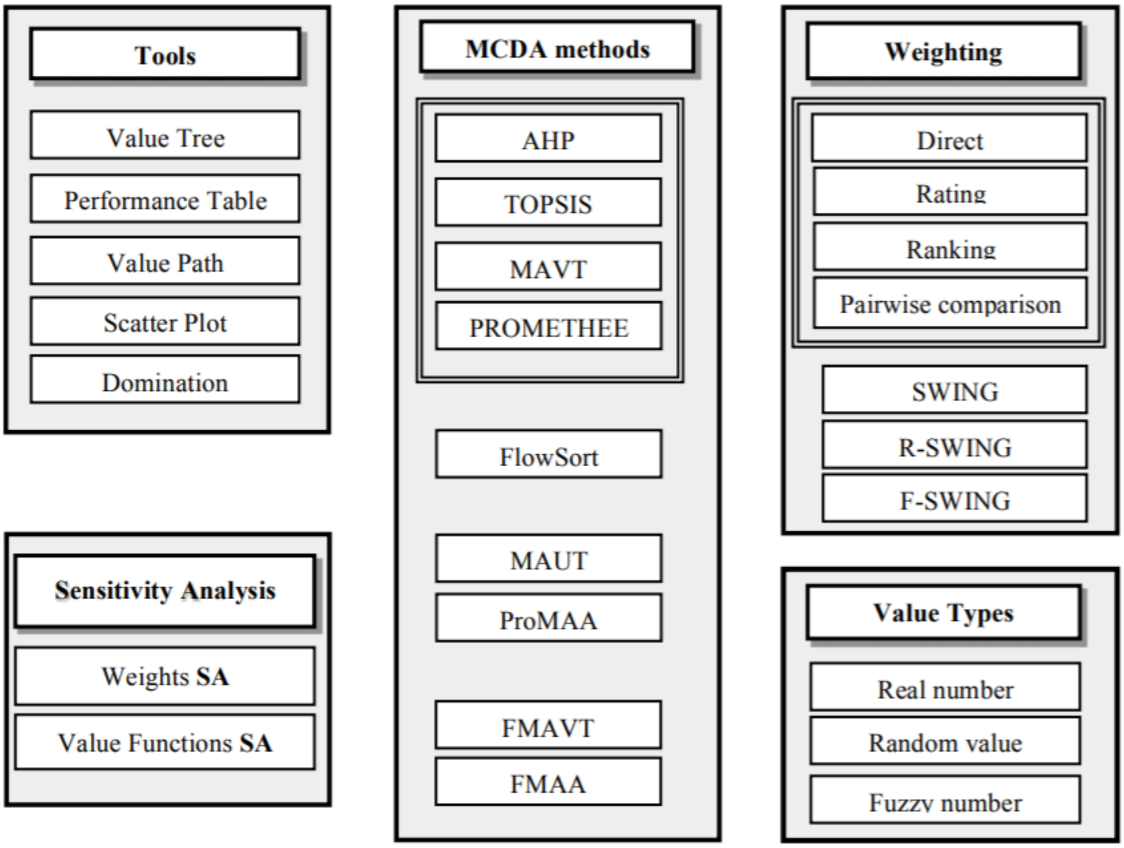

Figure 60: General schema of the DecernsMCDA (Yatsalo et al., 2015)

Yatsalo et al. (2015) introduce the DecernsMCDA, which is a multi-MCDA model tool. The general schema supported by DecernsMCDA provides a good overview of key elements of an MCDA system, including tools (value tree, performance table, value path, scatter plot, domination), sensitivity analysis (weights SA, value functions SA), MCDA methods (AHP, TOPSIS, MAVT, PROMETHEE, FlowSort, MAUT, ProMAA, FMAVT, FMAA), weighting (direct, rating, ranking, Raiwise Comparison, SWING), and value types (real numbers, random values, fuzzy numbers).

Yatsalo et al. (2015) identify the following as the most widely used MCDA methods:

- AHP: analytic hierarchy process
- FlowSort: sorting method with the use of net flows
- MAVT: multi-attribute value theory
- PROMETHEE: preference ranking organization method for enrichment evaluations
- TOPSIS: technique for order preference by similarity to the ideal solution

Because the hierarchical approach proposed by AHP seems to be closest to the needs of modeling and evaluating hierarchies of Ignite dimensions and IgniteWorx result sets, it is included in this thesis for further analysis (see section 3.3.3.1).

3.3.3.6 Hybrid Concepts and Algorithms

The research on potentially suitable concepts and algorithms for the implementation of IgniteWorx has shown that many of the above concepts and algorithms are also used in combination, also in the context of project management.

For example, Dey (2002) proposes to combine AHP and decision tree analysis (DTA) for project risk management, claiming that the combined "AHP and DTA approach not only determines probability and severity of risk factors, but also identifies risk responses for each work package."

Also, the combination of AHP and Bayesian networks are described in a number of papers. For example, Zubair (2014) describes a hybrid approach for reliability analysis based on analytic hierarchy process and Bayesian networks. Huang and Bian (2009) describe an approach for personalized recommendations based on combining Bayesian networks with analytic hierarchy process.

Finally, another interesting combination of concepts is discussed in Wątróbski and Jankowski (2015), who propose the use of ontologies for knowledge management in the MCDA domain. Ontologies are introduced in section 3.3.2.4 as part of the discussion on semantic reasoning systems.

3.3.3.7 Evaluation for IgniteWorx

The following provides a tabular evaluation of the different concepts and algorithms that have been considered for IgniteWorx.

Concept	**Evaluation**
Analytics Hierarchy Process	AHP seems structurally relatively close to the needs of IgniteWorx since it supports a simple mapping of

	multiple potential results (e.g., from IgniteWorx result sets) to different criteria (e.g., Ignite dimension selected by an individual IIoT project). However, the different criticisms of AHP (e.g., Pérez et al. (2006)) will have to be carefully taken into consideration.
Artificial Intelligence	AI in general as a concept is too broad to be evaluated here. As discussed in section 3.3.3.2, for phase 1 of IgniteWorx, formal logic might be the most suitable since it can be used to capture mapping rules explicitly. Self-learning techniques will only be applicable in phase 3 of the IgniteWorx evolution once reliable reference data will be available.
Bayesian Networks	Bayesian networks seem to be the most robust and proven approach for managing the causal dependencies and probabilities in IgniteWorx. They also support explicit knowledge modeling, which is key for phases 1 and 2 of the evolution of IgniteWorx.
Decision Tree Analysis	Decision trees provide a simpler way than, for example, BNs to deal with causal dependencies and probabilities. If the mappings between key IgniteWorx artifacts can be mapped to tree-like structures, they would be an option.
Multi-Criteria Decision Analysis	Again, MCDA is too broad a concept to be directly applicable to IgniteWorx. Instead, AHP as a key MCDA methodology is evaluated and considered here.

Table 24: Evaluation of related concepts for IgniteWorx

In conclusion, Bayesian networks and decision trees should be considered for the foundation of an IgniteWorx implementation, potentially in a hybrid implementation. Analytics hierarchy process could also be of interest but must be treated with care given that it is not without criticism.

3.3.4 Proposed System Architecture for IgniteWorx

The proposal for the IgniteWorx architecture must consider the findings from the research of the state-of-science of systems and concepts, as well as the anticipated system evolution. Consequently, the following looks again at the expected system architecture evolution before discussing the concrete proposal for the resulting IgniteWorx architecture.

3.3.4.1 Expected System Architecture Evolution

As discussed in 2.7, it is assumed that any concrete IgniteWorx implementation will evolve in phases. These phases relate specifically to the availability of IIoT project reference data, as well as user feedback. In phase 1, the assumption is to have very little of both, in phase 2 the first meta data will be partially available, and only in phase 3 will the system have reliable meta data.

As a consequence for the resulting system architecture, this means, that at least for phase 1 and probably also phase 2, an IgniteWorx system will have to rely on explicit knowledge modeling, while only toward phase 3 will any meaningful, self-learning principles make sense.

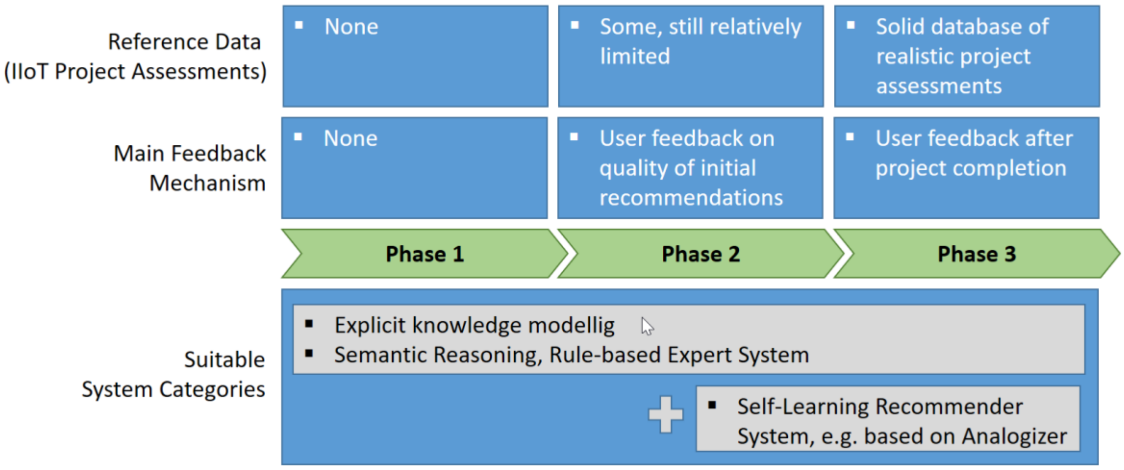

Figure 61: Expected system architecture evolution

Figure 61 adds the suitable system categories to the system evolution diagram from section 2.7, showing where explicit modeling fits in and where self-learning recommender system technologies would start becoming relevant.

3.3.4.2 Initial Hybrid Architecture

Taking the above discussion on the expected system evolution into consideration, as well as the learnings from the research on suitable systems and concepts, the proposal is to use a hybrid architecture for IgniteWorx, which draws on the different concepts evaluated in sections 3.3.1 and 3.3.3.

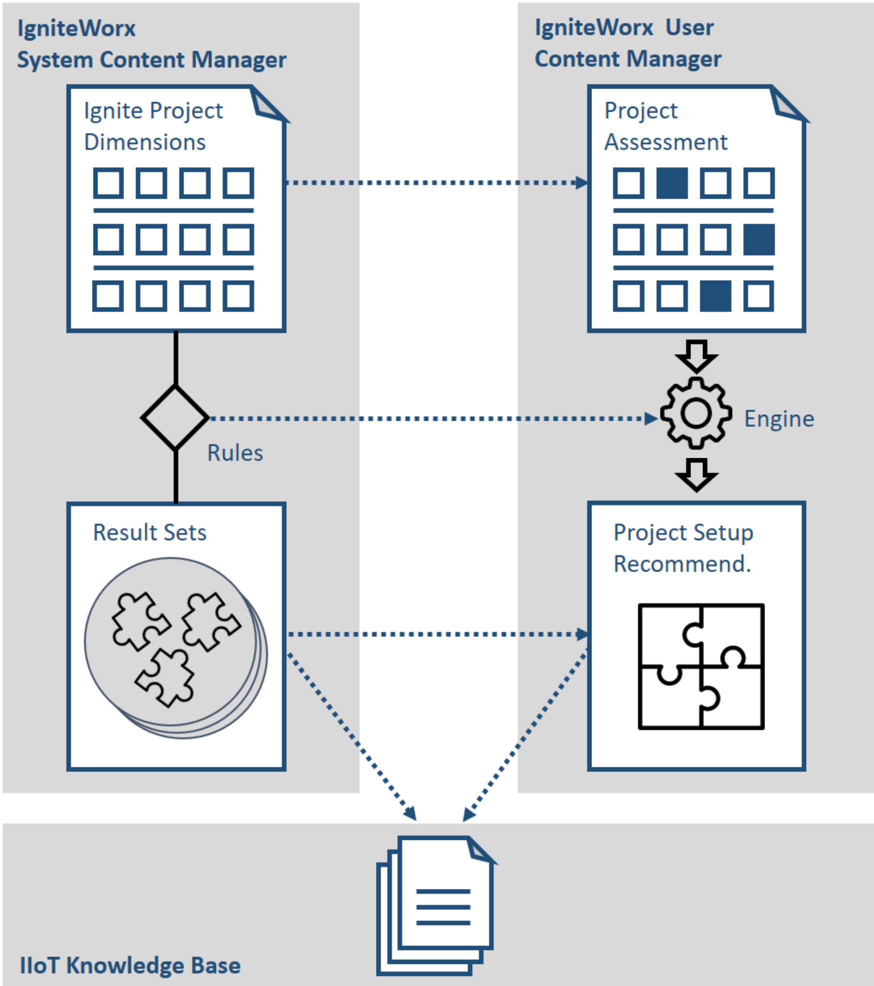

Figure 62: Proposed IgniteWorx architecture

The proposed IgniteWorx architecture depicted in Figure 62 includes key concepts recommended in section 3.3.1, including the use of architecture components from expert systems, semantic reasoning systems, and survey engines.

The concrete architecture proposal includes three main components:

- **IgniteWorx System Content Manager:** This component supports storage and management of Ignite project dimensions, IgniteWorx rules, and IgniteWorx rule sets. Effectively, this components is a specialized content management system with a custom scheme and UI to support the aforementioned entities.
- **IgniteWorx User Content Manager:** This component supports project assessments and recommendation of IIoT project setups.
    - **Project Assessments:** This module works like a survey engine, as described in section 3.3.2.5, driven by content from the IgniteWorx system content manager. Depending on user preferences, this module might support storage of project assessments either in the local browser (accessible only for the individual user) or in the backend (potentially shared).

- **Engine:** The IgniteWorx engine takes IIoT project assessments and creates recommendations for a concrete project setup. The engine must apply the IgniteWorx rules to the concrete project assessment data to create matching recommendations.
- **Recommendations:** The recommendations are the output of the IgniteWorx engine. The engine can make one recommendation per result set or provide multiple results per result set, including a ranking. Recommendations should not only provide a link to the result details but also an explanation why a particular recommendation was made (this was described as a key feature of recommender systems, e.g., in the Hulu example). Also, expert systems typically include an explanation system (see Figure 54).

- **IIoT Knowledge Base:** The IIoT knowledge base must provide detailed explanation of the different results from the IgniteWorx result sets. This can either be a central knowledge base or a distributed knowledge base (e.g., the Internet)

Each of these components could be implemented as a micro-service. This would ensure improved maintenance and easier extensibility of the system (Namiot and Sneps-Sneppe, 2014).

3.3.4.3 User Roles and User Interfaces

A clear understanding of user roles and user interfaces (UIs) is essential for any system design. In this section, both are described before being mapped against the proposed IgniteWorx system architecture outlined in the previous section.

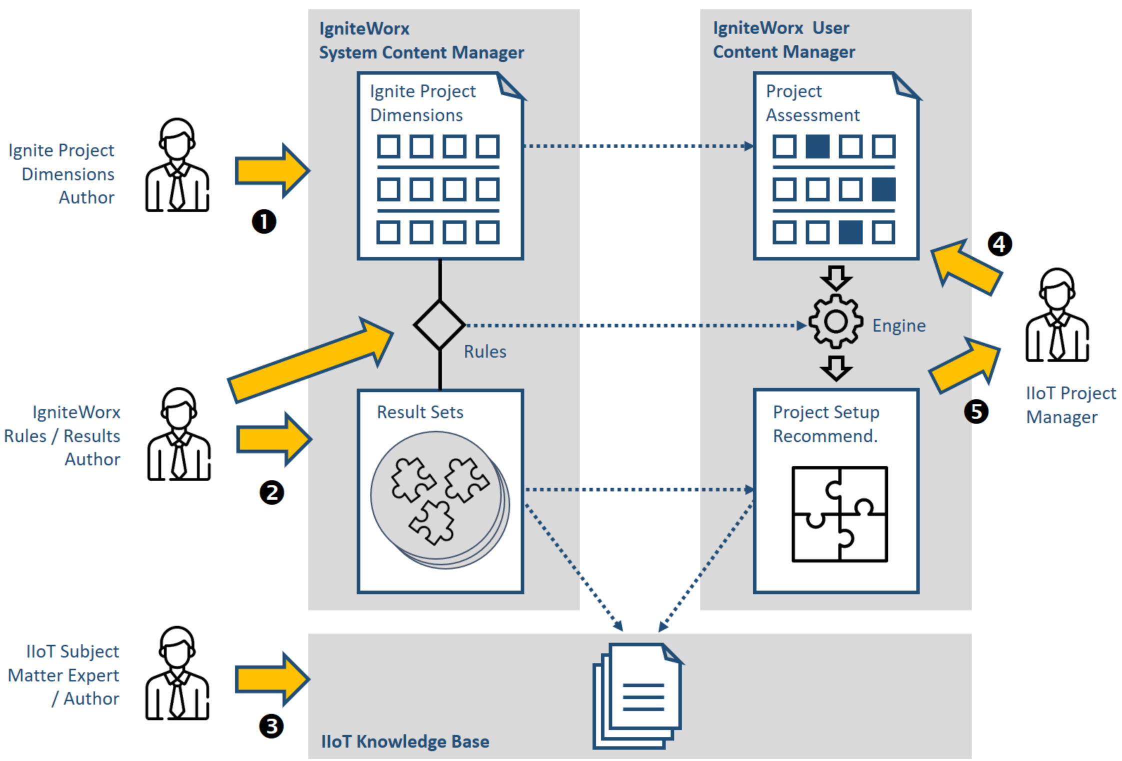

Figure 63: User roles and user interfaces in IgniteWorx

Figure 63 provides an overview of the key user roles and required user interfaces in IgniteWorx:

- **Ignite Project Dimensions Author (1):** Responsible for creation and maintenance of Ignite project dimensions, using specialized UIs for this task
- **Ignite Rules / Results Author (2):** Responsible for creation and maintenance of IgniteWorx results sets, results and the rules that create the mapping between project dimensions and results, using specialized UIs for this task
- **IIoT Subject Matter Expert / Author (3):** Responsible for creation and maintenance of the content in the IIoT knowledge base, using specialized UIs for this task
- **IIoT Project Manager:** The end-user of the system, using it for assessing his or her individual IIoT project and getting concrete recommendations. Will require specialized UIs to support
    - **Project Assessment (4):** The UI used to perform the project assessment based on the Ignite dimensions
    - **Project Assessment Outcome (5):** The UI providing assessment summary and recommendations for individual project setup

The system implementation must ensure that the UIs as described above are provided in a way that allows efficient data capturing and presentation and also supports data privacy policies for the IIoT project manager to provide him or her with a choice of whether he or she is willing to share his or her project assessment data or not.

3.3.5 IgniteWorx Design Models

To allow for an efficient implementation of a hybrid architecture as proposed in section 3.3.4.2, it is assumed that an efficient, flexible, and easily extensible implementation approach is required. Such an approach is supported by model-driven architecture (MDA), as introduced in the following. Following the MDA design philosophy, the remainder of this section provides descriptions of different model elements of the proposed IgniteWorx design, using simplified unified modeling language (UML).

3.3.5.1 Model-Driven Architecture

Kempa and Mann (2005) describe MDA as a software development process where models are the central elements. The goal is to generate platform-specific implementations, at least partially, from the platform independent models. The goal is to reduce software development costs and enable technology independence. Mellor et al. (2002) describe how UML can be used to generate concrete implementations from models, e.g., using modern programming languages such as Java or C#. Modern application development tools such as Ruby on Rails, Django, JHipster, and others are supporting a scaffolding approach, which automatically creates working—albeit usually rudimentary—applications from models (Sasidharan and Kumar, 2018).

To best support a hybrid architecture as introduced in 3.3.4.2, the IgniteWorx design proposes an MDA approach for the implementation. With this, one can select the most suitable elements from the different system categories described in section 3.3.1, as well as from the related concepts and algorithms described in section 3.3.3, model them using a modeling language such as UML, and then use an MDA-based rapid application development (RAD) or scaffolding framework to automatically generate the skeleton of the new system. This should help with minimizing development costs and provide a high degree of flexibility for supporting the evolution of the system.

In section 3.3.2.4, W3C's OWL is taken as an example for technologies to support semantic reasoning systems. In theory, a language such as OWL could be used to model parts of the required functionality of IgniteWorx (an example is provided). However, most tools that support OWL or similar ontology languages are not designed to support more modern, web-based applications in a way that the aforementioned scaffolding tools are providing. This is why in the following basic UML is used to provide a consistent design for all required micro services of IgniteWorx. Zedlitz et al. (2012) describe an approach for creating OWL models from UML. This approach could be used to derive OWL models for those services where it is required.

3.3.5.2 Ignite Project Dimensions

For the Ignite project dimensions, a hierarchical data structure is suggested to manage project dimensions and options, as described, for example, in 3.1.1.

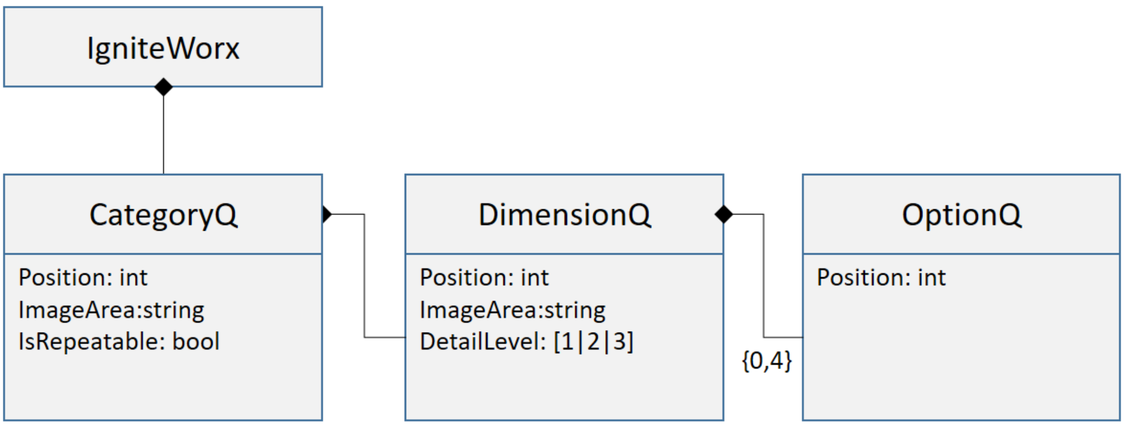

Figure 64: IgniteWorx project dimensions

Figure 64 provides an overview of the proposed design. IgniteWorx is defined as a so-called root entity, which aggregates all other entities underneath. All aggregated entity names have the character Q as a post-fix to indicate that they are actually describing the question side of the project dimensions. These entities are designed to actually capture the project survey questions (as opposed to the survey answers provided by the end-users, as described in the next section).

As the first aggregation level, the entity CategoryQ is proposed. This entity can be used like a folder, which groups together different dimensions. Examples of concrete instances would be assets and devices or backend services (see section 3.1.1.). The position attribute can be used by the IgniteWorx engine for ordering the categories when presenting them as questions to the end-user. The ImageArea attribute can be used by the IgniteWorx engine to display an illustrative image, e.g., as a reusable solution canvas that guides the user through the assessment like a visual navigation tool. The IsRepeatable attribute indicates whether a category can be repeated in the project assessment. For example, the category "Assets and Devices" could be repeated for multiple asset classes.

Categories aggregate 0-n dimensions. In addition to "Position" and "ImageArea," a "DimensionQ" has a "DetailLevel". This attribute can be used to support different levels of detail for the project assessment. For example, level 1 could indicate a quick survey, while level 3 indicates that an instance of this DimensionQ would be included only in a detailed assessment.

Each dimension aggregates exactly 0 or 4 options. An "OptionQ" represents one of the options offered for a question in the project assessment. As discussed in section 3.1.1., each dimension should offer exactly four options. Zero options will also be allowed for dimensions that are actually not really questions but rather an explanation about the next questions. Using this design, the aggregation hierarchy in this case is kept simple and does not use, for example, nestes categories.

3.3.5.3 Project Assessment

Having defined the question catalogue using the concept of Ignite dimensions, the next requirement is a structure to capture the concrete answers provided by the end-user in an individual project assessment.

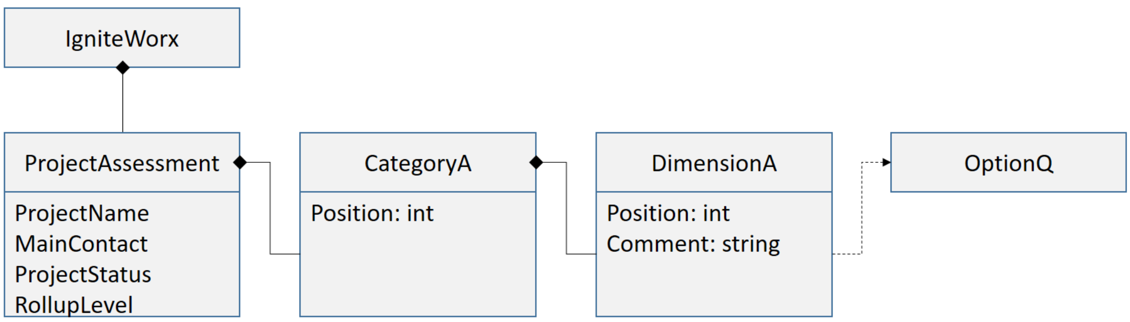

Figure 65: IgniteWorx project assessment

Figure 65 provides an overview of the proposed design: "IgniteWorx" aggregates "Project Assessments." From an MDA point of view, one complication here is that project assessments are not necessarily saved in the backend, so the MDA framework will have to allow for an exception here, for example, supporting data storage in the end-user's browser.

Other than this, the proposed aggregation hierarchy is mirroring the design of the previous section: "ProjectAssessment" aggregates "CategoryA," which in turn aggregates "DimensionA" ("A" is now used as a post-fix since these data structures are designed to capture the answers, not the questions).

The link between questions and answers is created through the association between "DimensionA" and "OptionQ". This means that a "DimensionA" is representing a concrete answer, which references one of the four options provided by the question.

3.3.5.4 Result Sets

For the design of the result sets, there are at least two options. The recommended one is examined before discussing the alternative.

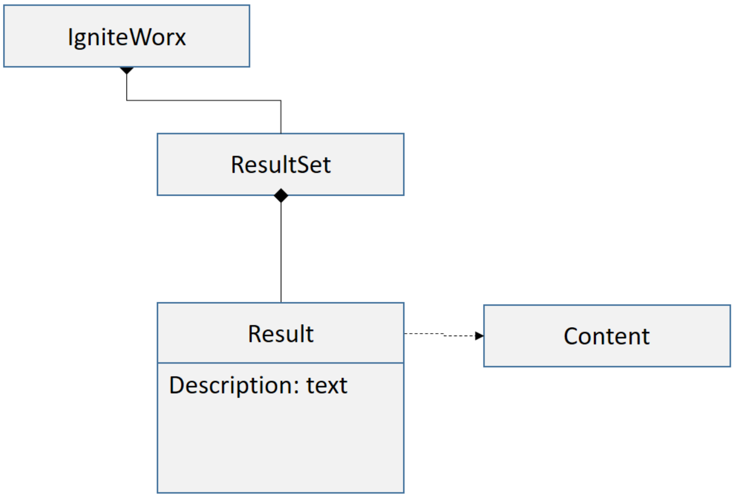

Figure 66: IgniteWorx result sets

Figure 67 provides an overview of the recommended design: "IgniteWorx" aggregates "ResultSets," which in turn aggregate "Results." Each result has an association with an abstract entity "Content," which represents content in the external services "IIoT Knowledge Base," introduced in section 3.3.4.2.

The alternative design would have been to use an inheritance relationship between "Result" and "Content." The advantage of this approach would be that the actual number of instances to be manually maintained would have been reduced by half. Instead of creating a new instance "Result" for each result, a "ResultSet" could simply refer to different content elements included in this particular "Result Set." However, the disadvantage of this design is that it would create a tight coupling between the components "IgniteWorx System Content Manager" and "IIoT Knowledge Base," which would be against the principle of micro-services, as introduced in section 3.3.4.2.

3.3.5.5 Rules

This work proposes to also apply the MDA approach to IgniteWorx rules and manage them as entities in the repository. The advantage is that the same MDA templates used for the other entities can be used to create and manage rules instances, including the required UIs for this process (assuming the chosen MDA system is supporting this).

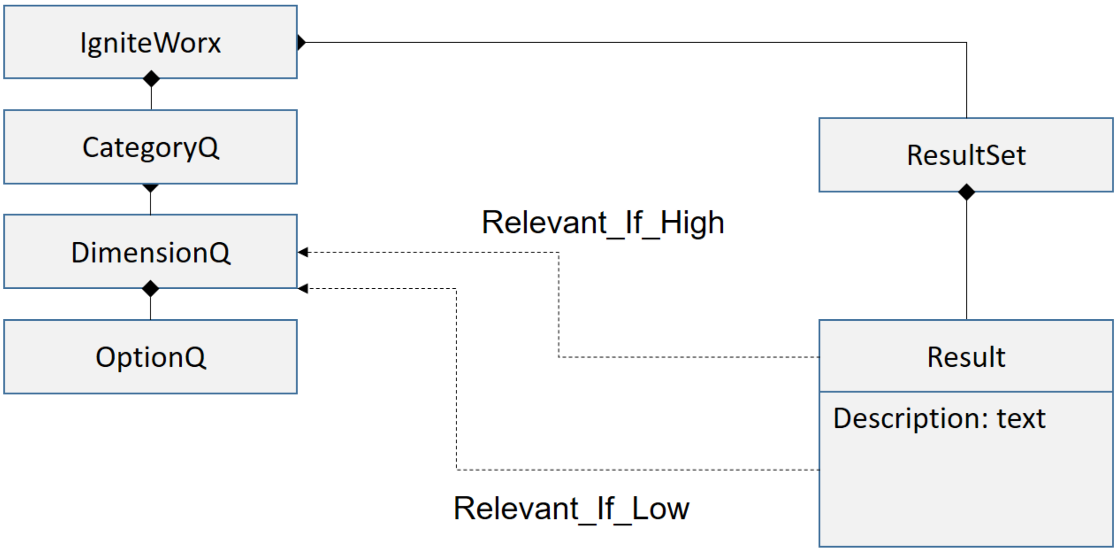

Figure 67: IgniteWorx rules

Figure 67 provides an overview of the proposed design for rules, which is actually relatively lightweight: Instead of modeling an "IgniteWorx Rule" as a dedicated entity, it is proposed to model rules as an association between "Result" and "DimensionQ." An instance of a rule is required to indicate whether a particular result has a high (or low) relevance for a given dimension. This design is supporting exactly this. By actually using two different associations ("Relevant_If_High" and "Relevant_If_Low"), one can express whether the relevance is based on a low or high value of the dimension.

As long as the expressiveness of a rule is limited to this semantic, not modeling a rule as a dedicated entity significantly reduces the number of instances created in the system. This is a huge benefit from a content management point of view since rules are—at least initially—created and maintained manually.

The idea is that this design will also support self-learning concepts in the next phase of the system evolution, as described in section 3.3.3.2. Analogizers or a Bayesian inference algorithm could extend the current data structures to make them suitable for its own needs without having to replace the manually created content from phase 1.

3.3.5.6 Engine and Matching Algorithm

The matching algorithm proposed for phase 1 of the system evolution is relatively straightforward. Figure 68 provides an overview using a concrete example.

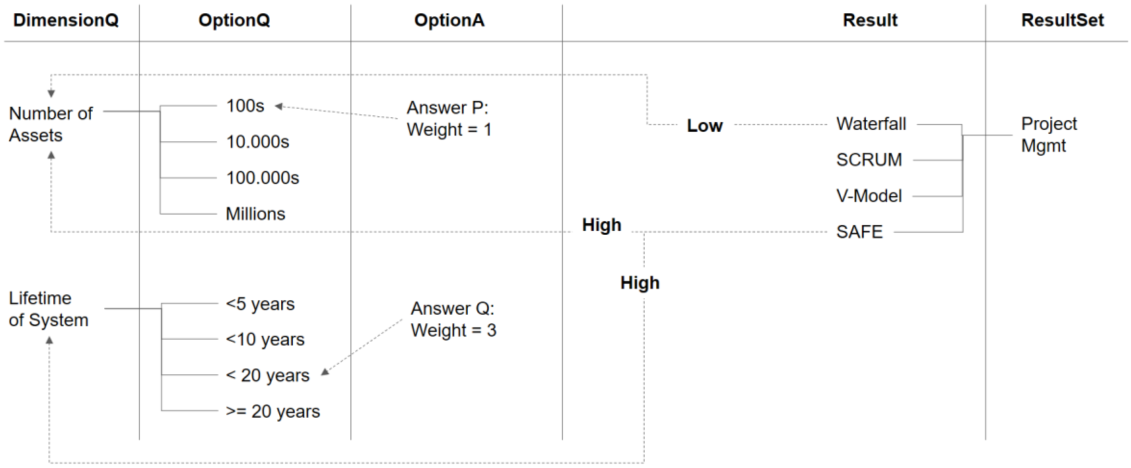

Figure 68: Example of IgniteWorx rules and matching algorithm

This example describes an instance of "ResultSet" with four "Results" as children: "Waterfall," "SCRUM," etc. On the other end, there are two examples: "DimensionQ"/"Number of Assets" and "Lifetime of System." "SAFE (Scaled Agile)" has a reference ("Relevant_If_High") to "Number of Assets." This means that if a user indicates that in his or her project he or she has a high number of assets, then SAFE could be considered as a project management methodology.

The options always go from 1 to 4. This means, for example, that such a rule match could add a 4 to an overall weight. For "Relevant_If_Low," the inverse order (4 to 1) could be used.

This design proposal is not specifying exactly how the algorithm should calculate the weights of the individual results, but there are multiple options:

- Sum: In this example, SAFE gets 1+3=4 (building the sum). This means that there is no maximum weight.
- Average: SAVE gets (1+3)/2 = 2 (take average). This means that all weights are always between 1 and 4.
- Pairwise comparison, as proposed by AHP (see 3.3.3.1)

3.3.6 Evaluation of IgniteWorx Architecture Proposal

In section 3.3.1, a set of IgniteWorx-specific architecture evaluation criteria were defined, based on the architecture tradeoff analysis method (ATAM). Table 25 looks at each of the previously defined, architecture evaluation criteria and provides a discussion about how far the criteria have been met and with which level of quality.

ATAM Criteria	How is the criteria met?	Quality of match

Performance	Standard web-architecture based on well-proven PaaS	Will depend on quality of PaaS and implementation details
Reliability	MDA, especially for design and implementation of all aspects of IoT project assessments (Ignite dimensions, result sets, rules)	Will depend on the quality of the chosen MDA framework
Availability	Standard web-architecture based on well-proven PaaS in combination with MDA	Will depend on the quality of implementation
Security	Standard PaaS approach for basic security. Also key: IoT project assessment data is always stored locally on the client's PC, never in the cloud.	Should be augmented with security validation process
Modifiability	MDA ensures high level of modifiability for built-in models. Also, MDA approach enables high-quality web-content editors for all configurable parts (Ignite dimensions, result sets, and rules).	Should be very high
Portability	Lock-in into PaaS-specific features should be avoided.	Will depend on implementation details
Functionality	Detailed domain designs in section 3.3.5 in combination with MDA approach will cover all required data-centric functionality. Implementation of required algorithm from section 3.3.5.6 must be supported via APIs.	Solid, data-centric foundations. Quality of algorithms and also user experience will depend on how well the MDA framework supports customizations, e.g., via APIs.
Variability	Requirements for system evolution described in section 2.7. will be supported by hybrid-architecture proposal, described in section 3.3.4.2. This will allow to start with	Will depend on the quality of integration of the different concepts in the hybrid architecture

	an explicit knowledge representation and move toward a self-learning approach in the next steps of the system evolution. This is not a technical limitation but is rather dependent on content availability.	
Subsetability	MDA approach and clear domain design in combination with micro-service architecture ensures “subsetability.” Detailed domain design from section 3.3.5.2 ensures reuse of the existing Ignite concepts.	Should be very good, given clear and concise domain design
Conceptual Integrity	MDA approach ensures unified design.	Should be very good, assuming MDA approach is consequently used

Table 25: Evaluation of IgniteWorx architecture proposal based on architecture evaluation criteria

The findings summarized in Table 25 allow the conclusion that the previously defined architecture evaluation criteria have been met for most points, as far as this can be evaluated on the conceptual level. A final evaluation is provided in section 9.6.1, this time based on the findings of the two research iterations in this thesis, including a concrete implementation of the design proposal described in this section.

4 Demonstration I

On October 16, 2018, the IIC announced the availability of the public beta of the IIC Resource Hub during the opening keynote of the IoT Solutions World Congress in Barcelona. A key part of the IIC Resource Hub (IIC, 2018d) is the IIC Project Explorer. IIC has licensed the Ignite methodology via the Creative Commons Attribution 3.0 Unported License (see section 1.9). The Ignite methodology has become an essential part of the IIC Project Explorer, tailored for the specific needs of the IIC (a main reason for the choice of CC BY 3.0 was to allow exactly this).

While the IIC Project Explorer is not a 1:1 implementation of IgniteWorx, it has been chosen as the main demonstration artifact of this thesis since it is structurally very close. In the following, an overview of the IIC Project Explorer is provided, including a discussion of the differences between the IIC implementation and the original IgniteWorx design. It should be noted that the IIC Project Explorer itself is copyright IIC and not part of the work results of thesis. Finally, the demonstration I part of this thesis only uses publicly available features of the IIC Project Explorer.

4.1 IIC's Actionable Intelligence

The IIC has used the term "actionable intelligence" to describe how its Resource Hub knowledge base and the IIC Project Explorer are providing a joint value proposition. The first page of the IIC Project Explorer provides a visual overview of how these concepts fit together, as shown in Figure 69.

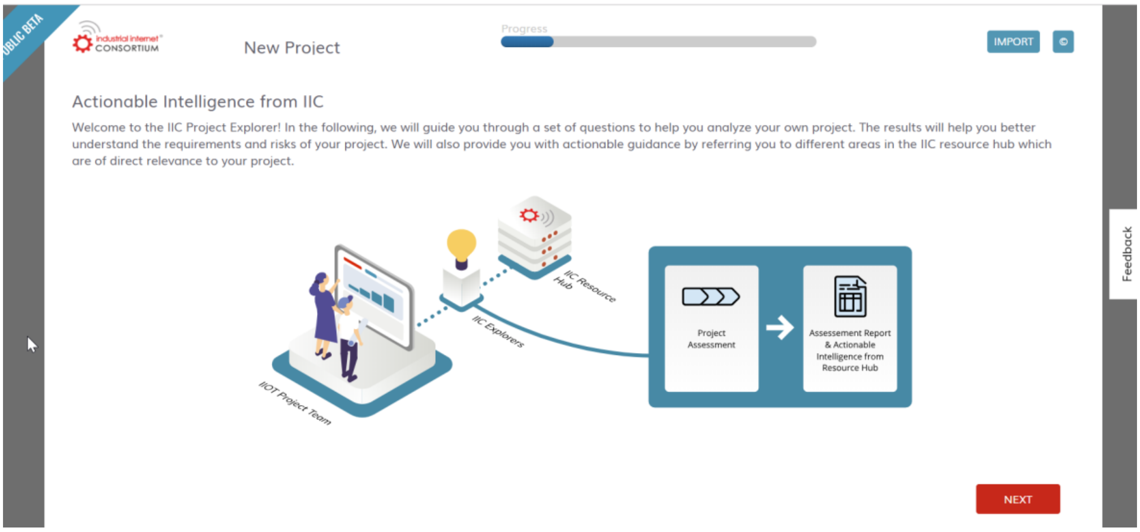

Figure 69: IIC's actionable intelligence

The overview diagram shows how the IIC Project Explorer is positioned as the interface between the IIoT project team and the Resource Hub. It also depicts how the Project Explorer starts with the project assessment and delivers an assessment report, as well as "actionable intelligence," referring to the recommended readings for the project setup, which are essentially the best-rated results from the underlying result sets (see section 4.8). Since making direct recommendations can be difficult for legal reasons, this implementation is using the concept of recommended readings.

Other than this, one can already see on this page that the IIC Project Explorer is designed like a classical wizard that can be found in many web applications that guide a user through a complex data capturing process, including progress bar, as well as load/save functions to support temporary suspension of the work. This kind of feature can be found, for example, in survey engines, as described in section 3.3.2.5.

4.2 IIC Project Explorer: Data Privacy

As described in section 3.3.4.3, an IgniteWorx system will have to ensure a strict data privacy policy; the project assessment is capturing data that has potentially significant commercial value for the company running the IIoT project. Figure 70 shows how this is supported in the IIC Project Explorer.

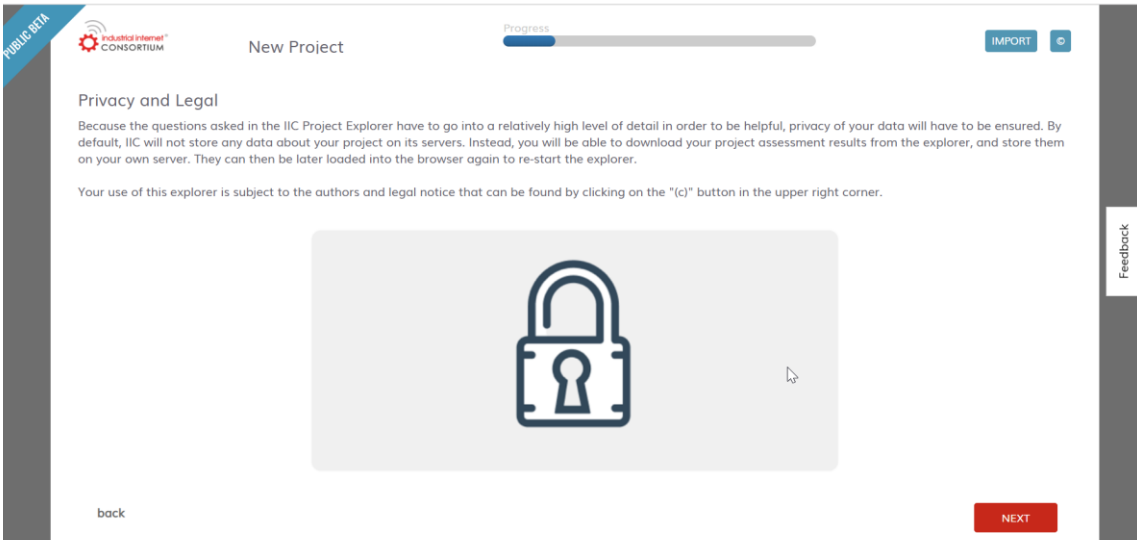

Figure 70: Data privacy in IIC Project Explorer

The IIC Project Explorer is storing all project-related data in the user`s browser, not on an IIC server. This means that IIC does not have direct access to this sensitive information.

4.3 IIC Project Explorer: Levels

One important choice the user must make in the IIC Project Explorer is to define the level of detail used for the assessment of this project. This is shown in Figure 71.

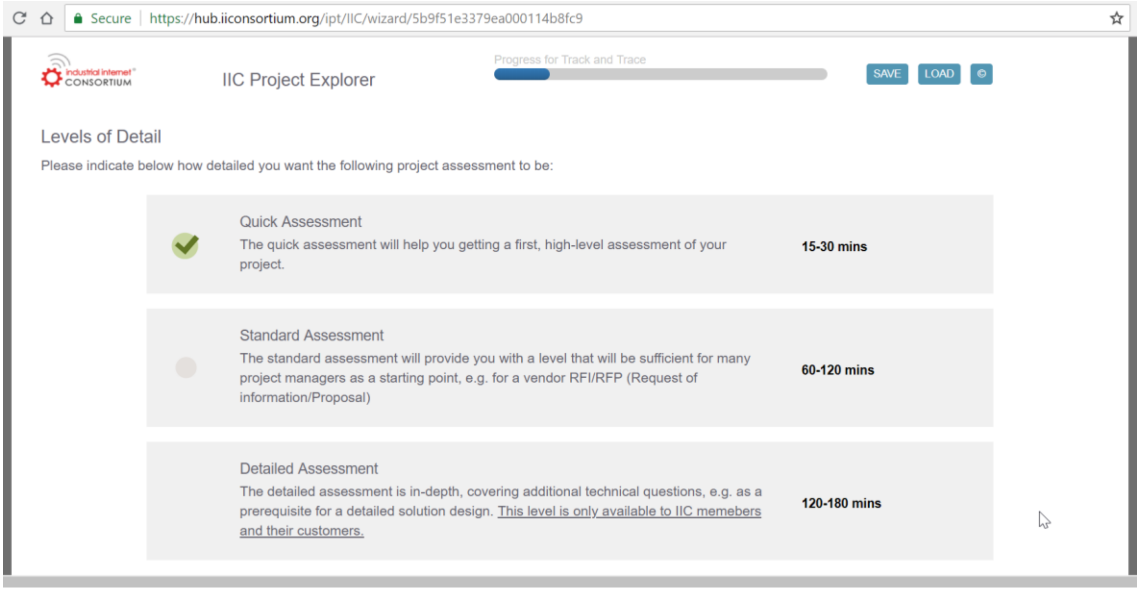

Figure 71: Assessment levels in IIC Project Explorer

The IIC Explorer offers three levels. The first level is a “quick assessment,” which is designed to support a high-level assessment of the project. The second level is the “standard assessment,” which could be used, for example, as the starting point for creating the nonfunctional requirements of an RFI/RFP. The third level is the “detailed assessment,” which is intended, for example, as the foundation of a detailed solution design.

Especially levels 2 and 3 can also be used in requirements management workshops with multiple stakeholders in the room. In this case, the tool would be used to help structure the discussion and to capture the results. This was done as part of the research with the IIC German Country Team using the “manual” version of the IIC IPT, as described in section 2.4.2, Figure 21. In this case, the IPT cards were used with the research team to capture the results of an example project. This process took approximately a half day. In practice, requirements capturing workshops can take multiple days. Having an online tool like the IIC Project Explorer to help structure the workshop and capture results could be quite valuable. Depending on the time available for the workshop and the types of participants, a matching detail level must be chosen.

The support for different assessment levels is described in the IgniteWorx design proposal in section 3.3.5.2.

4.4 IIC Project Explorer: IIoT Solution Canvas

Because of the high level of complexity involved in designing an IIoT solution, IgniteWorx proposes the concept of using an IIoT solution canvas to guide the user through the questionnaire (see section 3.3.5.2). An implementation of this concept is shown in Figure 72.

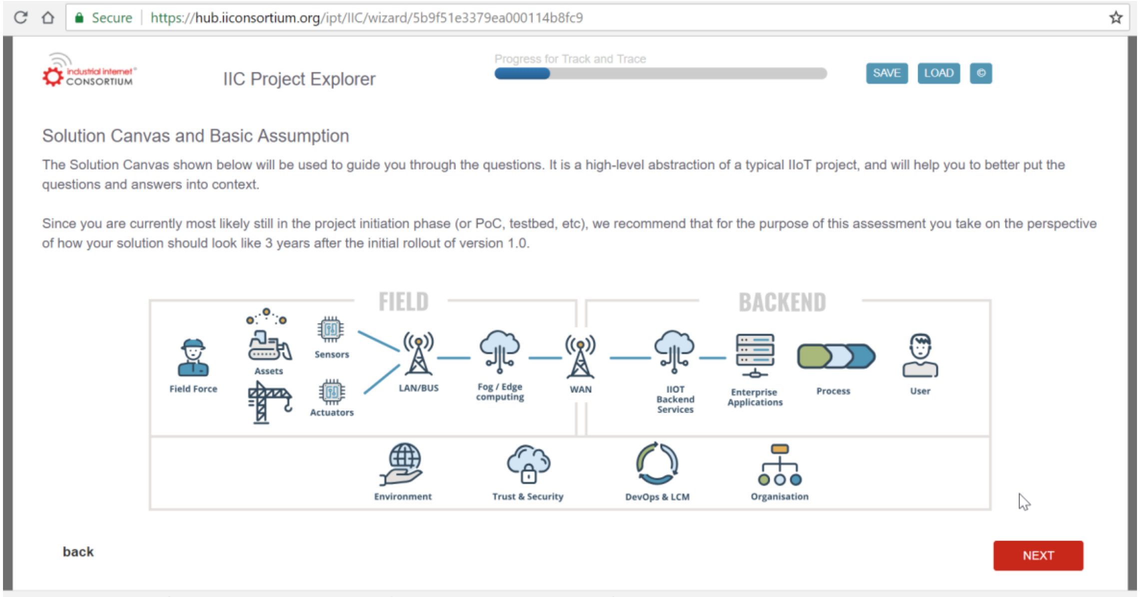

Figure 72: IIoT solution canvas in IIC Project Explorer

The IIC solution canvas for an IIoT project includes the field perspective, the backend perspective, and the general environment perspective. Each of the perspectives features the key elements usually found here (e.g., assets, sensors, and local compute capabilities in the field). This canvas is explained here and then reused for each of the detail questions below to visually guide the user through the set of questions and ensure that he or she always knows where he or she is in the process.

4.5 IIC Project Explorer: Repeatable Categories

In large projects, there can be multiple different instances of key artifacts, e.g., assets and backends. For example, an asset management solution for a construction company might have to support excavators, cranes, and bulldozers.

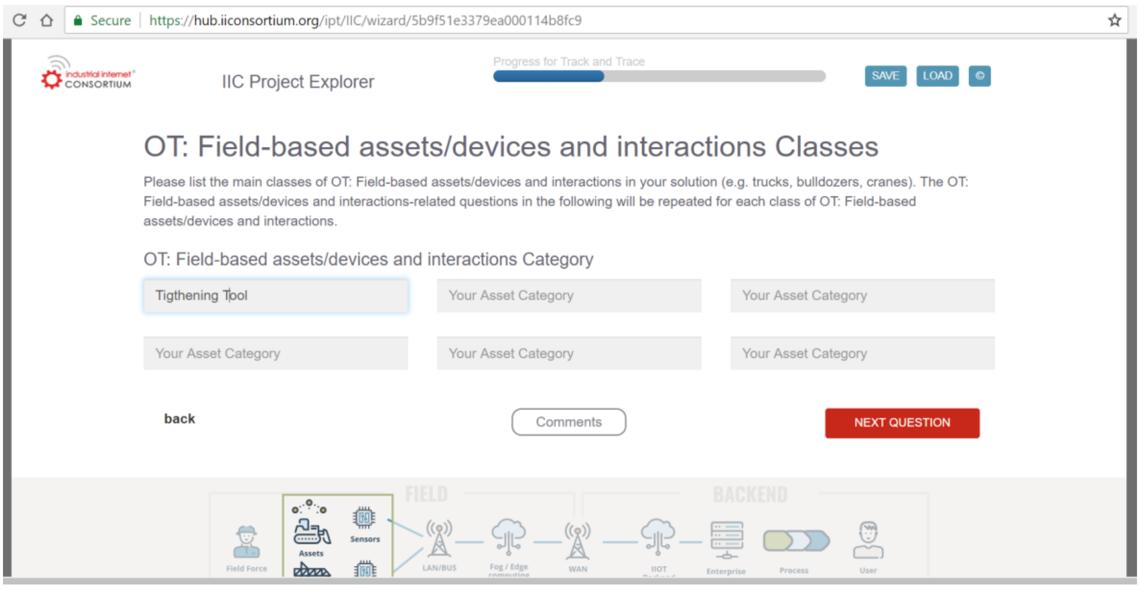

Figure 73: Example for capturing multiple artifact categories in the IIC Project Explorer

The IgniteWorx design proposal is supporting the capturing and management of multiple artifact categories (see the discussion on the "IsRepeatable" attribute in section 3.3.5.2). Figure 73 shows how this concept is supported by the IIC Project Explorer.

4.6 IIC Project Explorer: Project Dimensions

Finally, the capturing of individual project dimensions is a key feature of IgniteWorx and the IIC Project Explorer. As described in section 3.1.1, each Ignite dimension is offering exactly four options to choose from. This was designed to allow for easy processing of the user input in the algorithms that use this data.

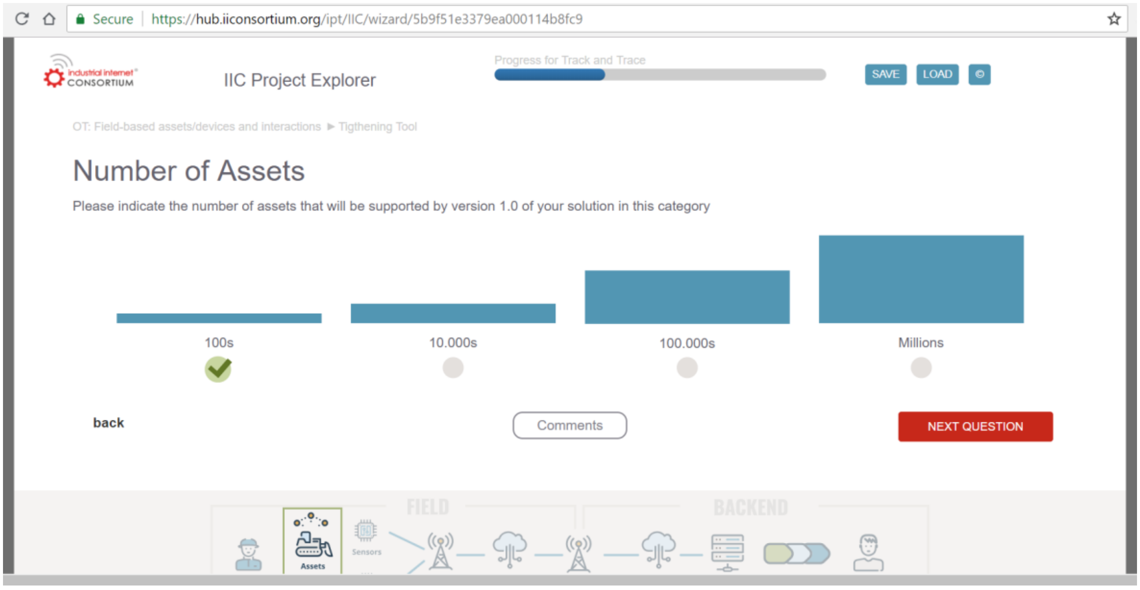

Figure 74: Example of capturing an Ignite project dimension in the IIC Project Explorer

Figure 74 provides an example of capturing an Ignite dimension in the IIC Project Explorer. At the bottom of the page, one can see the solution canvas, including a green box that indicates which solution artifact the current question is related to. This is supported by the IgniteWorx design described in section 3.3.5.2.

4.7 IIC Project Explorer: Assessment Summary

Having completed the project assessment, the IIC Project Explorer offers three main assessment results: a summary, a detailed breakdown of the assessment results, and related readings.

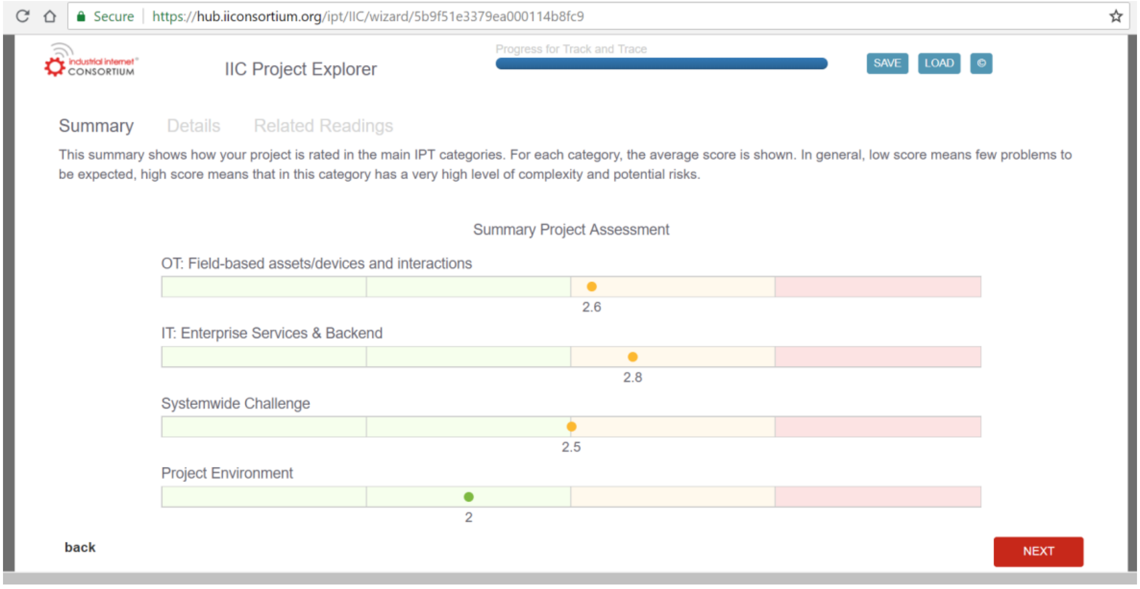

Figure 75: Example of project assessment summary in the IIC Project Explorer

Figure 75 shows an example of the project assessment summary in the IIC Project Explorer. Ignite dimensions are structured so that each answer to a survey question is always rated 1 to 4, where 1 indicates a low level of criticality from the project's point of view, while 4 indicates a high level of criticality (see section 3.1.1). This view is simply calculating the average for the answers in each category, e.g., IT, OT, or systemwide challenges. This provides the project manager with a management summary of the potential criticality of key areas in his or her project.

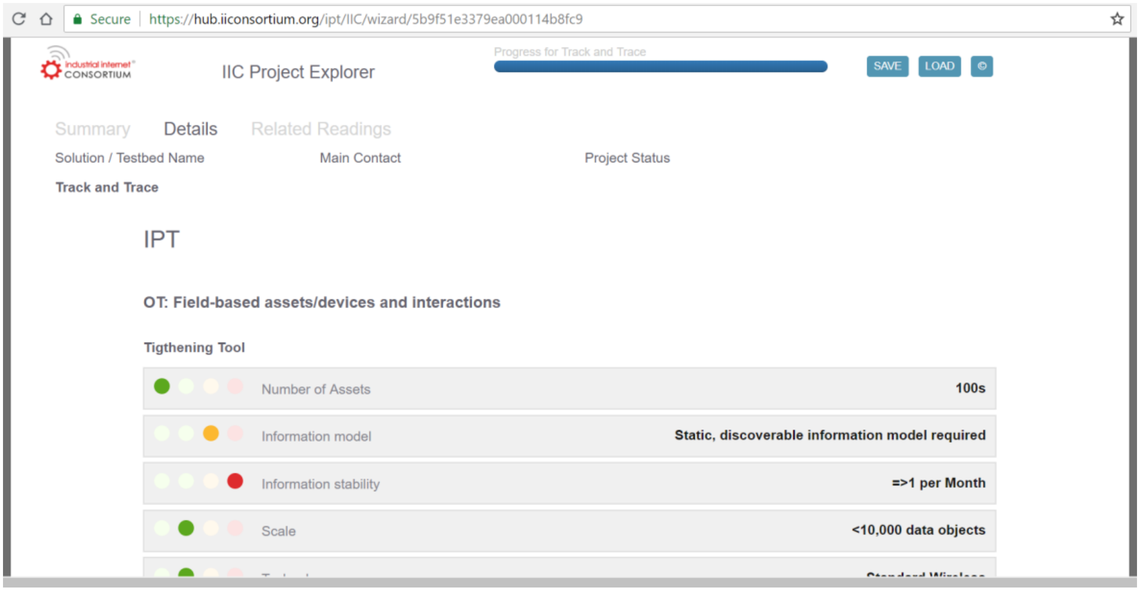

Figure 76: Example for project assessment details in IIC Project Explorer

Figure 76 shows an example of the details of a project assessment. It is essentially a visualization of each Ignite dimension captured as part of the assessment. This view can be interesting, for example, for the creation of the nonfunctional requirements of an

RFI/RFP. Consequently, the IIC Project Explorer also supports the download of a Word-based RFP-template, which includes this information.

4.8 IIC Project Explorer: Project Setup Recommendations

Finally, the IIC Project Explorer offers a view that provides the user with concrete recommendations for his or her project setup. In this case, this is actually not done as a direct recommendation but rather as so-called "related readings." For an organization such as IIC, it would be very difficult to actually provide direct recommendations, for obvious legal reasons.

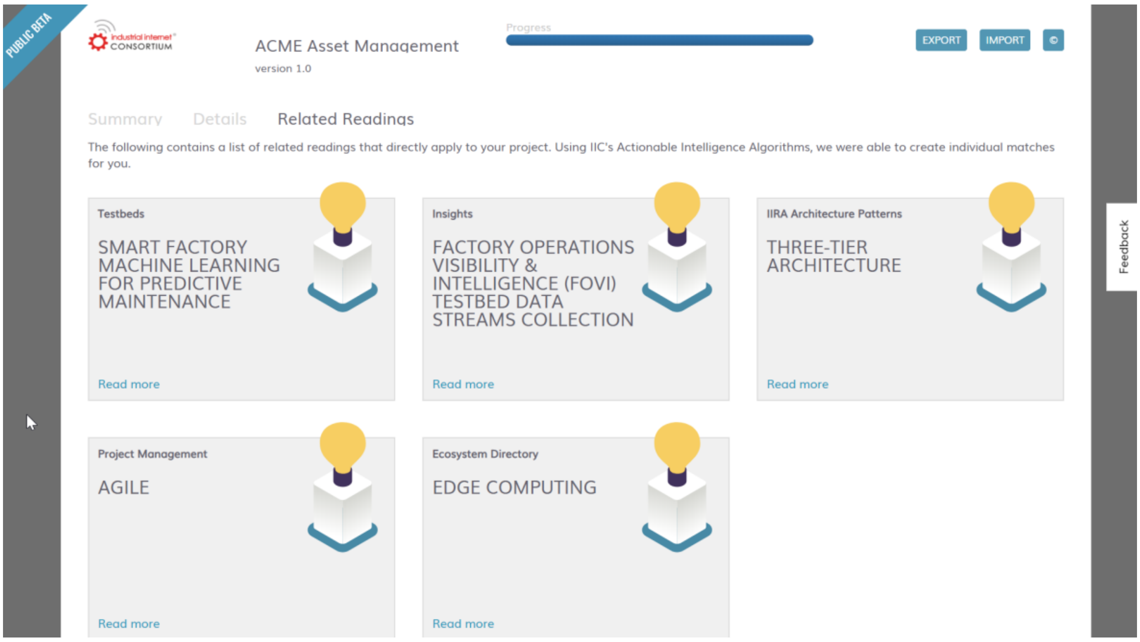

Figure 77: Recommendations for related readings for IIoT project setup

Figure 77 provides a concrete example of the recommendations for the related readings for an IIoT project setup provided by the IIC Project Explorer. The IIC Project Explorer is structurally following the IgniteWorx design proposal of result sets, as described in sections 3.3.5.4 (system design). However, the proposal for the IgniteWorx result set content structure (as described in section 3.1.2) is only used partially. The reason is that IgniteWorx looks at how the ideal content structure would look based on the research output of this thesis; the IIC must map the result sets concept to its existing knowledge base and support key IIC concepts, such as testbeds.

Result Set	IgniteWorx	IIC	Comments
Project Management Methodology	✓	✓	Very similar, both based on Sassikumar (2018)

Solution Design	✓	✓	IIC is actually providing different architecture patterns here.
Technology Selection	✓	✓	IIC is using its ecosystem directory to support this topic.
Resource Acquisition	✓	×	Currently not supported by IIC Project Explorer
Cost Estimate	✓	×	Currently not supported by IIC Project Explorer
Risk Management	✓	×	Currently not supported by IIC Project Explorer
Trust and Security	✓	×	Currently not supported by IIC Project Explorer
Reliability and Resilience	✓	×	Currently not supported by IIC Project Explorer
Verification and Validation	✓	×	Currently not supported by IIC Project Explorer
Service Operations	✓	×	Currently not supported by IIC Project Explorer
Testbeds	×	✓	Testbeds are a key asset from IIC, which provide relevant benchmarking information. IgniteWorx cannot make an assumption about the availability of such information.
Insights	×	✓	Similar to testbeds, this is a valuable but IIC-specific content category.

Table 26: Comparison of result sets (IIC versus IgniteWorx)

Table 26 provides a detailed comparison of the proposed IgniteWorx results sets versus the result sets actually supported in the public beta of the IIC Project Explorer.

4.9 Recommendation Details

In the IIC Project Explorer, each recommendation can be looked at in detail, as shown in Figure 78. For each recommendation, the following are shown: first, a synopsis of the actual "related reading," including a link to the actual page in the IIC knowledge base.

Second, a description of why this recommendation was made, consisting of a detailed list of the Ignite dimensions and how they were ranked for the project. This feature is actually conceptually close to the explanation subsystem of a typical expert system, as described in section 3.3.2.2.

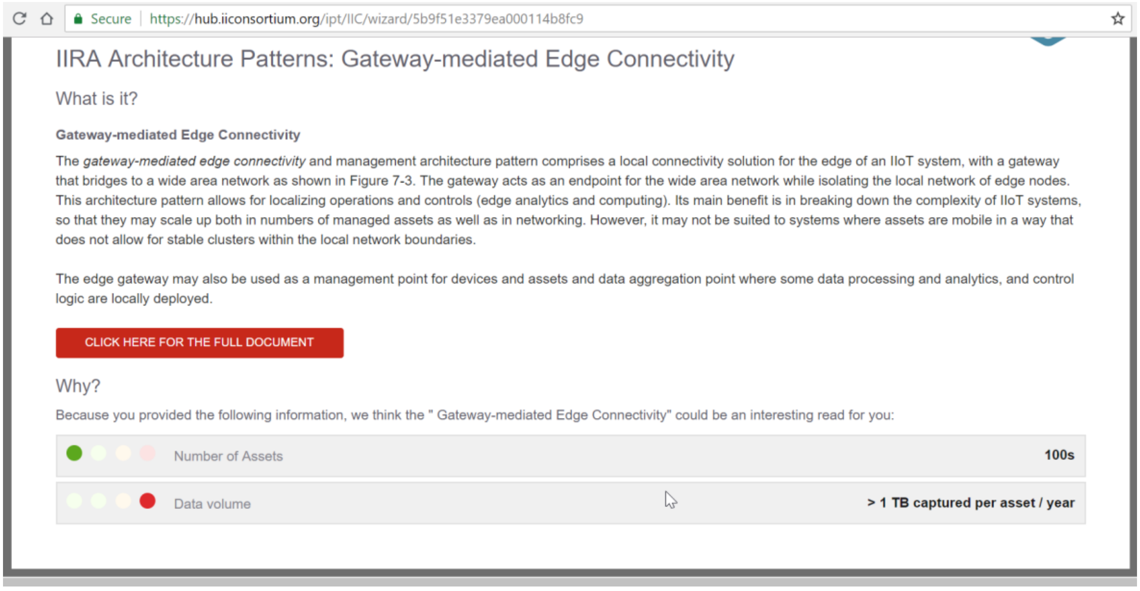

Figure 78: Recommendation details in the IIC Project Explorer

4.10 Summary

As can be seen in this section, the IIC Project Explorer—while not a 1:1 implementation of IgniteWorx—is a good demonstration of how the concepts of IgniteWorx can be implemented in an industrial setting.

The main differences between the IIC Project Explorer and IgniteWorx can actually be found on the content side. While IgniteWorx proposed ten results sets, the IIC Project Explorer in its initial beta version provides only six, which are also structurally slightly different than the ones proposed by IgniteWorx. As discussed in section 4.8, this is due to the fact that IIC has certain limitations from a legal point of view and has additional artifacts like IIC testbeds, which could not be assumed in this thesis.

However, despite these differences, it is assumed in the following evaluation that the IIC Project Explorer is a successful demonstration of the key concepts outlined by IgniteWorx because it shows how the general IgniteWorx concepts can be applied to a real-world application. This is especially true because these are not structural differences but mainly relate to details on the content level.

5 Evaluation I

As outlined in the introduction, the evaluation phase of the first iteration of this project is based on case study research. First, the specific case-study research approach is described before then providing an overview of four detailed cases, followed by a discussion of cross-case findings and their implications for IgniteWorx.

5.1 Goals and Approach Taken

According to Venable et al. (2016), “Evaluation of design artifacts and design theories is a key activity in Design Science Research (DSR), as it provides feedback for further development and (if done correctly) assures the rigour of the research.” The authors are proposing a “Framework for Evaluation in Design Science” (FEDS). The FEDS evaluation design process defines four steps:

1. Explicate the goals of the evaluation
2. Choose the evaluation strategy or strategies
3. Determine the properties to evaluate
4. Design the individual evaluation episode(s)

The passages below follow a process inspired by this proposal and outline/explicate the goals of the evaluation. The following sections explain how the evaluation strategy was chosen and which properties to evaluate and then describes the design of the individual evaluation episodes (project interviews, in this case).

The main goal of the evaluation is to see if the IgniteWorx approach is useful in principle and to get input to help improve the relevance for project managers. This is a first step toward validation. However, a complete and formal validation is not possible in the scope of this thesis. Consequently, the second iteration of the IgniteWorx system design actually aims to build elements into the system, which should help with continuous and incremental evaluation.

The focus of the initial evaluation is on the three main elements: Ignite project assessment, IgniteWorx result sets, and IgniteWorx matching rules. Please note that evaluation I is not evaluating the IIC implementation, but rather the structure and proposed initial contents of these three elements, as defined by this thesis. An evaluation of the IIC implementation is provided in evaluation II.

For each of these three elements, the following must be clear:

- Is the scope and value proposition of the design artifact suitable to the needs of industrial users?
- Does the design artifact have the right level of granularity?
- Is the structure of the artifact the right one?

- Are key elements missing, redundant, or irrelevant?
- Can the design artifact function in the real world?
- Does it play well together with the other related design artifacts?

To evaluate this, the following section presents a case study design as the foundation for the evaluation strategy.

5.2 Case Study Design

Case study design is a wide research field. Yin (2003) distinguishes between holistic case studies and embedded case studies. This is significant in the context of this project since the related studies must be done in the context of IgniteWorx to achieve a result that supports this phase of the design-science research-based approach.

5.2.1 Holistic versus Embedded Case Study

According to Yin (2003), holistic case studies analyze a case as a whole, while embedded case studies look at multiple units of analysis within a given case. Figure 79 provides an overview of the different approaches, based on Runeson and Höst (2009).

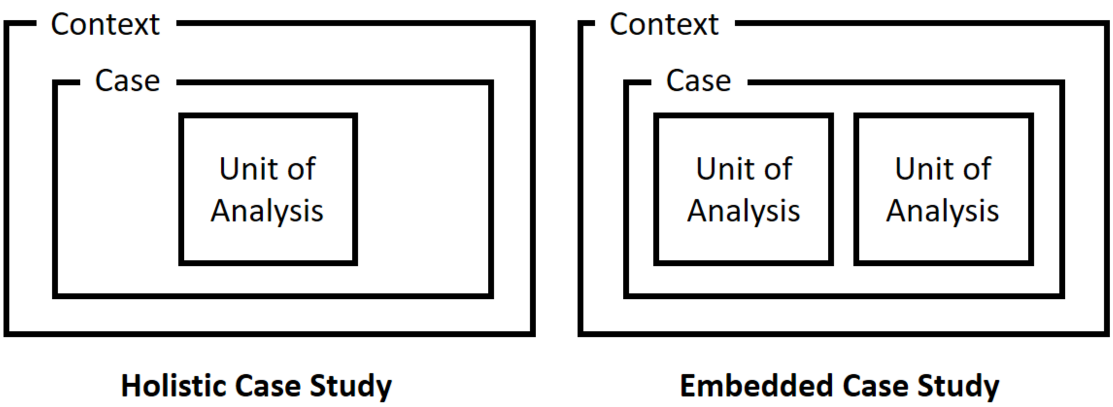

Figure 79: Holistic versus embedded case study

5.2.2 Approach Taken

The approach taken in this thesis is based on embedded case studies because the context for the cases is the IgniteWorx framework. This is consistent across the different units of analysis (see Figure 80).

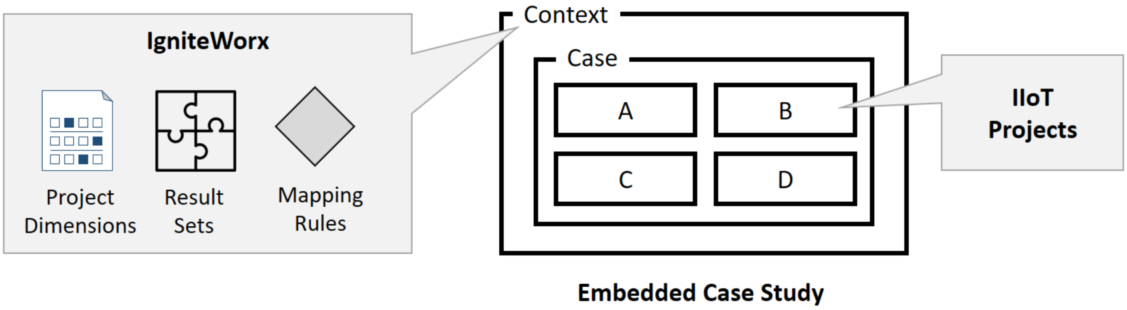

Figure 80: IgniteWorx as embedded case study

The context chosen here is IgniteWorx. This is managed consistently across the different units of analysis. For each unit, the following focuses especially on the Ignite project dimensions, the IgniteWorx result sets, and finally the mapping rules.

5.2.3 Project Selection

To provide the ideal support for the case study, a number of projects were selected to help advance the research on the initial IgniteWorx design proposal.

A number of factors were considered for the project selection:

- Accessibility and commitment for support: This may be the most basic but also the most important factor. If the project cannot be reached or is not willing or able to support the research, the project can be ruled out
- IIoT fit: The project must fit the definition of an IIoT project, as provided in the introduction on page 19.
- Domain: The projects should come from different IIoT domains to ensure that the evaluation is not domain-specific.
- Maturity of the project: IgniteWorx is designed to support projects at the very beginning of their lifecycle. However, for the evaluation, it is better to work with projects that are more advanced so that the experts can provide feedback from real-world experience (looking backward at the project initiation phase).
- Scalability: The method described here looks for projects that work at different scales, e.g., with respect to number of assets supported.

Based on these factors, the following projects have been identified and acquired:

Project	Description	Challenges that make it relevant for the case study
A: Automated Optical Inspection	Application of AI in manufacturing for inspection processes	Advanced use of AI, combined with IIoT and robotics

B: Tightening Quality Assurance Systems	Application of ML in assembly lines for process tools	Edge/fog-computing use case with intelligent data collection in the backend
C: Remote Maintenance	Remote monitoring and maintenance in highly heterogeneous IoT environments, e.g. commercial building operations	Good example of dealing with high level of heterogeneity
D: OTA in Automotive	Over-the-air updates for vehicles	Key feature to support more agile development in this industry

Three of the four projects are fairly advanced in their lifecycles with real-world deployments and large numbers of assets supported in the field. This has the benefit that the project experts can provide real-world feedback on IgniteWorx. However, it also means that the project experts must put themselves in the role of somebody who is in the early phase of his or her project because this is the role that IgniteWorx usually plays. This workaround was seen as able to sufficiently address the issue described here. The benefit of real-world expertise and feedback was seen as outweighing the disadvantages.

5.2.4 Goals for Project Documentation

Four goals for the project documentation have been identified:

- Project domain: Understand and describe the project domain in general to have sufficient context information
- Solution design: Understand and describe the solution design in some detail
- Lessons learned: Describe the generic lessons learned from the project
- IgniteWorx-specifics: Describe specific lessons learned that apply to IgniteWorx

The following describes how literature research and standardized, open-ended interviews are utilized to generate the required data to achieve these four goals for the project documentation.

5.2.5 Literature Research

To better understand the project domain, intensive literature research (both industrial and academic) is required. This is the foundation for the general description of the project domain for each project.

This research was performed based on the proposal from Budgen and Brereton (2006), who describe a process for "Performing Systematic Literature Reviews in Software Engineering."

5.2.6 Interviews

To analyze each project`s solution design and to derive the lessons learned, interviews were chosen as the main research instrument. They are a well-established tool to collect qualitative data and are frequently used in case study research. See Yin (2003), Merriam (1988), and Eisenhardt (1989), for example.

Interviews as a research tool are not new. As early as 1957, Kahn and Cannell (1957) described "The dynamics of interviewing; theory, technique, and cases." The described techniques have, of course, evolved over time. Comprehensive guidelines are provided, for example, by US General Accounting Office (2006), Dick (2002), and McNamara (2009). Based on these guidelines, the following section looks at interview types, interview design, interviewee selection, and interview conduction.

5.2.6.1 Interview Types

McNamara (2009) describes four main interview types:

- Informal, conversational interview: Open interview without prepared questions. Allows the interviewer to adapt to the interviewee's nature and priorities.
- General interview guide approach: This approach ensures that the same general areas are covered in each interview. It is a more focused approach but still with a relatively high degree of flexibility.
- Standardized, open-ended interview: A set of open-ended questions is prepared and used in all the interviews. Open-ended means that there is no set of predefined answers or options to choose from. According to McNamara (2009), "this approach facilitates faster interviews that can be more easily analyzed and compared."
- Closed, fixed-response interview: All interviewees are presented with the same questions and presented with a set of answers to choose from.

For this case study, the standardized, open-ended interview style was chosen because it seems best suited for capturing feedback on the research subject—the design of IgniteWorx—without limiting the interviewees too much in sharing their personal experiences.

5.2.6.2 Interview Design

US General Accounting Office (2006) describes a process for the identification, development, and selection of interview questions, as well as a standard procedure for composing and formatting them. The report differentiates among descriptive, normative, and cause-and-effect questions. After the development of the broad overall questions, US General Accounting Office (2006) describes how they are translated into measurable elements in the form of hypotheses or questions. The composition of the appropriate questions must consider the relevance, the selection of the respondents, and the ease of response. Again, different types of questions can be used, including open and closed. Finally, the organization of the questions is another main task in the interview design: "Present the questions in a logical manner, keeping the flow of questions in chronological or reverse order, as appropriate. Avoid haphazardly jumping from one topic to another." Finally, the right layout of the questions must be considered as well.

Another important part of interview design related to avoiding potential problems, including selection of the right language (appropriateness, level, use of qualifying language, clarity, avoidance of double negatives, etc.) and avoiding bias within questions; see (US General Accounting Office, 2006) and (McNamara, 2009).

For this case study, the interview guide was designed to match the main artifacts of IgniteWorx, namely, the Ignite project assessment dimensions, the IgniteWorx result sets, and the matching rules. Figure 81 shows the three main areas of the interview guide.

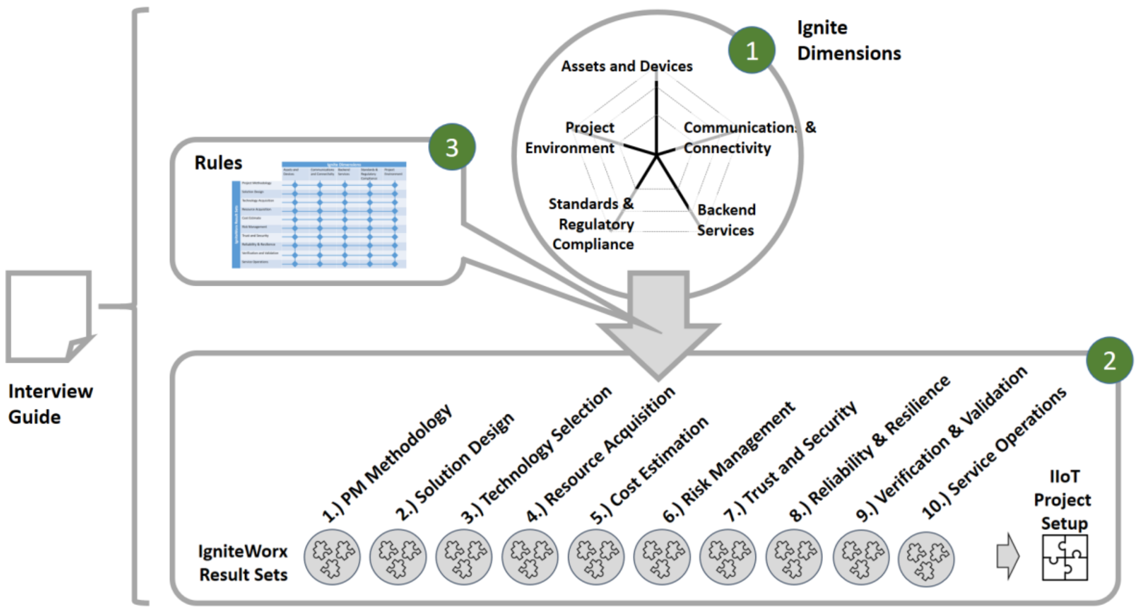

Figure 81: Structure of interview guide

These three main interview areas are presented sequentially. The order was defined as follows:

1. Ignite Dimensions: This is the first part of the interview guide since this is also the starting point from an end-user's point of view (he or she starts with the project assessment).
2. IgniteWorx Result Sets: This is the second part of the interview guide since this is the result that the end-user sees after doing the assessment.
3. Matching Rules: This is the third and last part because it requires some explanation of how the rules help in mapping the dimensions to the result sets. That would be difficult to explain without having introduced the result sets first.

Each interview area has the following parts:

- Feedback on the respective elements, e.g., concrete dimensions or result sets
- Key aspects from the perspective of the concrete project in the scope of the interview
- General observations

This is repeated for all elements of each area. At the end of each section, the interview guide asks if there are elements missing (e.g., missing dimensions, missing results sets).

For Ignite dimensions, the five main areas are included. For result sets, the ten are included. For matching rules, two concrete examples are provided.

Each area is mapped onto a matrix, with elements on one axis and project-specific/general observations on the other. Each cell in each of these matrixes is an interview question. The number of questions in the interview guide can be derived as follows:

Area	Number of Questions	Total
Ignite Dimensions	5x2+2	12
IgniteWorx Result Sets	10x2+2	22
IgniteWorx Rules	2x2	4
		38

Table 27: Number of questions in the interview guide

Thirty-eight is a relatively high number of questions, given that the planned interview durations are between 90 and 120 minutes. However, since the second column of each matrix (the general observations) is optional, this means that only 19 questions are mandatory (meaning that they should be answered, if possible at all, which also might not always be the case).

5.2.6.3 Interviewee Selection

Interviewee selection is another key part of the interview research design, especially for interviews with a broad reach. However, US General Accounting Office (2006) also describes, “For some structured interviews, because there is only one person who fits the category of interviewee […], no selection process is needed.” In the case of this project, the most likely candidates for the interviews are the project managers or product managers since they typically have the most significant insights and a broad overview over the entire project. In some cases (like in project B here), it can make sense to interview experts from the IT and the OT sides.

Project	Interviewee	Reason for Selection

A: Automated Optical Inspection	Project Manager	Has deep insights into both product design as well as team structure and performance
B: Tightening Quality Assurance Systems	Interviewee B1: Product Manager Backend Solution	Represents the backend part of the solution with many years of experience
	Interviewee B2: Product Manager Tightening Tools	Represents the asset perspective in this project (in this case, asset means tightening tool)
C: Remote Maintenance	Product Manager	Has the best overview of all project aspects
D: OTA in Automotive	Solution Architect	Is involved in solution design and implementation aspects

Table 28: Interviewee selection for integrated case study

5.2.6.4 Conducting the Interview

McNamara (2009) provides the following advice for conducting the interview:

- Occasionally verify that the tape recorder (if used) is working.
- Ask one question at a time.
- Attempt to remain as neutral as possible.
- Encourage responses with occasional nods of the head, “uh-huh,” etc.
- Be careful about the appearance when note taking.
- Provide transitions between major topics.
- Do not lose control of the interview.

Actually, most interview guides recommend recording and then transcribing the interview. In the experience of the author, this approach is very difficult to implement with the interviewees targeted here; project managers and product managers of large industry firms are dealing with many different stakeholders and tasks and are constantly under time pressure. Getting them to do an interview and then *also* approve a lengthy transcript is extremely difficult. Finally, they might not openly address more difficult topics if the interview is recorded.

To address these issues, for the interviews here, a different approach was selected. During the interview, the interview guide is shared between the interviewer and the interviewee on a laptop or a projector. The answers to the questions are transcribed in real time so that approval can be given directly at the end of the interview. This should ideally also increase the validity of the interviews since the approval is given instantly and there will be no discussions about details and semantics in the aftermath. Of course, there are also potential limitations, especially the fact that important details might not be recorded or that context information like intonations are lost. However, for the purpose of this thesis, it is assumed that the advantages of this approach outweigh the disadvantages.

5.2.6.5 Interview Evaluation

US General Accounting Office (2006) states that “the answers to open-ended questions are often left unanalyzed. The evaluator or auditor in reporting may quote from one or a few selected responses, but open-ended interviews generally do not produce uniform data that can be compared, summed, or further analyzed to answer the evaluation or audit questions.”

Consequently, the interview evaluation in this case actually aims to get as much circumstantial evidence for the validation of the different elements of IgniteWorx with all the limitations of this approach—as described, for example, in “The Nature of ‘Evidence’ in Qualitative Research Methods” (Miller and Fredericks, 2003)—e.g., with respect to the application of the “hypothetico-deductive model” to qualitative research cases.

The interview evaluation is actually done on two levels in this case study:

- Individual project analysis: For each individual project analysis, the interview input is used for the general “lessons learned” part, as well as the “findings for IgniteWorx” part.
- Cross-Study Findings: Across all projects, the findings for IgniteWorx are synthetized and summarized.

5.3 A: Automated Optical Inspection

Project A of the case study is a project to create a highly flexible automated optical inspection (AOI) tool for manufacturing, using advanced, AI-based image processing.

AOI is not a new discipline. For example, Jones (1927) already describes a patent for an optical inspection system, designed to support “sorting according to size measured by light-responsive means.” A modern example for the application of AOI technology is the market for printed circuit boards (PCBs). Gilutz (1988) encourages the use of AOI for PCB manufacturing as follows: “Early detection of improper artwork, leftover resist flakes on imaged panels or over-etched or under-etched copper lines can support prompt identification and retuning of a malfunctioning process and prevention of repeat production of faulty layers. Further, early flaw identification makes it easier and less costly to repair the flawed panel or to scrap it and to prevent the cost associated with fruitless further processing and handling. Meeting this goal requires an inspection technique that will handle the different mid-manufacturing products, support efficient flaw data collection and analysis, and have a cost-effective operation at production rates."

Pattern recognition plays an important role in AOI. In 1982, Fu (1982) already described three major approaches for pattern recognition for automatic visual inspection, namely, template matching, a decision-theoretic approach, and a structural and syntactic approach.

In the early 2000s, scientists looked at the application of neural networks to AOI. For example, Belbachir et al. (2005) proposed "an automatic optical inspection system for the diagnosis of printed circuits based on neural networks." Similarly, Acciani et al. (2006) look at the "application of neural networks in optical inspection and classification of solder joints in surface mount technology."

Today, two main factors are driving the evolution of AOIs:

- Enhancements in image acquisition
- Enhancements in neural networks-based image processing

Each of these two factors is examined in the following.

5.3.1 Image Acquisition in AOI

Depending on the material subject to the inspection, different image acquisition technologies can be used. For example, classic AOIs for PCBS would use a top view camera with telecentric optics (Richter et al., 2017). To optimize the image quality for different materials and surfaces, different illumination modules can be deployed. According to Richter et al. (2017), the most common options include user-controlled color, ultraviolet, and infrared illuminations, as well as coaxial light. The illumination modules can be deployed in different positions to optimize the illumination of the field of view.

Other image acquisition technologies include x-ray tubes with detectors or angled view cameras (Janczki et al., 2013). In addition, 3D AOI technology is now emerging. Koh (2009) describes how advanced 3D AOI technology is overcoming issues such as "shadowing problems, measuring range, and soldered PCB warp and distortion":

> The AOI imaging head employs a series of eight white projection lights mounted in a circular configuration to pick up visual data from all sides and the top of the device or feature. It also contains multiple circular rows of LED illuminators positioned at different heights and angles to the part, each ring firing in succession in different colors, or wavelengths. The camera observes the varying wavelengths and angle of projection of each wavelength onto the part and thus can construct accurate 3D images and measure the height of the component or feature. This configuration allows for completely shadow-free imaging as well as precise position, shape, and height measurement of the feature under inspection.

5.3.2 Image Processing in AOI

Modern image acquisition technology in AOI is delivering very high volumes of data. For example, Ye et al. (2018) describe an “AOI technology based on a high-resolution linear charge-coupled device (CCD) to implement surface defect inspection over wide area samples.” The data volume in this system is described as “147 megapixel image data per second transmitted from a host CPU to a graphical processing unit (GPU) device for image processing.” According to the paper, the “2D inspection system accurately detects microscopic flaws on touch panel glasses in less than 1 s.”

Recently emerging high-end GPUs and similar technologies are enabling advanced image classification algorithms based on artificial intelligence or machine learning. Deep convolutional networks enable a new quality of image classification. These networks take the entire image as input and extract the features of the image implicitly. The network is self-learning, meaning that the features of the image to be extracted are not explicitly defined by an engineer any longer; the system is trained to recognize them automatically. This training process requires large amounts of high-quality training data (Richter et al., 2017).

5.3.3 Project A: AI-Based Optical Inspection System

AOI is an important tool for quality assurance in manufacturing. However, especially for large manufacturers, it requires a high level of flexibility, given the potentially extremely high number of combinations of products, variants, and defect categories.

Especially for high-volume manufacturers, automating visual inspection can be very attractive due to the otherwise high cost involved. Today, most automated inspection machines are highly specialized, which has a number of disadvantages:

- The solution is usually specific to one product category
- Development of these machines can make up a significant part of overall engineering costs
- It can take months to design and build such a special-purpose inspection machine

The goal of this project is to develop an optical inspection machine that supports fully automated, flexible, and product-independent product inspection.

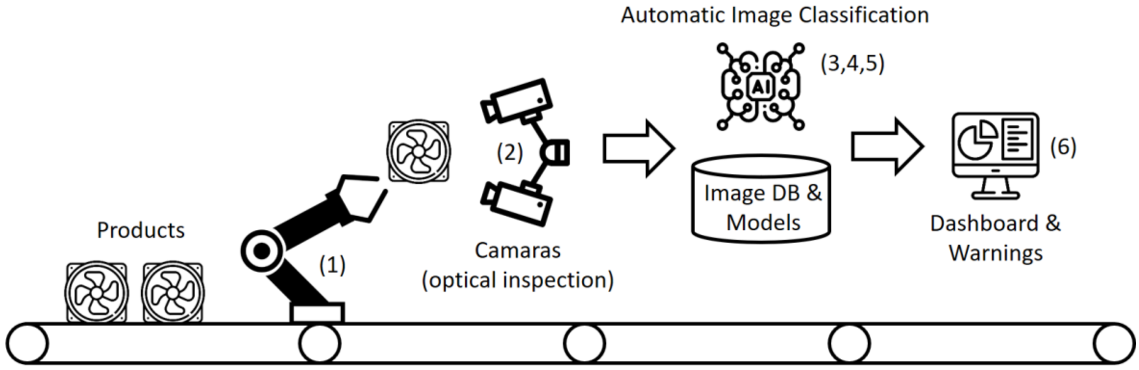

Figure 82: AI-based optical inspection

Figure 82 provides an overview of the solution architecture, which includes:

1. Robot support for product picking and presentation
2. Multi-camera photometric imaging
3. Automatic image classification through use of artificial intelligence/machine learning
4. Easy training and setup for new product categories, variants, and defect types, performed by manufacturing staff on the factory floor
5. Classifier stability, i.e., the automatic inspection process is not disturbed by small changes in the training data
6. Defect visualization and augmentation for inspection

This project is currently in transition from advanced development to a line of business. A number of systems have been developed and successfully tested in manufacturing lines.

5.3.4 Lessons Learned

This project is a combined hardware/software IIoT project, which is run based on an agile/SCRUM approach, slightly adapted to the needs of hardware development. Key lessons learned from this project include:

- In principle, SCRUM also works well for hardware development
- However, more long-term planning is required due to hardware constraints
- Incremental development is difficult for hardware, especially after it is deployed in the field
- Fixed sprint duration and implementing the “shippable product increments” approach is more difficult for projects which are including hardware development
- Pilot runs in production enforce collaboration among all branches

5.3.5 Findings for IgniteWorx

Regarding the Ignite dimensions, some highly relevant feedback was provided, especially in the area of standards and regulatory requirements, as well as the project environment.

Since the solution has all intelligence directly deployed in the edge layer (see IIC (2018b)), the feedback on the distribution of intelligence between the edge and the backend was minimal (Ignite dimensions #1–#3). The integration aspects here are mainly related to integration with manufacturing execution systems (MES) systems.

Regarding Ignite dimension #4 (Standards and Regulatory Compliance), the feedback was that from the point of view of advanced development, the existing standards of the transfer partner must be taken into consideration (e.g., which controls and robots to support). This is currently not an option in Ignite.

Regarding Ignite dimension #5 (Project Environment), the feedback was that the stability and availability of the team are seen as critical success factors, especially during the transition to the line of business. This feedback would be difficult to reflect directly in an Ignite dimension, but it is valuable input for result set #4 (Resource Acquisition).

Regarding the discussion of potentially missing Ignite dimensions, one input was that the understanding of the complex stakeholder network is highly relevant. In this example, this includes many stakeholders in the development and productization process, as well as many different end-user roles (including system operators, system maintenance team, manufacturing planning experts, and quality managers). Another recommendation was to include a new Ignite dimension to determine the current stage in the project's lifecycle.

Other feedback was regarding the availability and maturity of use cases. It would seem logical to extend IgniteWorx to actually capture such information. The issue here is that, structurally, the current concept of Ignite dimensions would not allow the capturing of such information (since the Ignite dimensions and available options are fixed and do not allow to capture input lists).

Regarding the result sets capturing the IIoT project setup, the feedback from the project also included a number of important points.

Regarding result set #1 (Project Management Methodology), it was confirmed that this is an important one, and guidance on the most suitable methodology (e.g., SCRUM versus SAFE) would be seen as helpful.

Regarding result set #2 (Solution Design), the input provided was regarding decision points and major milestones in the project. It would have been seen as useful if IgniteWorx would be able to support this as part of the project setup. For example, a list of decision points that could be expected, depending on the project characteristics, could be generated. As a concrete example, the decision for use of a robot versus linear rotation axis was mentioned. Again, the current structure of the Ignite dimensions would make this difficult since it would mean being able to capture open lists of requirements, as opposed to a more static project survey as provided by Ignite.

Regarding result set #3 (Technology Selection), the point from above regarding the importance of existing standards from the productization partner was repeated.

Regarding result set #4 (Resource Acquisition), the importance of resource availability was emphasized again. An important point that was raised was to ensure that allocated resources are 100 percent available and not only partially assigned to the project.

Regarding result set #5 (Cost Estimation), the feedback was that for this particular project, the feeling was that it was too early in the project lifecycle to look at this in detail.

Regarding result set #8 (Reliability and Resilience), the feedback focused on the specifics of a manufacturing environment as the target for product deployment. For example, the project has a long list of parts that need to be checked in different conditions, e.g., components with oil, minor differences in components from different suppliers, etc.

Regarding result set #9 (Verification and Validation), it was mentioned that most criteria for the approval of an inspection system are highly specific to the industry of the components evaluated; the validation requirements for parts produced for the automotive industry are different from what can be found in aircraft manufacturing, as an example. In general, this cannot be done without deep domain knowhow.

Regarding result set #10 (Service Operations), it was confirmed that this is another key aspect. Operations of a product in a factory mean that all the involved roles (see above) must be efficiently supported. This is something that already should be anticipated during the project initiation.

During the discussion of potentially missing result sets, a number of points came up that could eventually be mapped to the existing ones.

A detailed discussion of rules to create mappings between dimensions and result sets did not happen due to time constraints.

In general, it can be learned from this interview that:

- The general structure of Ignite dimensions and IgniteWorx result sets was confirmed as useful and comprehensive.
- For some areas, a deep dive would require capturing details like the specific industry (manufacturing in this case) and the supported use cases (e.g., optical inspection) to derive more meaningful recommendations from the project assessment.
- This is something currently not supported in the initial IgniteWorx design and can be seen as a weak point that should be addressed.

5.4 B: Tightening Quality Assurance System

Project B of this case study is an industrial tightening quality assurance system (T-QAS). The following section looks at tightening in manufacturing environments, tightening quality assurance, and finally a T-QAS. Based on this contextual information, the results of the expert interview are presented with respect to the evaluation of IgniteWorx.

5.4.1 Tightening in Manufacturing Environments

Tightening is an assembly technique in manufacturing, widely used for many different parts, e.g., safety belts and steering wheels in the automotive industry (Hermansson, 2016).

There are a number of different methods used. Bolt Science (2018) describes the following ones: torque control tightening, angle control tightening, yield controlled tightening, bolt stretch method, heat tightening, and use of tension indicating methods.

Bolt Science (2018) says about torque control tightening that "controlling the torque which a fastener is tightened to is the most popular means of controlling preload." Angle controlled tightening, on the other hand, is using a fixed tightening angle, e.g., for use with power wrenches, where the bolt is being tightened with a predetermined angle.

In addition to the chosen tightening method, there are a number of other factors that determine the quality of the tightening process, e.g., fastener torque rate and operator position (Radwin et al., 2014), or the materials used, e.g. the fastener coating (Archer, 2009).

Given the mission criticality of the outcome, as well as the complexity of the process, it is no wonder that quality assurance for tightening processes plays an important role, as described in the following.

5.4.2 Tightening Quality Assurance

Tightening quality assurance is a complex process that must consider a large number of parameters. According to Remtulla, et al. (2014), the "current quality standards require that all quality data, such as time and date of manufacturer, operator ID, torque data, process complete, process stops, retries, etc., for each part of an overall complete assembly be recorded."

On the technical level, there are different approaches. For example, Carlin and Lennart (2006) describe a method for quality assurance of screw joint tightening, which is "using programmed data relating to the screw joint geometry, expected frictional conditions, power tool characteristics, a tightening strategy and suitable tightening parameter values."

Hermansson (2016) describes that “not only the screw needs to be in position but the strength of the joint must also be guaranteed. To accomplish this, classical process surveillance is combined with angle monitoring and screws with special features.”

To analyze the quality of an individual tightening process, the tightening system can record torque and angle for the process, from which a tightening curve can be created. This process data can be evaluated against a predefined control curve. For example, Figure 83 shows how such a tightening curve could be contained within an upper and a lower control curve. If the actual process curve is not within the bounds of the control curves, a quality issue must be recorded and addressed on the process level.

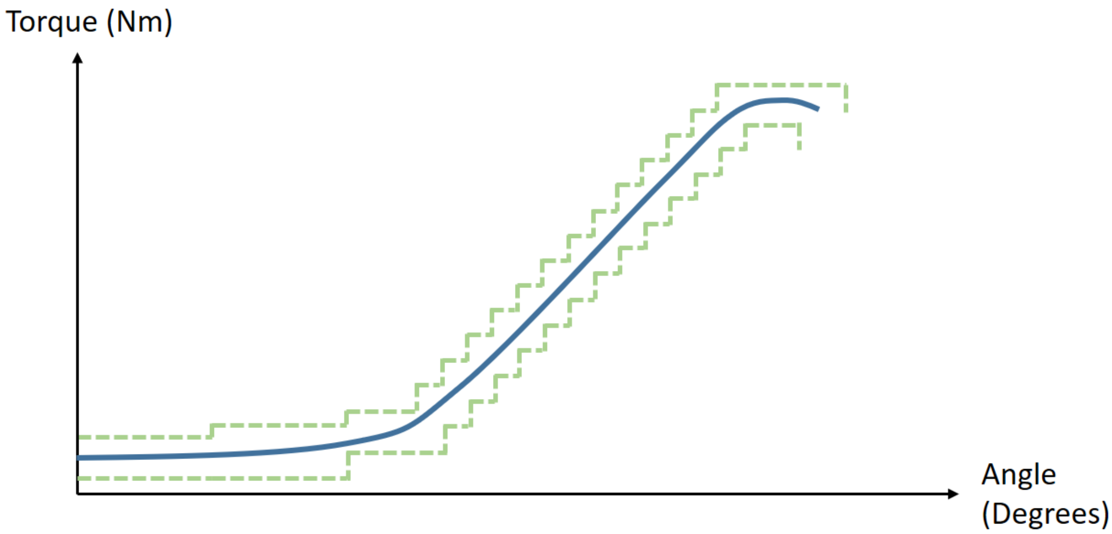

Figure 83: Tightening curve analysis (example)

5.4.3 Project B: Industrial Tightening Quality Assurance System

Project B in this case story is implementing an industrial tightening quality assurance system (T-QAS). According to the expert interview, the system is described as follows. The T-QAS integrates with the different tightening systems deployed on the assembly line. In this particular case, the tightening systems are cordless, battery-operated tightening systems, also called cordless nut-runners. These nut-runners have the intelligence of a complete tightening system onboard. Different tightening programs can be deployed on them, which are used for different steps in the manufacturing process (often with different combinations of materials, threads, coatings, and screws). The T-QAS can activate the right program depending on the process requirements. Also, the T-QAS can download tightening data (e.g., torque and angle) for each tightening process. A machine learning-based algorithm can analyze the data and raise an alarm in case of a process validation (using the control curve mechanism described in the previous section). Figure 84 provides an overview of the T-QAS and how it can be deployed in a factory.

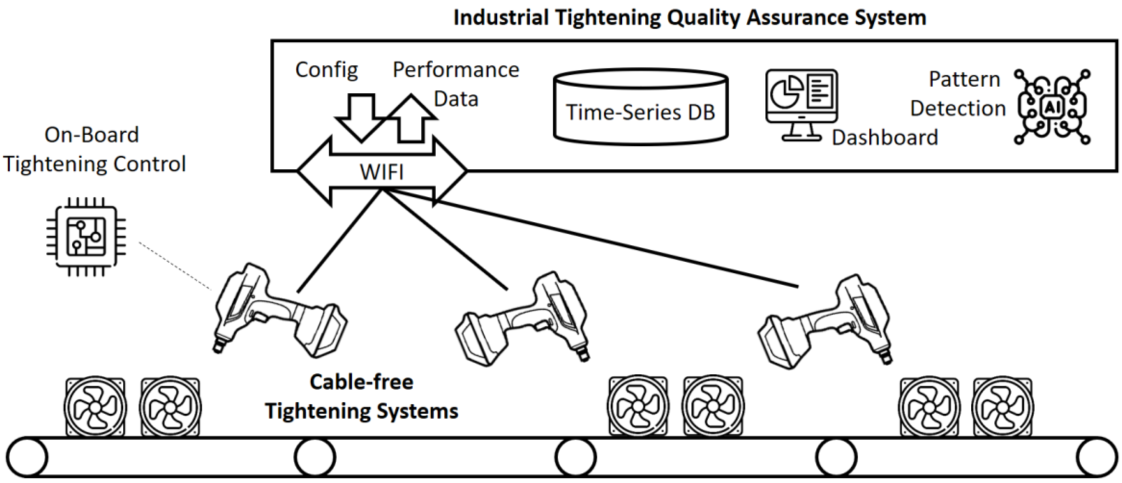

Figure 84: Industrial tightening quality assurance system

5.4.4 Lessons Learned

The asset (cordless nut-runner) is developed by one business unit, the T-QAS by another. In this research, the focus was actually on the T-QAS side. However, the T-QAS could not be built without having the required functionality and interfaces on the nut-runner. This led to a project setup that is also described in Slama, Puhlmann et al. (2015): the T-QAS project was focusing on the backend solution, with a dedicated "asset preparation" project happening in the nut-runner organization in parallel.

A key challenge for the T-QAS project was to balance frequent customer requests, on the one hand, and the evolution of the nut-runner functions and interfaces on the other. Since T-QAS is developed as a standard solution deployed with multiple, independent customer environments, a stringent yet sufficiently agile development process had to be set up. The team chose a strictly agile, six-week release cycle. Another key challenge was to build up a development team with software development and tightening knowhow.

5.4.5 Findings for IgniteWorx

This project is developing a standard QAS solution for industrial tightening systems, which is then deployed with different end-customers. This perspective is not a perfect fit for the current version of Ignite since Ignite does not differentiate between the solution provider and the solution operator perspective; this is currently merged into one. Consequently, for the purpose of this interview, the interviewee took on the perspective of a typical customer project. This worked out quite well in the IgniteWorx simulation since the QAS team must naturally take on this perspective, e.g., when defining requirements about the deployment of nut-runners in the factory.

Regarding the feedback on Ignite dimensions, the following could be observed:

- Ignite dimension #1 (Assets and Devices): The current options seem to capture most aspects of the project, including advanced topics such as energy supply (in this case, battery support for handheld tightening systems).
- Ignite dimension #2 (Communication and Connectivity): All required aspects could be captured here.
- Ignite dimension #3 (Backend Services): All required aspects could be captured here, including advanced topics such as integration with external systems.
- Ignite dimension #4 (Standards and Regulatory Compliance): One topic where Ignite is currently not differentiating is external standards versus newly emerging standards as output of the project (e.g., capturing the importance of setting standards as part of the project survey). Also, the importance of capturing industry domain-specific requirements was mentioned.
- Ignite dimension #5 (Project Environment): One requirement that came up here was the need to potentially differentiate between the asset-related project (tightening tool, in this case) and the IIoT solution environment to capture potential dependencies (e.g., in terms of development speed).
- No feedback for additional/missing Ignite dimensions was given.

Regarding the IgniteWorx result sets, the following feedback was captured:

- Result set #1 (Project Management Methodology): Confirmed as an important choice during project setup (although agile seemed assumed as the default).
- Result set #2 (Solution Design): The impact of the business model on the solution design was highlighted; furthermore, a micro-service architecture was assumed. Capturing this kind of information would (again) require the ability to capture open lists, as opposed to static questions in Ignite.
- Result set #3 (Technology Selection): Confirmed as an important part of project setup (data storage technology was explicitly mentioned).
- Result set #4 (Resource Acquisition): Confirmed as important part of project setup. Team skills along the value chain (development, operations, support, consulting) were explicitly mentioned.
- Result set #5 (Cost Estimate): Confirmed as important but as something where, in an agile project, the only decision made during project setup is that the cost estimates are repeated for each sprint (i.e., no upfront TCO estimation during project setup; alternatively, one could also say the costs can be derived from the team setup and the involved personnel costs).
- Result set #6 (Risk Management): Confirmed as important. One important point was managing the difference between standard product development and customer-specific requirements.

- Result set #7 (Trust and Security): Confirmed as important. In this context, it was suggested that safety must be added here (or in a new result set). An example given here was worker safety during the tightening process.
- Result set #8 (Reliability and Resilience): Confirmed as important; here, with a reference to OEE (overall equipment effectiveness), which again is industry-specific.
- Result set #9 (Verification and Validation): Confirmed as important, with a reference to hardware versus software perspective.
- Result set #10 (Service Operations): Confirmed as important, with a reference to the need to differentiate between the supplier service operations (support hotline and services, cloud operations) and the onsite support operations (tightening system support team at the factory).
- Overall, the list was seen as comprehensive, with one reference to safety as a missing point that could be added either to #7 or as a new result set.

For the IgniteWorx rules, one suggestion was to look at the use of machine learning. A high level of complexity of the on-asset processes (in this case, the control of torque and angle) could indicate the need for machine learning-based evaluation of process data in the backend (e.g., the mapping of tightening curves against control curves).

5.5 C: Remote Maintenance

Project C of this case study is a solution for remote maintenance, which may be one of the most well-established IoT use cases. In a patent from 1989, Yoshida and Nakamura (1989) already describe a "remote maintenance/supervisory system and method for automated teller machines." The described system is designed so that "in the event that a fault occurs in the automated teller machine, the cause of generation of the fault and a position at which the fault occurs are displayed in the remote supervisory controller." Over the course of time—and with the emergence of IoT and machine learning—the concept of remote maintenance has evolved. The following looks at basic condition monitoring, reactive maintenance, and proactive maintenance from the perspective of building operations equipment.

5.5.1 Condition Monitoring

ISO 17359:2018 provides general guidelines for condition monitoring and diagnostics of machines (ISO, 2018). The document identifies a number of parameters typically associated with performance, condition, and quality criteria of machines, including:

- vibration
- temperature
- tribology (especially friction, wear, lubrication)

- flow rates
- contamination
- power
- speed

Depending on the specific parameter, specialized IIoT sensors must be deployed, combined with data capturing and analytics. The next step is to use the data and analytics results for maintenance. Muhonen et al. (2015) describe condition-based maintenance (CBM) as a maintenance strategy "based on the actual condition of the asset": "Unlike in planned scheduled maintenance, in which the maintenance is done based on predefined scheduled intervals, in CBM the maintenance is done when it is caused by asset conditions."

5.5.2 Reactive Maintenance

Reactive maintenance is based on an approach where equipment is restored to normal operations after a breakdown. Condition monitoring can be used to inform the operator if a repair is imminently required. Also, the condition monitoring data can be used to help identifying what kind of repair is needed.

While this approach has many disadvantages (especially potentially longer equipment downtimes since the repair can only start after the failure), it still seems very common. For example, Hemmerdinger (2014) reports that 55 percent of US building operators rely on reactive maintenance approaches for their equipment in buildings.

5.5.3 Proactive Maintenance

Proactive maintenance is using condition monitoring data to help anticipate and manage equipment failures before they actually occur. To detect deteriorating equipment conditions based on parameters such as those listed above (e.g., vibration, temperature, flow rates, etc.), usually advanced computational concepts such as machine learning must be deployed (Mobley, 2002). A number of different concepts can be identified supporting proactive maintenance:

- Preventive maintenance: Regular maintenance intervals, determined by uptime statistics. According to Swanson (2001), "This type of maintenance relies on the estimated probability that the equipment will fail in the specified interval. The work undertaken may include equipment lubrication, parts replacement, cleaning and adjustment."
- Predictive maintenance: According to Swanson (2001), "Under predictive maintenance, diagnostic equipment is used to measure the physical condition of equipment such as temperature, vibration, noise, lubrication and corrosion. When one of these indicators reaches a specified level, work is undertaken to restore the

equipment to proper condition. This means that equipment is taken out of service only when direct evidence exists that deterioration has taken place."

- Prescriptive maintenance is based on predictive maintenance but proposes prescriptive maintenance measures in addition. Matyas et al. (2017) write, "Based on the analyses of historical data and incoming real time data, required maintenance measures are predicted by a system and a course of action is prescribed."

5.5.4 Project C: Remote Maintenance System Architecture

The system described in project C of this case study is mainly supporting reactive maintenance, according to the definitions above. Figure 85 provides an overview of the system architecture. In this example, the applied domain is commercial buildings. In this domain, equipment deployed in the building is functionally highly diverse and technically heterogeneous, including intrusion detectors, fire detectors, video cameras, access management, shading controllers, lighting, heating, ventilation, and air conditioning.

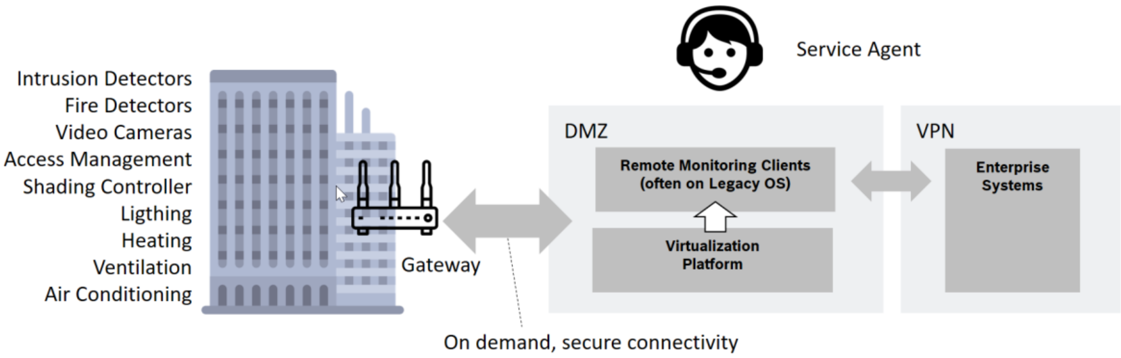

Figure 85: Remote maintenance system for commercial buildings

The remote maintenance system examined here is a generic platform for remote maintenance of devices in the field. The focus is on retrofitting of existing human workflow-centric environments, as well as the enablement of secure, virtualized on-demand device connectivity and digital workplace provisioning. This is a standard product that can be customized for different customer requirements.

The system is utilizing gateways to connect locally to these different types of equipment. The gateway is connected to a demilitarized zone (DMZ). In the DMZ, a virtualization platform is used for on-demand instantiation of remote-monitoring clients. These clients often run on legacy operating systems managed and instantiated by the virtualization platform. Service agents can then use the different remote monitoring clients for maintenance purposes.

5.5.5 Lessons Learned

One key lesson of this project was that heterogeneity is a clear driver of complexity. Given the many different types of equipment that often must be integrated by this solution, this is a key cost driver. This project has the requirement to integrate many different legacy protocols, APIs, and legacy remote-monitoring clients in a secure manner.

Security is another driving force, especially the requirement to minimize the amount of time during which the backend stays connected to equipment in the field.

The virtualization architecture needs to play hand in hand with the connectivity architecture to support these requirements.

5.5.6 Findings for IgniteWorx

Regarding the Ignite project assessment, important feedback was regarding the project environment. This relates especially to the IT organization that builds and operates the solution. The recommendation was to define a new dimension as follows:

Dimension	IT Organization
Description	Please indicate the type of IT organization that will be responsible for building and operating the IIoT solution.
Options	1) Small team, single location 2) Small team, multiple locations 3) Multi-tier IT organization, with different responsibilities for network, database, server ops, etc., in single location 4) Multi-tier IT organization, with different responsibilities for network, database, server ops, etc., in different locations

Regarding the IgniteWorx result sets, the feedback included:

- For project management methodology, also consider ad-hoc project management. This is a management style very often found in projects with stakeholders from multiple organizations. Ad-hoc project management does not follow any of the established project management methodologies; it is, as the name implies, ad hoc.
- For technology selection, one should not only focus on new technologies but also on integration of existing systems and legacy systems.
- For risk management, the organizational risks also must be considered: who is defining the requirements, who is implementing the system, who is operating it? Also, risks are different for standard products versus custom-built solutions.

Finally, regarding the IgniteWorx rules, the following was proposed:

- A high number of assets in the field is a strong indicator that OTA updates will have to be supported. The interviewee suggested that there is a “Murphy’s IoT law”: “If it is not remotely accessible, it will break and require remote updates.”
- A high number of stakeholders requires efficient stakeholder management.
- A long project duration (or project solution lifetime)
    - Increases the risk of changes to the customer organization, which makes tracing of requirements difficult.
    - Requires efficient management of technology evolution. The example given was RDBMS versus NoSQL-DBMS for telematics data.

While the last three points sound like fairly generic observations, the interviewee was specifically referring to the requirements for the integration of many different asset categories in a heterogeneous environment. Thus, the suggestions must be seen in this light.

5.6 D: OTA in Automotive

Project D of this case study is a solution for OTA (over-the-air) updates, operated by a large automotive manufacturer.

The complexity of the hardware and software deployed on modern vehicles is constantly increasing. Mahmud et al. (2005 - 2005) provide concrete examples of vehicle functions that are driving this, including drive-by-wire, telematics, entertainment, multimedia, pre-crash warning, highway guidance, and remote diagnostic. The trend toward autonomous driving and AI onboard the vehicles is only increasing this development further (Parkinson et al., 2017), and modern cars can include up to one hundred microcontrollers (Dhaneshwar, 2017).

The ability to update the software of large fleets of vehicles in the field is becoming increasingly important to manage this complexity. This requires being able to provide software updates OTA on a large scale. For example, Andrade et al. (2017) describe how to manage “Massive Firmware-Over-The-Air Updates for Connected Cars in Cellular Networks.”

5.6.1 OTA Framework

The OTA framework developed by IIC as part of the intelligent transportation systems (ITA) framework describes the main elements of an OTA architecture (IIC, 2018a). See Figure 86.

From the OTA perspective, the software development environments are authoring systems, which create the software bundles that will eventually be distributed. The authoring process is not controlled by a single organization. The fact that the OEMs themselves, and also tier 1 and tier 2 suppliers, are delivering individual components and modules is increasing the level of heterogeneity found in modern cars (Jiacheng et al., 2016).

The distribution component must support campaign management and package installation, including bundle preparation, filtering, rules-based scheduling, and execution, as well as tracking and reporting (IIC, 2018a).For on-vehicle deployment, there are different options, as described in the next section.

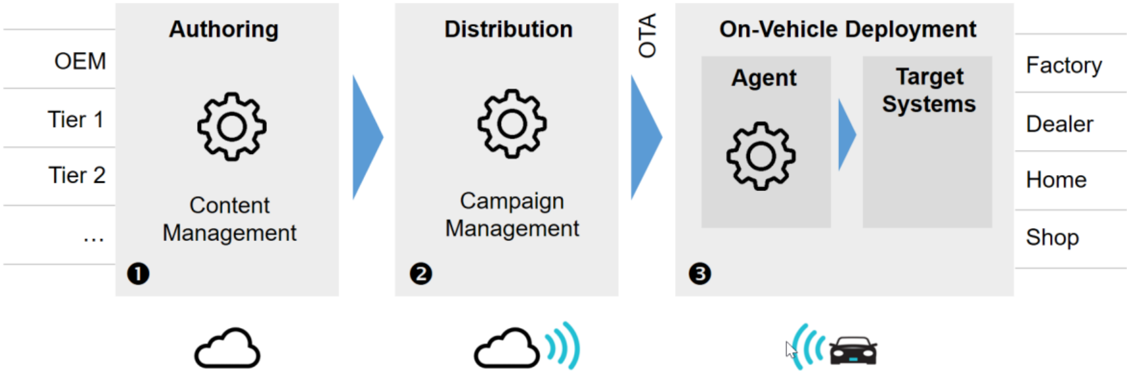

Figure 86: OTA in automotive (IIC, 2018a)

5.6.2 On-Vehicle Deployment

One common option for on-vehicle deployment is the use of an agent, which receives software updates from the backend (e.g., via WiFi or GSM) and then controls the local distribution process, as shown in Figure 87 (IIC, 2018a).

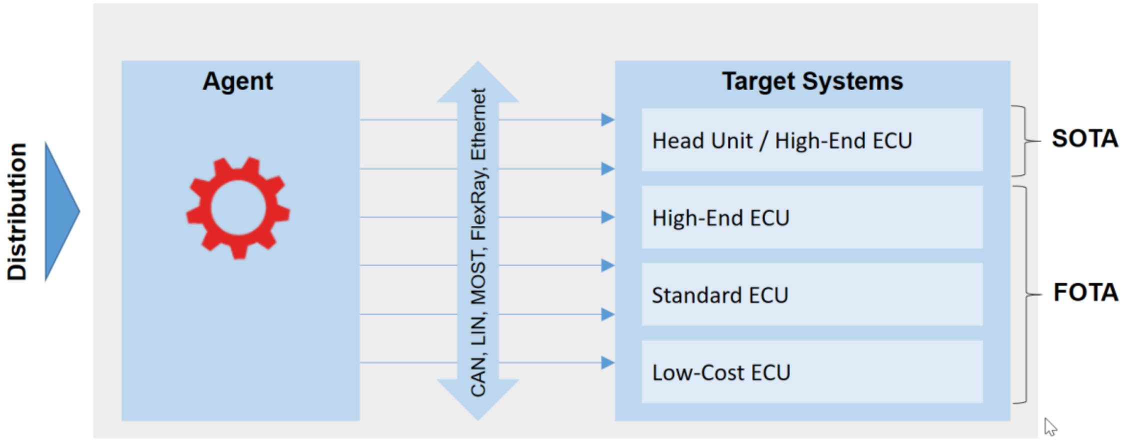

Figure 87: OTA on-vehicle architecture (IIC, 2018a)

The agent must distribute the software bundles via a multitude of different bus systems, including CAN, LIN, and FlexRay (Dhaneshwar, 2017). Target systems can be different types of electronic control units (ECUs), depending on the functions they must support.

5.6.3 OTA and Security

Given the mission criticality of the use case, it seems clear that security is a key element of OTA. Any OTA security framework will have to address all layers described in Figure 86, from the backend to the vehicle. Mahmud et al. (2005 - 2005) propose a solution that combines virtual private network (VPN), secure socket layer (SSL). and symmetric public key encryption. Idrees et al. (2011) are proposing an onboard security architecture that combines hardware and software modules. Nilsson et al. (2008) present a set of guidelines for creating required, secure infrastructure for wireless diagnostics and software updates in modern vehicles.

5.6.4 Lessons Learned

The expert interview has confirmed the complexity of OTA, especially for very large fleet of vehicles. Especially at this scale, a sophisticated distribution component is required. One example that was given is a distribution rule that can be used to avoid updates to cars that are too far away from a service station (in case a problem comes up).

Another key issue here is the integration of the onboard agent with a multitude of highly heterogeneous ECUs, potentially via a number of different heterogeneous bus systems.

Finally, the compliance with functional safety regulations like ASIL in the context of OTA is still not finally resolved since OTA is not built into concepts like ASIL (confirmed by Birch et al. (2013)).

5.6.5 Findings for IgniteWorx

For the Ignite-based project assessment, the proposal was made to actually differentiate the Ignite dimensions based on the specific industry or use case. This would basically mean that specific dimensions would only be used for projects in specific industries or with specific use cases. The argument brought forward was that a smart home use case might benefit from different questions than an industrial asset management use case.

Another input on the Ignite dimensions was the lack of a "global scale" dimension since that is an important factor.

Specifically for the project management result set, it was proposed to add an "ad-hoc" project management style. The argument was that this still seems to be the reality in many situations.

Another suggestion was to add a "make versus buy" result set to cover an essential question at the beginning of the project lifecycle.

For the rules, three example were discussed:

- The scope of operations (local, global) should be a strong indicator for need of

    - Custom hardware for local requirements
    - Global communications network
- Deployment strategies (Delta updates, update group configurations) could be linked to
    - Number of assets
    - Size of SW update
    - Network: LAN in the repair shop, WiFi at home, 4G/5G
- If a contend distribution network (CDN) is required could depend on:
    - Global scale
    - Heterogeneity of on-asset software
    - Frequency of updates

While this interview was not looking at all aspects of IgniteWorx and instead focusing in depth on some of the specific points outlined in the above, the interviewee confirmed the feasibility of the IgniteWorx approach in principle.

5.7 Cross-Study Findings

In the following, an overview of the cross-study findings is provided. This includes findings on the Ignite dimensions, IgniteWorx result sets, and rules. Finally, some general conclusions are drawn.

5.7.1 Findings on Ignite Dimensions

Three main lessons can be taken from the case study regarding the Ignite dimensions. First, the overall findings are discussed. Second, the lessons learned regarding the concept of an asset in Ignite in general. Third, the different project perspectives and how to better handle them.

5.7.1.1 Overall Findings

Each of the projects from the case study looked at the Ignite dimensions with a different perspective. A number of interesting suggestions were made for extending the current Ignite dimensions. A visual overview is provided in Figure 88.

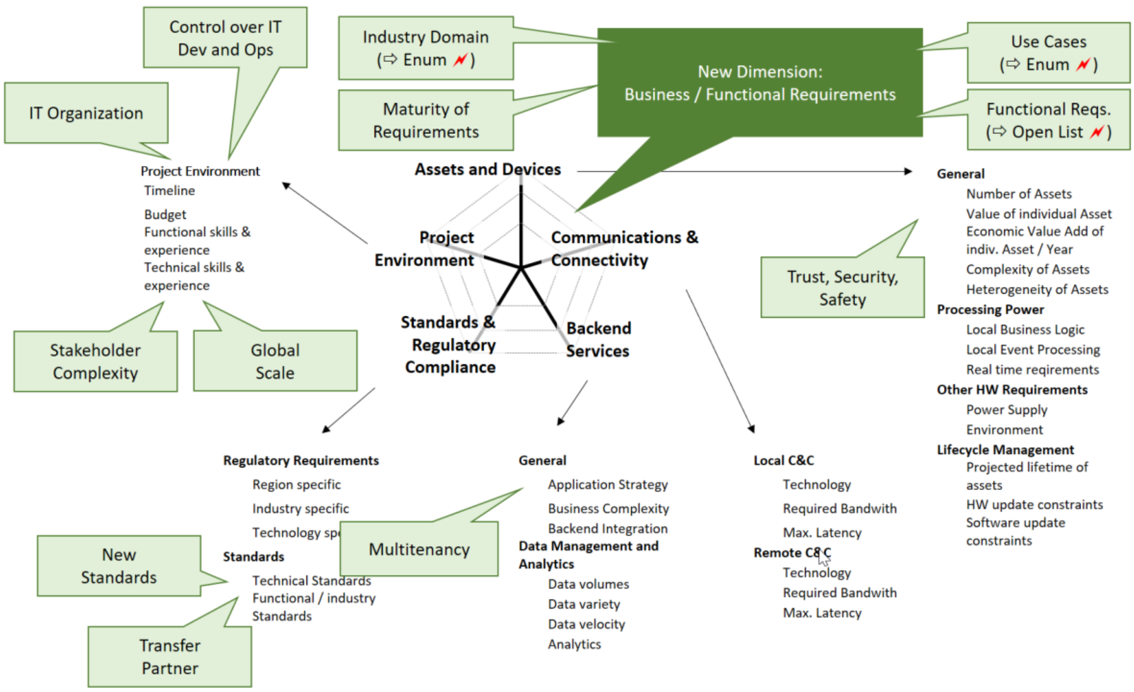

Figure 88: Overall findings on ignite dimensions

Maybe the most fundamental insight is that the current Ignite dimensions are lacking a dedicated dimension to cover the business perspective, including the supported use cases and functional requirements. Based on the learnings from the case study, this new dimension, Business Dimension/Functional Requirements, would require the following:

- Industry Domain: This would most likely be an enumeration or a selection list, meaning that one would have to break with the "four options per dimension" principle. This would have a significant impact on any IgniteWorx implementation since this would require an extension to the data model and UI implementation. Also, the proposal for IgniteWorx rules that create mappings from Ignite dimensions to results in result sets would have to be reconsidered.
- Use Cases: Very similar to industry domain, at least in the ideal case, in which this could be mapped to a predefined selection list of use cases. However, it is unclear if a complete list of use cases that would fit all possible user projects could be compiled. Thus, this might even have to allow the provision of additional use cases as text input, which would make processing (e.g., via rules) much harder.
- Functional Requirements: This would require an open list, i.e., a user would be able to specify 1-n functional requirements as string values. This would be valuable input but also much more difficult to process for IgniteWorx rules compared to the four options approach or selection lists with predefined options.

While technically not easy to handle with the initial design of IgniteWorx, this should be considered for the next major release.

5.7.1.2 Asset as a Key Entity

Another key lesson learned from the interviews was that sometimes it is not immediately obvious what the asset is that the Ignite dimensions are referring to. One core idea for Ignite is that it should focus on assets instead of lower-level entities such as sensors and devices. The reason is that assets are typically what the business is focusing on; the asset is what you would find in the balance sheet of a company. The concept of an asset in Ignite allows one to look at dimensions like number of assets in the field, economic value add of assets, etc.

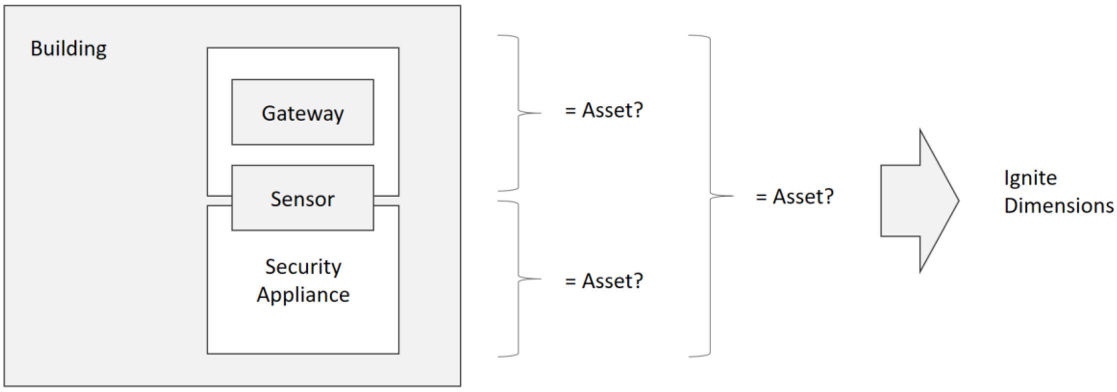

Figure 89: Different possible scopes of an asset in Ignite

However, during the interviews, it became clear that there often seems to be a fair share of ambiguity here. As is shown in Figure 89, an example is project C, which provides maintenance to domains like building facility management and security. In this example, it is not immediately clear what the asset is: the building, the legacy security appliance in the building, or the gateway that is deployed as part of the solution.

To overcome this ambiguity in Ignite, ISO 55000:2014 could be used (ISO, 2014). This standard provides principles and a clearly defined terminology for asset management.

5.7.1.3 Project Perspective: IIoT Solution Supplier versus Customer

Finally, another issue with Ignite that became visible through the cross-case analysis was a question regarding the project perspective. For commercial IIoT solutions that are typically sold multiple times, the question is which part of the project IgniteWorx is addressing: the perspective of the team that is building the solution or that of the customer implementing the solution in his or her company. This dilemma is exemplified in Figure 90.

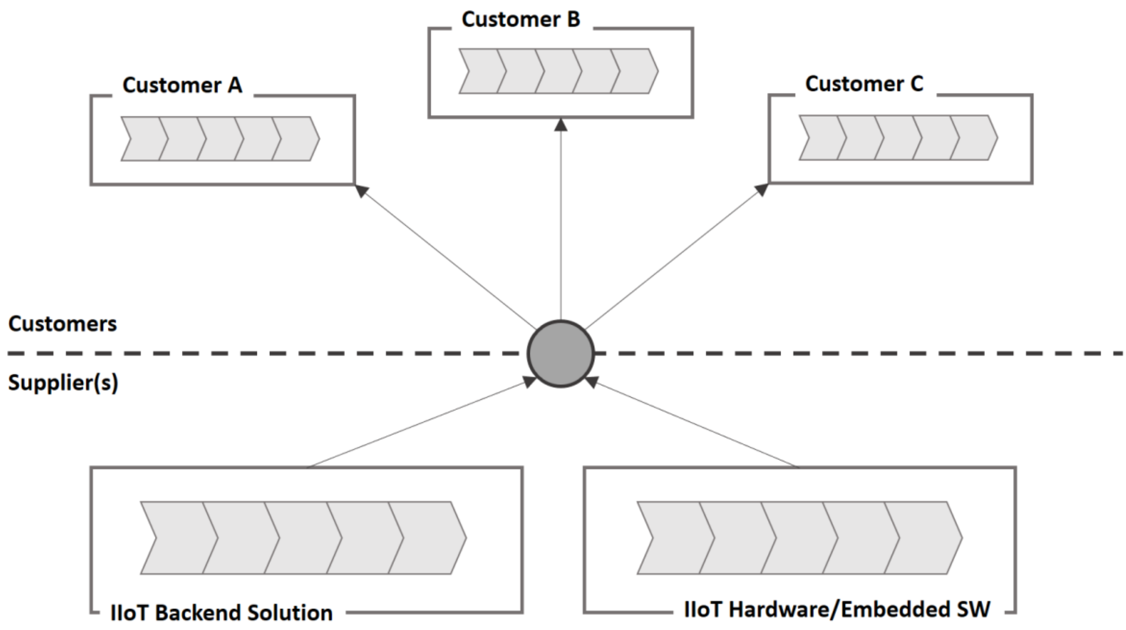

Figure 90: Project structure for IIoT COTS solutions

It seems there are at least three options for the perspective that the user of Ignite could take on during an IIoT project assessment:

1. Development of IIoT COTS Solution (Solution Provider): In this perspective, it would be difficult to answer any question specifically regarding topics like assets (e.g., number of assets in the field) since the solution provider is not him- or herself deploying assets in the field. This gets further complicated in case the solution is provided by different suppliers for the IIoT backend solution and the IIoT hardware/embedded software, for example, the suppliers of the tightening quality assurance solution in project B, including the actual tightening system as well as the QAS solution
2. Rollout of IIoT COTS Solution (Customer): In this perspective, some details regarding the software development side in Ignite may be seen as irrelevant since in COTS projects no new software is usually developed
3. Development of Custom IIoT Solution (Bespoke Solution Development): This perspective would actually combine perspectives 1 and 2, so this would benefit most from the current structure of Ignite

Especially for solution providers (1), it seems to make sense to combine the provider perspective with a realistic example scenario for 2. For example, the provider could assume a typical customer deployment and use this to go through the Ignite dimensions. This is what was actually done for the assessment of projects B and C in this case.

5.7.2 Findings on IgniteWorx Result Sets

Overall, the feedback on the IgniteWorx result sets concept was supportive, indicating that it provides a relatively comprehensive perspective on the different aspects critical to the setup of an IIoT project. The table below provides an overview of the cross-case analysis results, as well as a short discussion of the match with the existing definitions.

#	Result Set	Expectations (Cross-case analysis)	Match with existing definitions
1	Project Management Methodology	Guidance on the most suitable methodology	Good; could add "ad hoc" as another project management category; also, could add recommendations for suitable milestones and decision points
2	Solution Design	Strong dependency on business model was identified	Good
3	Technology Selection	Recommendations for suitable technologies; make versus buy; must also include integration of legacy technologies	Dependencies on solution design must be considered; should also consider perspective of potential productization partner
4	Resource Acquisition	Project-specific skills and resource availability	Good
5	Cost Estimation	Was not a strong requirement in any of the projects interviewed	Focus more on recommendations for the process (e.g., cost estimation in agile) than on IIoT-specific estimation techniques
6	Risk Management	Organizational risks; standard product development versus customer-specific requirements	Good
7	Trust and Security	Can be assumed as essential	Good but must add safety

8	Reliability and Resilience	The question is, how much can really already be built into the project setup?	Good
9	Verification and Validation	This depends heavily on the industry for which the IIoT solution is deployed.	Improved support for this would require extending IgniteWorx to include an analysis of the target industries (see discussion in previous section on adding support for this to Ignite dimensions).
10	Service Operations	Must include both the IIoT solution provider perspective and the solution operator perspective	Good

Table 29: IgniteWorx cross-case findings

The most critical finding here is that some of the recommendations that the project managers would like to see depend on knowledge of the industry for which the solution is designed, the supported use cases, or other functional requirements. As discussed in section 5.7.1.1, this will require extending the Ignite dimensions to support open input lists as opposed to the standardized four-options answers.

5.7.3 Findings on IgniteWorx Rules Concept

Overall, the idea of using rules to create a mapping between Ignite dimension and the recommendations from the result sets got a positive reception. Also, some interesting candidates for rules were mentioned during the interviews. However, it also became clear that it is unrealistic to perform a comprehensive capturing of rule candidates as part of these interviews for two reasons:

- Time constraints: The number of rules that are relevant for any given project is potentially very large, including up to two hundred rule definitions.[1] Given the time constraints of the interviews performed for this thesis (usually 90 to 120 minutes), it was not possible to go beyond a couple of examples that were most relevant for the interviewee.
- Manual process: The interviews were performed using a Word document template, which was used to capture the interview data in real time (see section 5.2.6.2). Using this format, it would have been very difficult to actually review

[1] If 5 rules in average for each result are assumed, and there are 10 result sets with 4 results on average, this could be more than 5x10x4=200 rules for an IgniteWorx system.

existing rules definitions and maintain a consistent view of all potential new rule candidates.

Because rules play an important role in IgniteWorx, and because of the difficulty to efficiently capture input and feedback on rules using a manual interview process, it was decided to actually focus the next iteration of the IgniteWorx design done as part of this thesis on providing more efficient tools for capturing user feedback, especially on rules.

5.7.4 Summary and Conclusions

Overall, the first evaluation of the IgniteWorx design artifact created as part of this thesis finished with a positive result. While the evaluation approach taken here is by no means a formal validation, the feedback from the interviews and the cross-case analysis confirmed that the approach taken is valid and that the first iteration of the design artifact is already relatively comprehensive. The feedback and suggestions for future enhancements can be grouped into three main categories:

1. Content improvements: There are numerous suggestions to improve IgniteWorx content, which can be done without actually changing the underlying system design, e.g., adding “safety” to result set #7 (Trust and Security).
2. New Ignite dimension “business requirements” with open input lists: Adding a new business requirements dimension to Ignite will require a system design review since this would require open lists, e.g., to capture functional requirements. This is not a data-capturing issue. The main issue here is how to include such semi-structured input into the existing rules mechanism for recommendations.
3. Improved support for capturing user feedback, especially on rules definitions: As became clear during the cross-case analysis of the input on IgniteWorx Rules, tool support for working with subject matter experts on rules definitions could potentially make the process much more efficient.

The third and last category is the most important one from the point of view of creating a working IgniteWorx system. Categories 1 and 2 describe areas for improvements, while the availability of semantically meaningful rules for recommendations—validated by subject matter experts—is a basic prerequisite of a comprehensive version of IgniteWorx. Consequently, the second iteration of IgniteWorx design presented in this thesis makes a proposal for how to best address this issue.

6 Iteration II: Continuous Content Quality Improvement

Based on the learnings from the cross-case analysis presented in the previous section, as well as first experience with a real-world implementation of IgniteWorx (the IIC Project Explorer, as described in section 4), the second iteration of the IgniteWorx design process looks at the best way of continuously improving the quality of the results presented to the user, in particular the recommendations for the custom IIoT project setup.

6.1 Improving the Quality of Results

The main objective of the second design iteration is to improve the quality of the results presented to the user, i.e., recommendations for the project setup. To enable this, IgniteWorx must efficiently capture user feedback on the recommendations as well as the underlying rules used to create the recommendations (this was a key learning point from the cross-case analysis).

However, recommendations and the rules to create them are not the only drivers for the quality of the system. Another key factor is the quality of the assessment itself, based on the Ignite dimensions. For this, it should be possible to capture both direct feedback on the definitions of the Ignite dimensions as well as the user performance when going through the project assessment based on the Ignite dimensions.

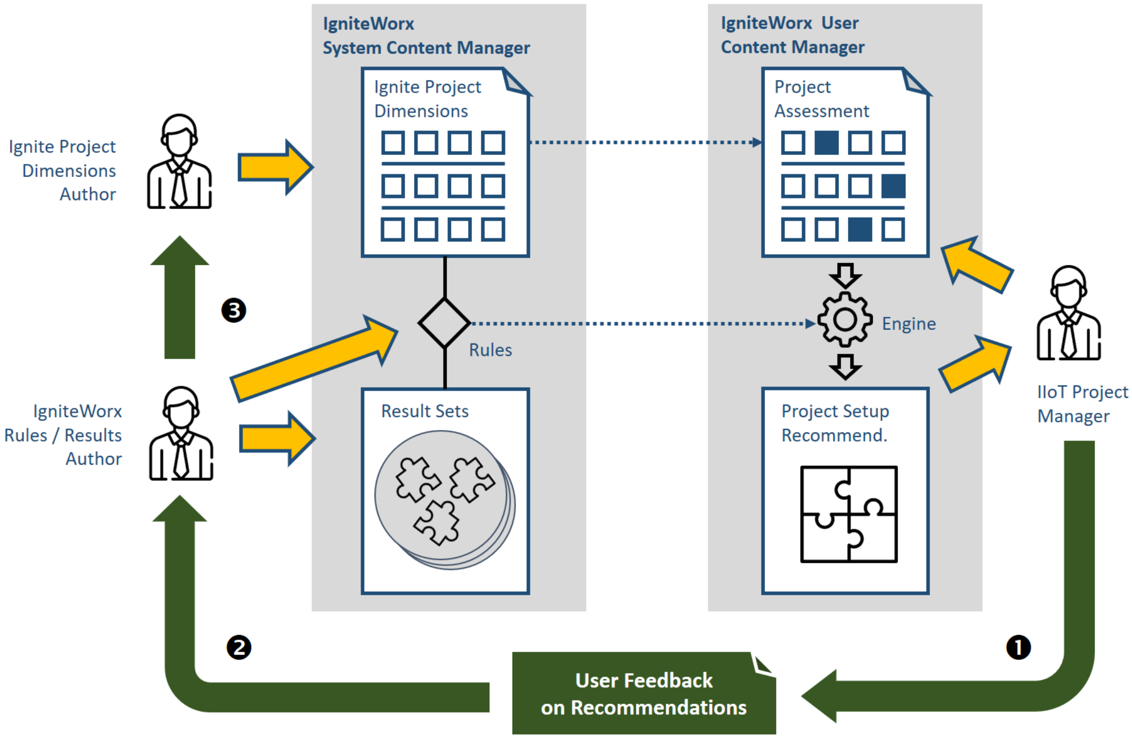

Figure 91: Incorporating user feedback on recommendations

Figure 91 summarizes how the feedback cycle is envisioned: First, a user in the role of the IIoT project manager will perform a project assessment and review the recommendations he or she is getting (1). He or she will then provide detailed feedback on these recommendations, including feedback on the reasons for the recommendations provided by the system. His or her feedback will then be used by the IgniteWorx rules/results author to fine-tune the rule mappings between project dimensions and results (2). In some cases, the results themselves will have to be adapted, depending on the type of feedback received. Furthermore, the end-user input can also include feedback on the project dimensions, which can be picked up by the Ignite project dimensions author (3).

Improving the IgniteWorx content will be a continuous process. Especially in the first phase of the anticipated system evolution (see section 2.7), this will be a manual process. This means that efficient tool support for capturing user feedback and improving content will be important.

6.2 Capturing Project Background

When capturing user feedback, as discussed in the previous section, it will be very important to understand the experience level of the user who is providing the feedback. To better understand this, the feedback system should ask about the user's background, as follows:

1. Real-world project, e.g., in the initiation or implementation phase
2. Real-world project, e.g., in field tests or in production
3. Pilots, testbeds, experience in general
4. Others

A user in category 1 would fit the profile of the main IgniteWorx target user (given that IgniteWorx is designed to support the project setup). The issue with feedback from this perspective is that it is actually not based on real project experience. This is why feedback from users in category 2 is potentially even more valuable (which is why for the projects in the case study in section 5, experts in this category were mainly selected). User category 3 would still be valuable but possibly less so than 2.

7 Design and Development II

The following discussion of the second design and development iteration is based on the design models from the first iteration, presented in section 3.3.5. IgniteWorx is following a model-driven architecture (MDA) approach, which means that the models are an important part of the system design. In the following, the original models are extended by using callouts that describe the additional data that needs to be captured. There are two icons used: cogwheels to indicate this this information should be captured automatically and a person to indicate that this information must be captured manually.

7.1 Capturing Feedback on Ignite Artifacts

The first set of feedback that must be captured by the system is related to the Ignite artifacts: categories, dimensions, and options (as shown in Figure 92). This is based on the design from iteration I in section 3.3.5.2.

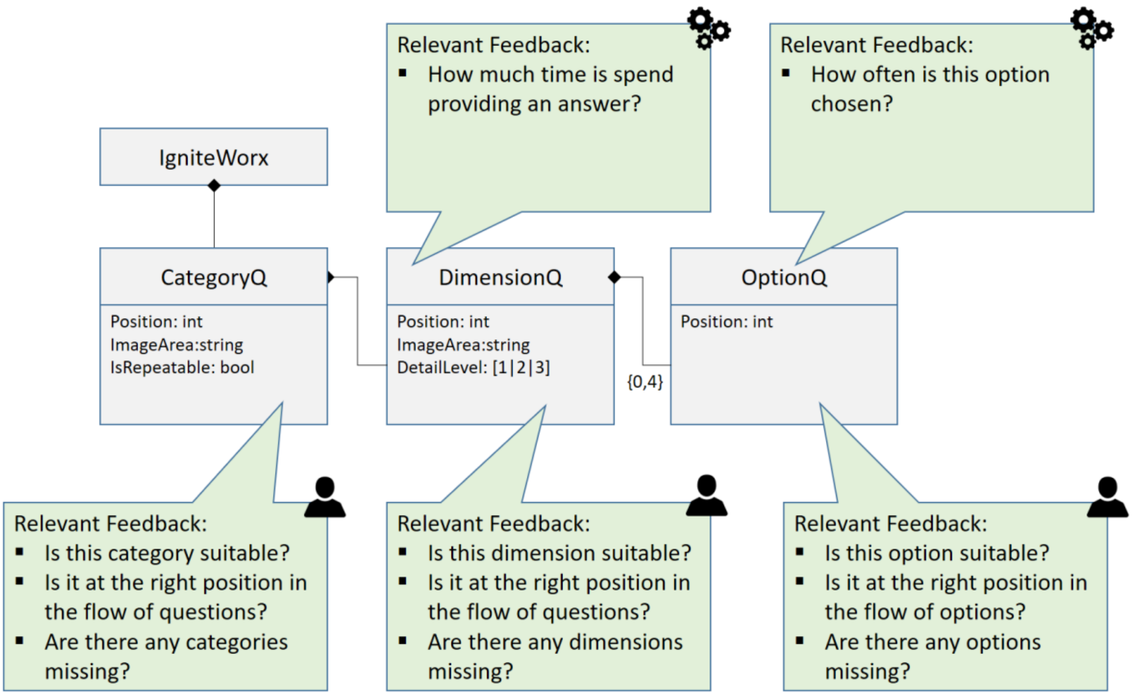

Figure 92: Capturing feedback on Ignite artifacts

For categories, this feedback includes information on the general suitability of the category, as well as the position and input regarding potential new categories. Very similar feedback is required regarding Ignite dimensions and options.

In addition, for dimensions, it would be interesting to automatically capture how much time is spent, on average, answering them. For the options, it would be interesting to see how often an option is chosen.

7.2 Capturing Feedback on Results Sets, Rules, and Recommendations

The second set of feedback that must be captured is related to result sets, rules, and recommendations, as shown in Figure 93. This is based on the design from iteration I in section 3.3.5.5.

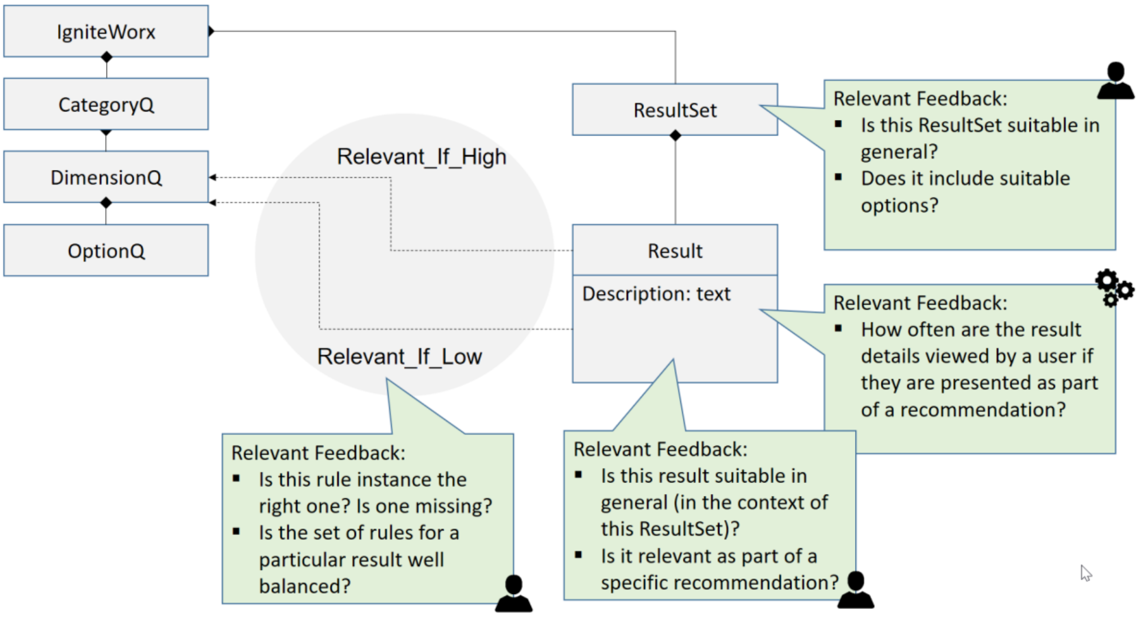

Figure 93: Capturing feedback on result sets, rules, and recommendations

User feedback is needed on whether the result set is suitable in general and in how well it includes suitable options. For the individual results, it would be helpful to automatically capture how often they are viewed in detail by the users.

Regarding the rules, feedback on two main areas must be captured:

- Feedback on individual rules instances: Are they the right rules? Which rules might be missing?
- Feedback on the composition of the rule sets in general: Are they well balanced between the different results and result sets?

8 Demonstration II

The demonstration is based on the IIC Project Explorer version 1.1, which implements the design proposals for IgniteWorx from iteration II of this thesis (the IIC Project Explorer version 1.0 was introduced in section 4).

8.1 Project Background

As per the enhancement objectives described in section 6.2, the feedback mechanism should ask the feedback provider about the background of his or her feedback. This is shown in Figure 94.

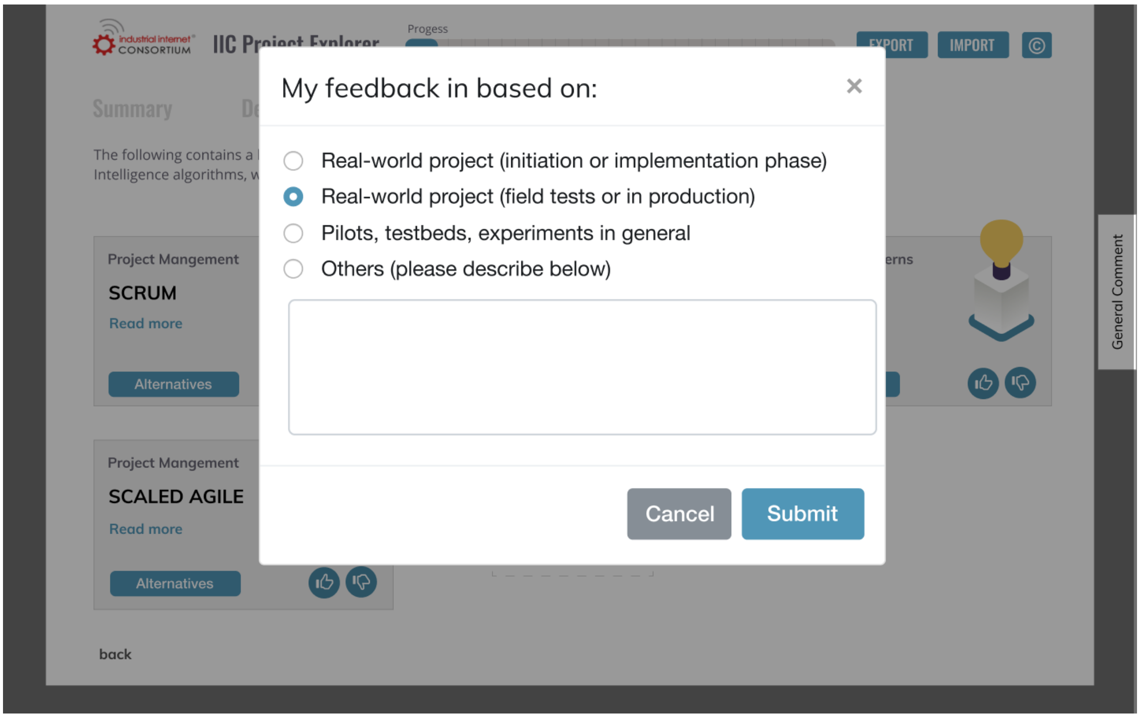

Figure 94: Capturing project background data

8.2 Feedback on the Project Survey

The feedback on Ignite artifacts in the IIC Project Explorer 1.1 is relatively straightforward. The feedback button on the right side of the screens for the different Ignite dimensions opens a modal that allows the user to provide feedback on the active dimension (i.e., the current question in the project survey). This is shown in Figure 95.

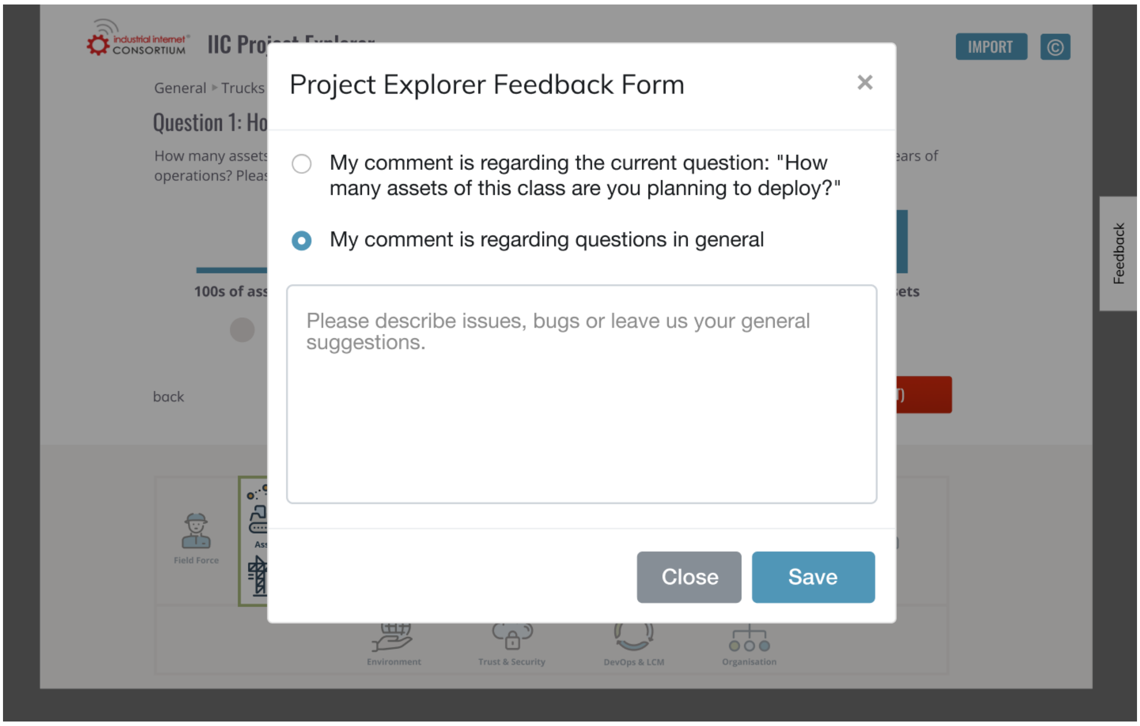

Figure 95: Feedback on the project survey

8.3 Feedback on Recommendations

The results page ("Related Readings") in the IIC Project Explorer 1.1 has been extended to allow for up- or down-voting of each recommendation provided on the page. Furthermore, feedback regarding proposals for new categories can be added as well, as seen in Figure 96.

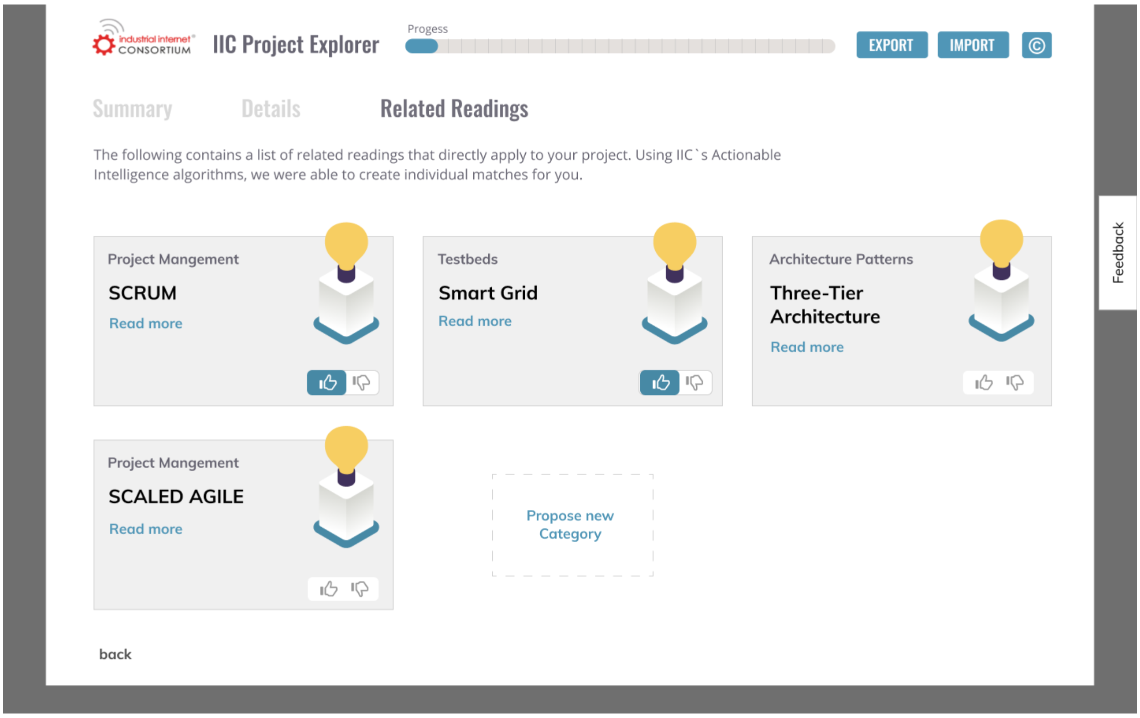

Figure 96: Feedback on recommendations

8.4 Feedback on Rules

Finally, the feedback on individual rules is captured on the recommendations details page of the IIC Project Explorer. As can be seen in Figure 97, under the "Why?" heading of the page, the selections from the project assessment that have led to this particular recommendation are shown. The enhanced IIC Project Explorer 1.1 now allows up- or down-voting of each element on this list. Internally, these votes are translated to votes on the actual rules, which create the link between the assessment choices and the recommendation.

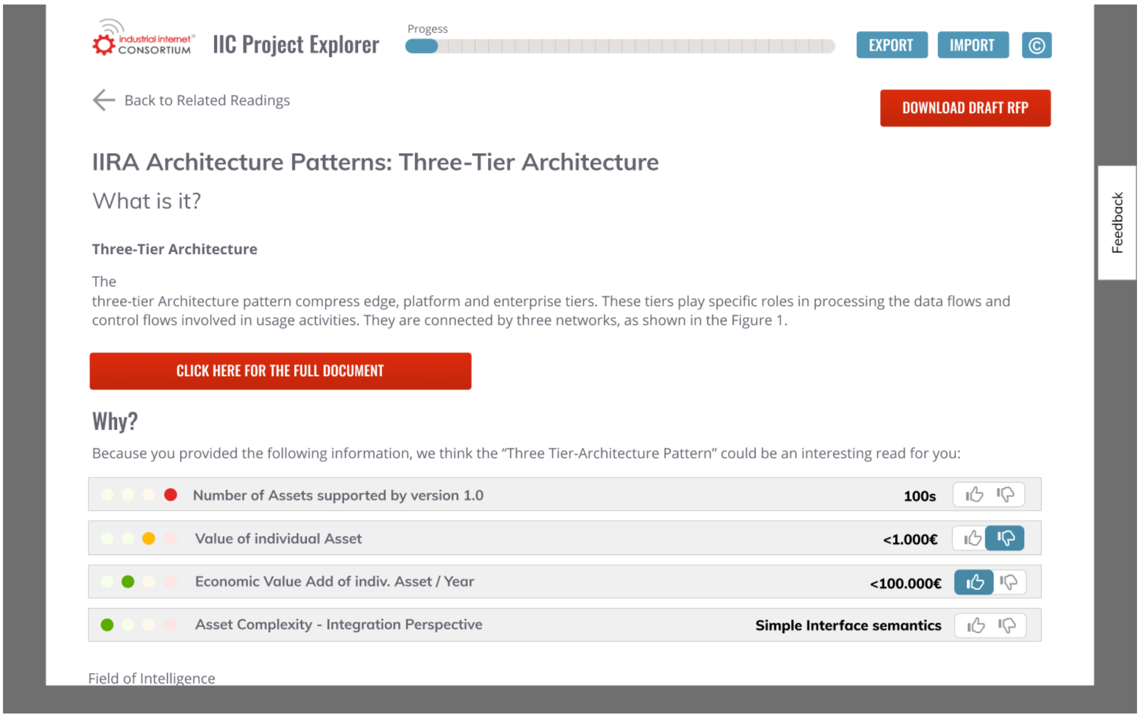

Figure 97: Feedback on rules

9 Evaluation II

Evaluation II must provide answers to the following questions:

- Is the general usability of the new UI features for capturing end-user feedback during and after the IoT project assessment acceptable?
- Can the end-user feedback be used to efficiently to improve the quality of IgniteWorx content, as described in section 6.1?
- Can the results of iteration II generally support a continuous content improvement process for an IgniteWorx instance?

These questions are partially of a qualitative nature, and partially they must provide answers regarding the actual functionality of the system. Consequently, evaluation II uses a combination of user experience evaluation and functional testing.

9.1 User Experience Evaluation

User experience (UX) is a maturing research area that includes various forms of user feedback capturing and evaluation. For example, Law and Abrahão (2014) discuss how evaluation feedback shapes software development. Much of this is still happening on the qualitative level, although Law et al. (2014) also look at quantitative measurements for UX measurement.

Web sources provide many different recommendations for UX evaluations. For example, UX for the Masses (2010) offers concrete usability testing guidelines, which include the following recommendations:

- Ensure the recruitment process is yielding participants that match the required end-user profile.
- Do a planning session where a dry run of the usability tests is performed.
- Write a test script that summarizes the key goals and tasks for the tests.
- Ensure the setup of the test room is similar to the expected environment in the field.
- Do a pre-session briefing that summarizes the expectations on the test user.
- During the session
    - Ensure that issues like filling out consent forms and answering any questions are taken care of.
    - Consider using a warm-up task to help build a rapport.
    - Verbally talk the test user through the tasks at hand and keep a written summary at hand.
    - For each task, prepare a set of questions that should be answered.
    - Take notes using pen and paper (using a computer could be distracting).
    - Use a questionnaire to gather additional feedback during the wrap-up.

- After the session
    - Analyzing the results based on the notes taken during the tests should be done as quickly as possible so that the experience is still fresh in the mind.
    - Create a concise report, highlighting the key findings.

9.2 Functional Testing

Functional testing is a well-established discipline that is distinct from user experience testing. Functional testing is designed to help verify that each function of an application actually works in conformance with the corresponding requirement specification. This is different from UX or usability testing, which is more focused on customer acceptance.

As a subset of system testing, functional testing (or functional system testing) can be viewed at as a combination of UI testing and application logic testing. Alkhalid and Labiche (2016) provide the following definition: "We define Functional System Testing as checking conformance of the entire GUI against its functional requirements, either by directly interacting with the application logic in isolating and focusing only on the UI, by focusing on the UI in combination with the application logic, or a combination of those." Functional testing can either be manual or automated. A typical goal is to maximize test coverage, e.g., by creating comprehensive test suites that address as many combinations as possible of user input and system output. Alkhalid (2018) looks at ways to bridge the gap between GUI functional system testing and functional system logic testing by using static analysis as a tool for software verification that helps to analyze the relationships between inputs and executed parts of software code.

For the purpose of this test, the following assumptions are made:

- A manual test approach is chosen because automating functional tests requires a significant investment.
- The focus is on testing the expected functionality on the UI level because performing code-level test would be difficult with the target test users (especially the domain experts).
- A minimum level of repeatability of the tests must be ensured through well-documented test scripts, as well as regular snapshots of the before and after test data.

Given these assumptions, the following section describes the resulting test design.

9.3 Test Design

The proposed test setup is based on a combination of UX evaluation and functional test, as outlined in sections 9.1 and 9.2, respectively. The following sections describe the test setup, including goals and supporting tasks, test context, test user selection, and test environment.

9.3.1 Test Goals and Supporting Tasks

The overall test goals are based on the goals for evaluation II, as outlined in section 9. To achieve this, the following is proposed:

- Perform a test that iterates through the complete IgniteWorx content optimization cycle, including
  - IoT project assessment by end-user
  - Capturing of end-user feedback on recommendations provided
  - Analysis of recommendations by IgniteWorx content authors
  - Update of IgniteWorx content based on end-user feedback, especially IgniteWorx rules to optimize quality of recommendations
  - Reenter evaluation loop by going back to end-user, and reevaluate recommendations based on same test project with new recommendation rules
  - Review end-user feedback based on optimized recommendations
  - Perform final test interview with all test stakeholders to evaluate quality and feasibility of the new content optimization process
- Make the test as close to a real-world use of the system as possible. To achieve this, select a concrete IoT project to serve as the foundation for the test. Select test users as close as possible to the project domain.
- Use the existing IIC system for the test (with permission of the IIC Steering Committee, see section 1.9). Start with the content available in the IIC system at the time of testing. Note that this content includes IIC project dimensions, which are based on Ignite 1.0 but modified by IIC to fit the needs of IIC. The IIC system also includes results sets that are IIC-specific and partly different from the result sets proposed in this thesis. See Table 26, which provides a detailed comparison of the result sets from this thesis and the result sets in the IIC system at the time of testing.
- A key assumption is that the end-user feedback on the initial recommendations will result in proposals for optimizing the initial IIC content (dimensions, results, and rules). A clear goal of this test is to enhance the existing content to reflect these proposals. For the purpose of this test, this can be done in a sandbox test environment, which has no direct impact on the IIC production system. The results will then also be made available to IIC for consideration.
- The findings from evaluation I of this thesis can also be taken into consideration for the next version of the content. For example, some new Ignite project

dimensions can be considered, which are prosed for Ignite 1.1 and summarized in section 12.1.
- Entering the evaluation loop for a second time should then yield direct feedback from the end-user on the quality of the improved content. This concludes the test.

9.3.2 Test Content: IIoT Project and Results Set

To make the test as realistic as possible, a concrete project should be used as the underlying foundation. This project should be a real-world IoT application with a strong industrial focus.

Also, a key goal of the test is to evaluate the quality of the recommendations made by the system after the initial project assessment. Consequently, one area for recommendations should be selected as the focus area for analysis during the test; this means that the test should focus on one particular result set.

Based on the above assumptions, the following has been identified as a suitable combination for this test:

- Test Project: As the underlying project, the tightening quality assurance system from section 5.4 has been selected. This is one of the four projects analyzed as part of the case study from evaluation I. The benefit is that this project shares many typical characteristics of an IoT solution. Also, access to domain experts for the test is ensured.
- Test Result Set: As the result set in focus, project management methodology (PMM) was chosen. The reason is that this was already used as the example for the design considerations for IgniteWorx rules, as described in section 3.1.3. This will allow a later comparison of initial design considerations versus test results.

9.3.3 Test User Selection and Test Environment

For test users, the following roles were identified and allocated:

- End User: Test user with expert knowhow of the test project (tightening quality assurance system) and its domain (torque analysis)
- IgniteWorx Results Sets and Rules Author: Expert on the IIC-specific implementation of IgniteWorx, including results sets and the mapping of the results to IIC content based on rules.
- Ignite Dimensions Author: Expert on the IIC-specific version of the Ignite project dimensions.

As the test environment, a normal office setting was chosen because this is a typical environment in which end-users or content experts would use the system.

9.4 Test Execution

The following documents the test execution and evaluation. Test execution includes IoT project assessment, recommendation details, end-user feedback, content author dashboards to review the feedback, incorporation of required changes, and reentering the test loop from the end-user perspective.

9.4.1 IoT Project Assessment

As discussed in section 9.3.2, the tightening quality assurance system (TQA) from section 5.4 was selected as the test project. Figure 98 shows the summary of the high-level assessment results for the test project data, which has been put into the system by the test-user with the role of domain expert, as defined in 9.3.3.

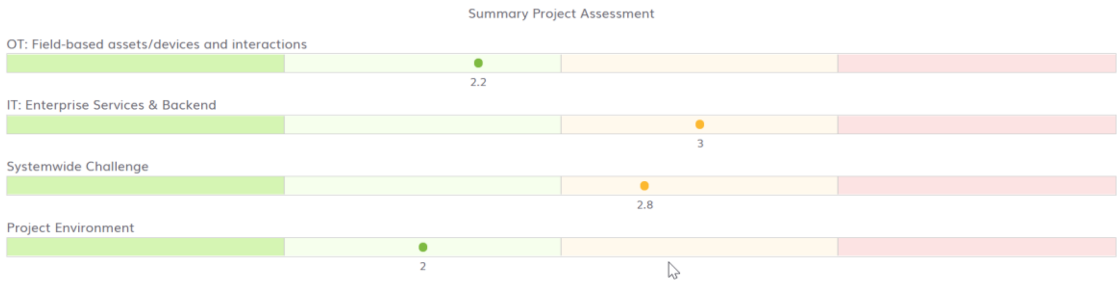

Figure 98: High-level assessment results for test project

As can be seen, the OT side has a complexity score of 2.2 for the TQA, which is relatively moderate. Looking at the details, this is based on some assumptions like moderate tool fleet sizes, use of standard technologies, etc. The IT side has a higher complexity score of 3, which is due to factors like use of machine learning for the analysis of torque data, as well as creation of tightening profiles for predictive analytics. The systemwide requirements are slightly above average (2.8), which is due to intra-factory networking requirements. The project environment is again moderate (2).

9.4.2 Recommendation Details

The IIC system provides recommendations from five result sets, as can be seen in Figure 99. See section 4.8 for a comparison of IIC result sets and proposed results sets from this thesis.

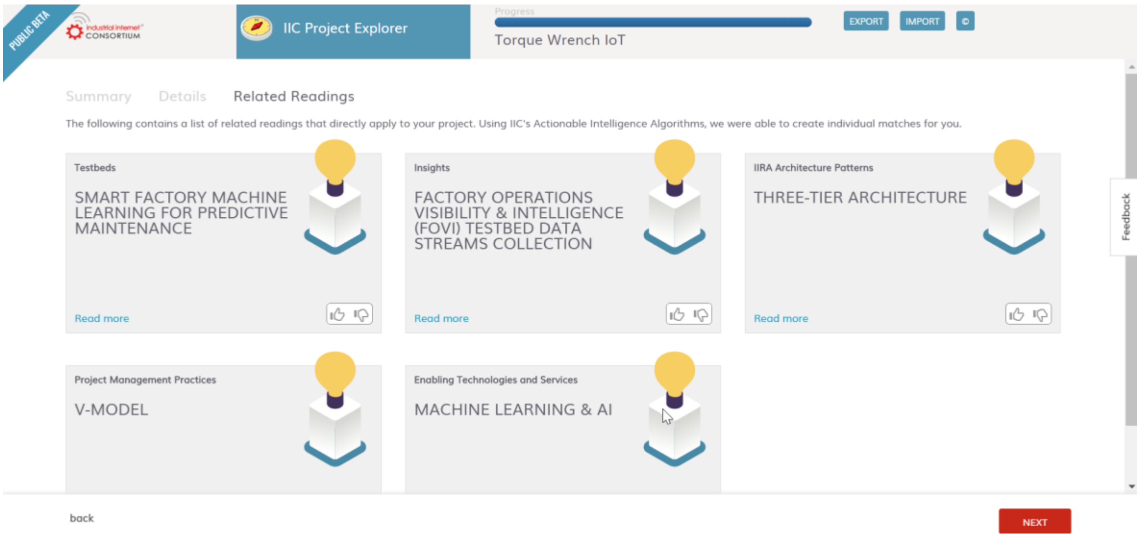

Figure 99: Recommendation details for test project

As can be seen from the recommendations in Figure 99, some of them seem to make sense. For example, the IIC system recommends looking at the smart factory machine learning testbed as a reference, which seems logical for our test project, which also has ML as an explicit requirement. Similarly, the recommendations for IIC insights and enabling technologies seem suitable. However, for project management practices and potentially also architecture patterns, a deeper look seems required regarding the quality of the recommendations. In the following, the focus is on project management practices, as mentioned in section 9.3.2.

9.4.3 End-User Feedback on Recommendations

Figure 100 shows the details of the recommendations for project management practices made by the IIC system for the test project.

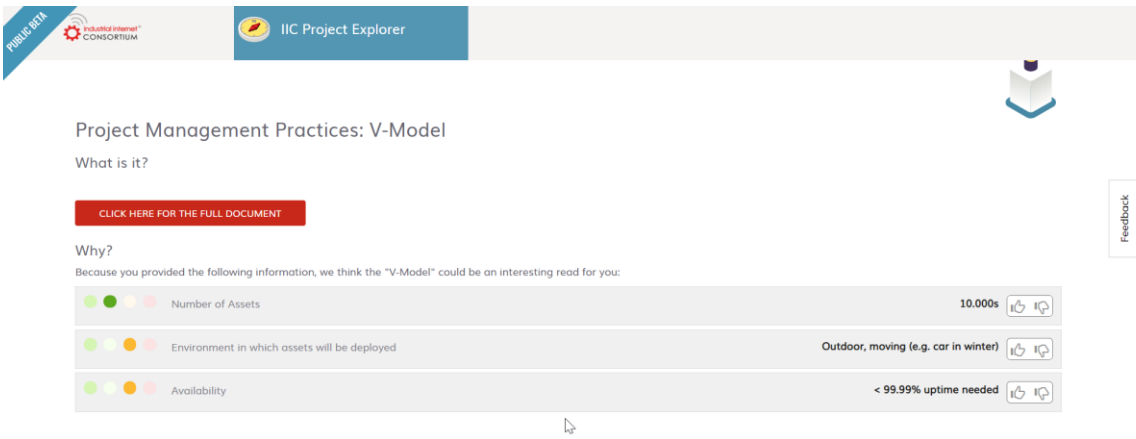

Figure 100: Details of the recommendation for project management practices

As can be seen, there are three dimensions used as the foundation for the recommendation by the IIC system:

- The number of assets deployed in the field (relatively low in this case)
- The complexity of the work environment in the field (high)
- Availability requirements (high)

Given the discussion on the PMM result set in section 3.2.1.3 (“Result Candidates and Matching Considerations”), it seems questionable that the number of assets is actually a strong indicator for the use of the V-Model as the PMM for the test project. The same seems to be true for high complexity of the work environment. Only the high availability requirements seem to be a good reason to actually recommend a PMM with a strong focus on verification and validation.

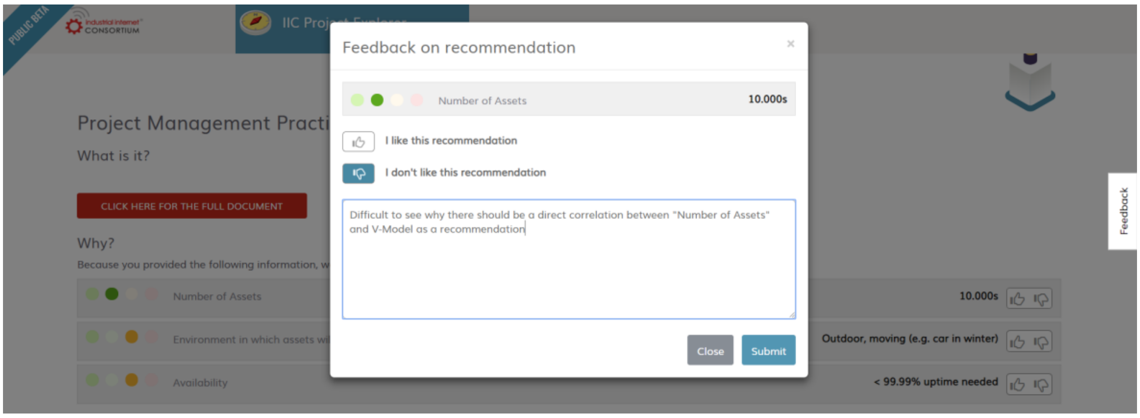

Figure 101: Providing feedback on recommendation rules

Consequently, the test user is now using the new feedback mechanism to comment on these inconsistencies. As an example, the UI for leaving feedback on the number of Assets as a reason for recommending V-Model is provided (see Figure 101). The comment describes the issue from the end-user’s perspective.

9.4.4 Dashboard for Content Authors

Having completed the project assessment and provided feedback on the recommendations, a new test user role now takes shape: the content author. For content authors, the IIC system is providing a new dashboard based on the requirements identified in iteration II of this work (see section 7.2). This dashboard can be seen in Figure 102.

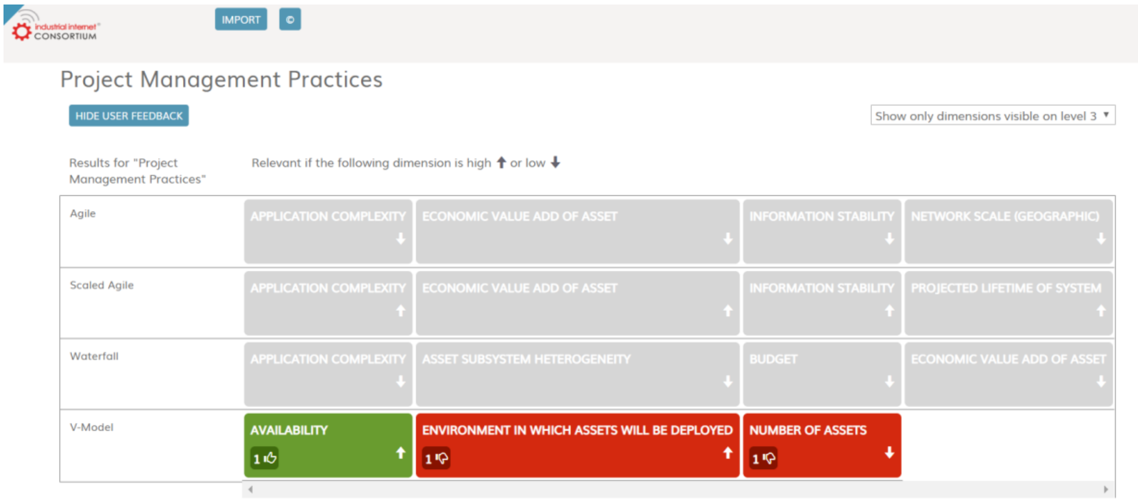

Figure 102: Dashboard for content authors

The new dashboard shows all relevant rule matches for a given result set. In this example, the result set is project management practices. The row on the left shows all results from this result set, e.g. Agile, Scaled Agile, etc. To the right of each result, the dashboard shows all rules that have been defined for a given result. For example, in the bottom row, there are three rules: availability, environment and number of assets. The arrow indicates whether this is a high or low rule, i.e., whether this rule is triggered if the project is indicating it has a high or low match here. Furthermore, some of the rules in the dashboard are colored red or green. In addition, a small icon indicates the number of up-votes and down-votes for a given rule. The color red signifies that the majority of the votes are down-votes, and green shows up-votes. In the example given, the three examples of feedback left by the torque domain expert can be seen, as described in 9.4.3. Availability was seen as a valid rule, while environment and number of assets were seen as questionable.

9.4.5 Incorporating the End-User Feedback

Reviewing the feedback from the domain expert helps the content author to review the rules used by the system to make recommendations. In this case, a discussion between the experts in the test team concluded that the problem with some of the recommendations was partially based on a lack of suitable dimensions in the project assessment. In the case of the PMM result set, an inability to express the following was identified:

- The expected team size (or even the need to have multiple teams), e.g., to differentiate between SCRUM and scales agile.
- The stability (or volatility) of the functional requirements, e.g., to differentiate between waterfall and any of the agile PMMs

Since the expected team size should be a result of the assessment and not a question asked to the project manager, the overall functional complexity of the project as a strong indicator for team size was agreed upon. Both volatility and functional complexity were added as Ignite dimensions to the IIC system for testing purposes. They were also added as candidates for Ignite 1.1; see section 12.1.1.1. In addition, the test authors also did a review of the result sets and changed them according to the proposed PMM in this thesis.

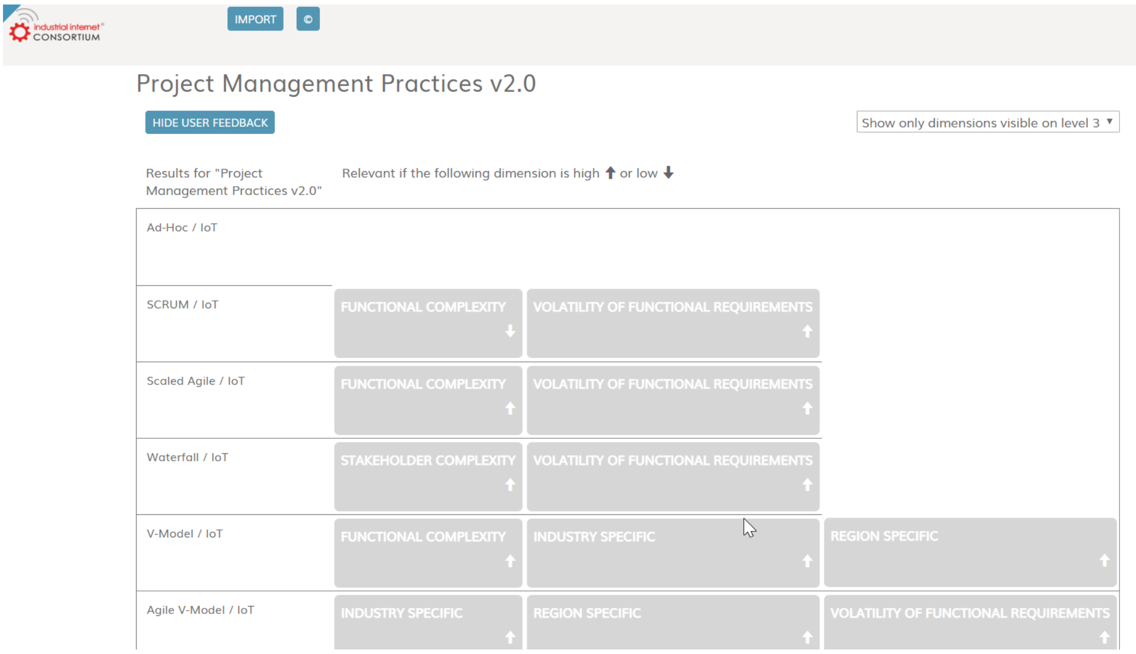

Figure 103: Revised rules for PMM result set

Figure 103 shows the new dashboard with the revised rules for the PMM result set. For example, scales agile for IoT is now deemed relevant if the project has a high functional complexity and also a high volatility of the functional requirements.

9.4.6 Reentering the Test Loop: End-User Perspective

Switching back to the test user in the role of torque domain expert, the project assessment was extended to include the new project dimensions, especially volatility and complexity.

Figure 104: New PMM recommendation

After reviewing the project assessment, the recommendations engine is run again. The results are shown in Figure 104. The IIC version of IgniteWorx is now showing scaled agile/IoT as a recommendation of the PMM for the torque project. This is a good match for the real project. As such, this partial test can be seen as a success.

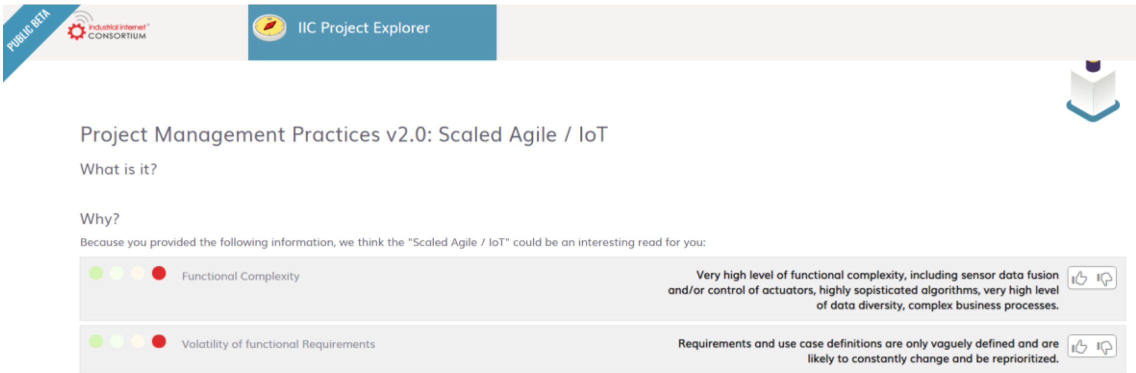

Figure 105: Details of the new PMM recommendation

Figure 106 shows the details of the recommendation. It can be clearly seen that scaled agile/IoT was recommended because the torque project has a high functional complexity and a high volatility of functional requirements.

9.5 Test Results

The test has shown the basic feasibility of the approach developed in iteration II of this research:

- Domain experts are now able to provide detailed comments on the recommendations and the reasons they were made.
- The new dashboard implemented by the IIC system seems to be an efficient tool for reviewing results sets and rule definitions in general and the feedback of the domain experts in particular.
- The content authoring features of the IIC system enable content and rule authors to efficiently modify all main content types, including results sets and rules.
- The recommendation engine is picking up all changes to content and rules in real time and makes improved recommendations based on the improved content.
- The end-user is presented with the optimized results, as expected.

While this test was based only on one iteration and one particular result set, it seemed to have confirmed that the results of iteration II support the continuous optimization of the system.

9.6 Overall Findings in Iterations I and II

In addition to the very focused tests summarized in the previous section, iterations I and II included many more tests of various aspects of the IIC system. A summary of the key findings is provided in the following.

9.6.1 System and Architecture

In section 3.3.1, a set of architecture evaluation criteria for IgniteWorx are defined, based on the architecture tradeoff analysis method (ATAM). Section 3.3.6 uses these criteria to do an evaluation of the proposed IgniteWorx architecture on the conceptual level. The following does another evaluation based on the same criteria, this time from the point of view of the IIC implementation of an IgniteWorx system, based on the findings from evaluations I and II:

- Performance: Performance of the IIC implementation of IgniteWorx is acceptable, both for end-users as well as content authors. Average response times of the system are below or at least close to one second, which is often the minimum performance requirement for web-based applications (Nielsen, 2010).
- Reliability: During the tests performed as part of evaluations I and II, no reliability issues were found.
- Availability: Availability has been acceptable during the tests performed. Given that the IIC system was still relatively new at the time of writing, occasional downtimes for system upgrades were experienced but within acceptable limits.
- Security: The IIC implementation is storing all IoT project assessments locally in the end-user's browser. This seems to be a good solution to ensure that no critical project data is accessed by outsiders. However, no detailed security assessment of the solution was performed as part of this thesis.
- Modifiability: As could be seen especially during iteration II, the MDA approach chosen for the IgniteWorx design and implemented by the IIC instance has helped ensure a high level of modifiability. All required extensions, e.g., for capturing end-user feedback, could be easily implemented by the development team. Also, iteration II has shown that all content modifications required for improving the system's recommendations could be done via the provided web UIs for editing project dimensions, result sets, and rules.
- Portability: This criteria was not evaluated as part of this thesis.
- Functionality: All functionality described as part of iterations I and II in this thesis is supported by the IIC instance.
- Variability: The focus of iterations I and II of this thesis was on phase I of the anticipated system evolution (explicit semantics). Consequently, no evidence was provided that the system would also support automated learning. For this, the conceptual evaluation will have to suffice for now.
- Subsetability: It seemed that the model designs provided by this thesis for project dimensions, result sets, and rules really enabled loosely coupled domains, which are easily extensible. The required reuse of Ignite dimensions was supported.
- Conceptual integrity: No deep architectural evaluation regarding the conceptual integrity of the design of the system beyond the conceptual evaluation was done.

Overall, it seems that the IIC implementation of the IgniteWorx concepts produced satisfactory results, with a positive evaluation of most of the above criteria.

9.6.2 Content Structure and Content Quality

Overall, the proposed approach of a content structure for IgniteWorx that combines Ignite dimensions with results sets and rules seems to be working, as far as can be determined from evaluations I and II.

Especially during evaluation II, one interesting finding was that the original idea of always deriving exactly one recommendation per result set and project assessment could be further qualified by differentiating between:

- Recommendations: As per the original design, for each project assessment, the system selects the best matching results from each result set and presents it as a recommendation.
- Rankings: For some result sets, a ranking seems better suited than an actual recommendation. For example, a future version of IgniteWorx could present the list of all potential project risks as a rating. This would make sense because it seems likely that a project will not only face a single risk but multiple risks with different likelihoods.

In addition, the concept of Project Classifications could be introduced: In a future version, it would be desirable to be able to group certain project dimensions together to create project classifications. These classifications could then be used to simplify the rules required for recommendations and rankings.

Figure 106 below shows how these three different concepts could work together. For example, a new classification, “Project Category,” could be created to provide an initial assessment of the project (e.g., from “small project” to “very large project”). This project category could then be used to simplify the matching rules, e.g., to the recommendation for the most suitable PMM. Of course, one could also simply make “Project Category” one of the primary project dimensions, but it feels like it would defeat the purpose of the entire IgniteWorx approach if the end-user had to make such a project categorization him- or herself instead of relying on the system to do this for him or her.

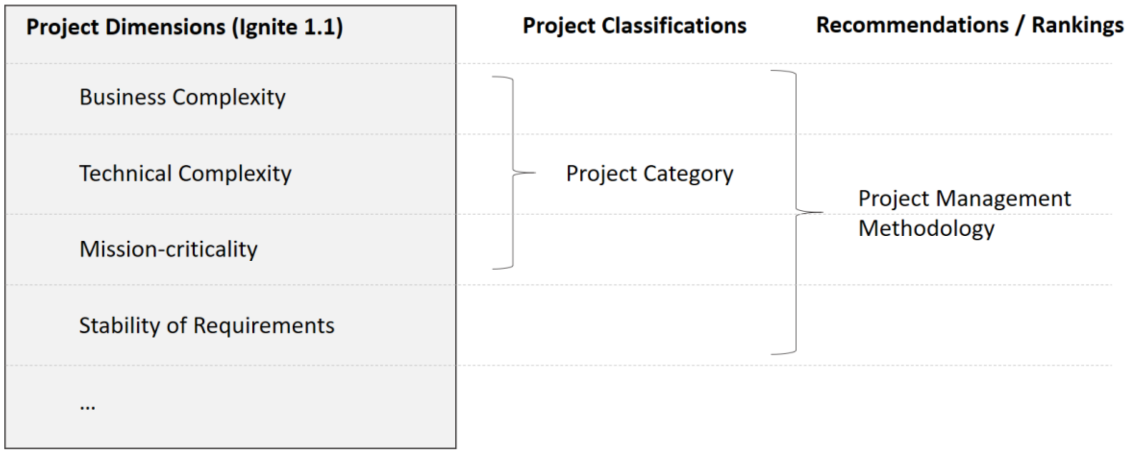

Figure 106: Project classifications versus recommendations and rankings

However, despite this small improvement area, overall, the assessment of the content structure is positive: the system is working and producing results, as was expected. Also, it seems like the content structure is allowing easy model extensions, as could be seen in evaluation II. Of course, the long-term maintainability and extensibility of the system functionality, which depends on the content structure, must still be proven.

Regarding the content quality in the IIC system, it is more difficult to make an assessment because of the subjective nature of content quality. Looking at the three main content types, the following can be said about the content quality at the time of writing:

- Project Dimensions: IIC has derived its own version of project dimensions from the original Ignite. Many of the proposed changes seem to make sense, and Ignite 1.1, described in the appendix, is proposing to include them. However, it would also be recommended to review the IIC project dimensions again with respect to the findings of this thesis. See section 12.1 for more details.
- Result Sets: The version of the IIC system tested as part of evaluations I and II did not include all results sets proposed by this thesis. Instead, some additional result sets were included, with IIC-specific content like testbeds (see Table 26 for a detailed comparison). Building on the IIC content, it would be recommended to also review which additional inputs from this thesis could be considered here.
- Rules: It seems that the currently provided matching rules could benefit from more user feedback, as it is now enabled by the output of iteration II. This was only reviewed in detail for one result set and should be done for all others as well.

10 Conclusions and Outlook

This section provides conclusions and a summary of key findings of this thesis as well as an outlook for future research.

10.1 Conclusions

The following section provides the general conclusion of this thesis, including the results of iterations I and II. First, an analysis of the overall research goals is provided. Second, the practical implications and limitations are looked at.

10.1.1 Research Goals

The research question defined in section 0 was “How can a system-supported methodology be designed and evaluated that supports project managers in finding a suitable setup for their IIoT projects?” To answer this question, IgniteWorx was designed and evaluated in two iterations.

The research question is accompanied by seven research aspects to help better understand the different facets of the question. The following provides a discussion on how each of these aspects was addressed as part of this work.

What are the main challenges of IIoT project managers today?

This was addressed in detail in section 1.4, including a discussion of IIoT projects risks and failure rates. The risks identified included the normal risks of any IT project, combined with the added risks of an IoT project. These include added functional complexity, added technical complexity, and added security risks.

As one important challenge, especially from the project manager’s perspective, the lack of well-established IoT project methodologies was cited. This means that project managers cannot access well-documented best practices to run their projects more efficiently and better manage IoT-specific project risks. This finding was also an important motivation for this thesis.

Which methodologies and frameworks already exist to help address them?

First, a number of methodologies and frameworks for normal IT projects could be identified, which are also useful in the context of IoT (see section 3.1.2). Second, a number of IoT-specific methodologies and frameworks could be identified (see, for example, sections 2.4, 1.4.2.2, and 3.2.2.2). While they may not yet be widely established and proven, they provided important impulses for the work outlined in this thesis.

What actually constitutes the setup of an IIoT project?

Section 3.1.2 proposes ten important practice areas that must be addressed during the setup of an IIoT project (referred to as result sets in this work): IoT project management methodology, solution design, technology selection, resource acquisition, cost estimate, risk management, trust and security, reliability and resilience, verification and validation, and service operations.

How can a methodology or framework be designed that generalizes the problem and provides actionable guidance for IIoT project setup?

This thesis proposes the IgniteWorx approach, which combines a detailed IoT project assessment (based on the Ignite dimensions) with a framework for deriving concrete recommendations for project managers based on the ten result sets, plus a set of mapping rules.

How should the ideal design of a system look that supports such a methodology?

This thesis proposes a hybrid architecture, which combines explicit knowledge representation using an MDA approach with aspects of other system categories, including expert systems, recommender systems, semantic reasoning systems, and survey engines. Details are described in section 3.3.

How can real-world experience and feedback be incorporated back into the system to optimize the quality of the methodology over time?

Iteration II of this thesis suggest an extension to the initial IgniteWorx design that is specifically aimed at supporting continuous content improvement. This is based on combining end-user feedback on recommendations with a dashboard for content authors to evaluate this feedback. See section 7 for details.

How can the quality of the proposals developed in this thesis be improved?

As discussed in section 2.7, this work assumes that the content of an IgniteWorx system will evolve in phases. In the first phase, continuous manual content improvement will be required, as enabled by iteration II. In the next phases, once the content in the system is reaching critical mass, advanced techniques like self-learning could be applied to improve the quality of the recommendations made by the system.

10.1.2 Practical Implications and Limitations

From the point of view of the end-user, evaluations I and II have shown that IgniteWorx is providing already practical benefits for end-users like IoT project managers:

- Being able to do a structured project assessment alone was seen by some interview partners as a significant benefit. The ability to take a holistic look at the nonfunctional aspects of an IoT project was seen as an important step toward minimizing project risks. In a sense, IgniteWorx can function here as a structured project checklist that ensures that no important aspects are forgotten. While this was theoretically already possible with Ignite alone, some interviewees stated that being able to do this online and to include multiple project stakeholders in the process was seen as a benefit.

- The recommendations provided by the IIC system at the time of writing were not something that the interviewees believed they could follow without further validations on their own. Also, the range of topics captured was not sufficiently complete. However, the general feedback was that the approach of providing guidance and also well-structured options to choose from was seen as valuable since this can trigger important discussions in the project teams regarding the best possible project setup. In this sense, one way forward in the near future may be to go from individual recommendations more toward a ranking of options, as proposed in the previous section.

From the point of view of an IgniteWorx content administrator, the IIC implementation shows that the system architecture and content structure proposed by this thesis works, at least within the limitations of evaluations I and II. The results of iteration I were important to bootstrap the system, as manifested by the IIC implementation. The results of the design iteration II were important to ensure that a mechanism for the continuous optimization of content is provided. Evaluation II showed that the concept is feasible in principle. As per the general assumption of this thesis regarding the expected evolution of an IgniteWorx system, the IIC example is at the early stages of phase I. It will be interesting to observe in the long term how the system will yield continuous content improvements in phase I of the system evolution and eventually move toward a more automated learning-based approach, as envisioned for phases II and III.

10.2 Outlook

Finally, the following section provides an outlook on potential directions that the research on IgniteWorx could take in the future.

10.2.1 Specialized Result Sets

The result sets proposed in this thesis are still on a relatively high level. For example, the result set for risk management only provides guidance on a suitable methodology for the project but does not contain recommendations for actual risks. The same applies to many of the other result sets. Consequently, during the development of IgniteWorx and the work on the IIC system, a number of ideas for more specialized result sets were developed, which are summarized in the following:

- TCO Analysis: An extension for the analysis of the total cost of ownership (TCO) would combine more specialized project dimensions with more concrete recommendations for the expected cost range of a given project.
- Risk Assessment: Similarly, a specialized extension of IgniteWorx could provide a ranked list of actual risks that the project might face.
- Security Assessment: A specialized security assessment could help identify likely attack vectors and make recommendations for concrete countermeasures.
- Industry 4.0 Extensions: Currently, the focus of IgniteWorx is mainly on the IoT solution and less on the potential impact on the design, sourcing, and manufacturing of the asset or physical product. This is fine for retrofit projects. However, line-fit IoT projects could benefit from an extension that looks at this more closely.

10.2.2 Review of System Adoption and Content Evolution

The development of IgniteWorx is based on the assumption that an instance of IgniteWorx will go through an evolution in three phases, as described in section 2.7. Future research could evaluate the validity of this assumption and propose measures to speed up the evolution.

10.2.3 Matching Algorithms and Self-Learning

It seems that there would be abundant room for future research in the area of the matching algorithms and also the vision of a system based more on automated or even self-learning algorithms. As identified in section 2.7., one clear limitation here is the availability of reference content, which would be a prerequisite for any kind of self-learning algorithms.

However, for the current approach, which relies on explicit knowledge representation and semantic reasoning-like algorithms, it would be interesting to research potential improvements in the matching algorithms.

Furthermore, it would be beneficial to develop an algorithm to support the analysis of the quality of the matching rules. For example, it could detect rule inconsistencies or situations where rules trees are not sufficiently balanced to ensure a fair and even consideration of all potential options.

10.2.4 Addressing Other IoT Project Phases

The focus of IgniteWorx as part of this thesis was exclusively on the project setup phase. It could be enlightening to do research with a focus on other research phases.

For example, the start of production (SOP) after the initial system development is another project phase that could justify its own specialized set of project dimensions and result sets.

10.2.5 Generalize the Approach for Other Project Categories

IgniteWorx as it was developed in this thesis has a clear focus on IoT projects. It could be interesting to investigate how far a version of IgniteWorx could be developed to focus on other kinds of projects, e.g., AI projects, blockchain projects, etc. One could even consider a more generic version of IgniteWorx that would be suitable to IT projects in general.

10.3 Thank You and Getting in Touch

Dear Reader,

Thank you very much for your interest in IgniteWorx and this research project. Again, thanks also to everybody who contributed to this exciting project. In case this work has triggered your interest and you would like to get involved in the future development of IgniteWorx, please contact me via LinkedIn (de.linkedin.com/in/dslama).

Best regards,

Dirk Slama

11 References

Acciani, G., Brunetti, G. and Fornarelli, G. (2006) ‘Application of Neural Networks in Optical Inspection and Classification of Solder Joints in Surface Mount Technology’, *IEEE Transactions on Industrial Informatics*, vol. 2, no. 3, pp. 200–209.

Alkhalid, A. (2018) *Bridging the Gap Between GUI Functional System Testing and Functional System Logic Testing,* Carleton University.

Alkhalid, A. and Labiche, Y. (2016) ‘Comparing GUI Functional System Testing with Functional System Logic Testing-An Experiment’.

Alwadain, A., Fielt, E., Korthaus, A. and Rosemann, M. (2013) ‘A Comparative Analysis of the Integration of SOA Elements in Widely-Used Enterprise Architecture Frameworks’, *International Journal of Intelligent Information Technologies*, vol. 9, no. 2, pp. 54–70.

Ancveire, I., Gailite, I., Gailite, M. and Grabis, J. (2015) ‘Software Delivery Risk Management: Application of Bayesian Networks in Agile Software Development’, *Information Technology and Management Science*, vol. 18, no. 1, p. 26.

Andrade, C. E., Byers, S. D., Gopalakrishnan, V., Halepovic, E., Majmundar, M., Poole, D. J., Tran, L. K. and Volinsky, C. T. (2017) ‘Managing Massive Firmware-Over-The-Air Updates for Connected Cars in Cellular Networks’, *Proceedings of the 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services*, pp. 65–72.

Archer, D. (2009) *Fastening Threads: Torque Control vs. Angle Control* [Online]. Available at assemblymag.com (Accessed 13 April 2019).

Arnott, D. and Pervan, G. (2005) ‘A critical analysis of decision support systems research’, *Journal of Information Technology*, vol. 20, no. 2, pp. 67–87.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S. and Holley, K. (2008) ‘SOMA: A method for developing service-oriented solutions’, *IBM Systems Journal*, vol. 47, pp. 377–396.

AXELOS (2019) *ITIL - IT Service Management* [Online]. Available at www.axelos.com (Accessed 13 April 2019).

Axelrod, W. (2009) ‘Investing in software resiliency’, vol. 22, pp. 20–25.

Ayala, I., Amor, M., Fuentes, L. and Troya, J. M. (2015) ‘A software product line process to develop agents for the iot’, *Sensors*, vol. 15, no. 7, pp. 15640–15660.

Babar, S., Stango, A., Prasad, N., Sen, J. and Prasad, R. (2011) ‘Proposed embedded security framework for Internet of Things (IoT)’, *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace &*

Electronic Systems Technology (Wireless VITAE). Chennai, India, 28.02.2011 - 03.03.2011, IEEE, pp. 1–5.

Badampudi, D., Wohlin, C. and Petersen, K. (2016) 'Software component decision-making: In-house, OSS, COTS or outsourcing-A systematic literature review', *Journal of Systems and Software*, vol. 121, pp. 105–124.

Bailey, J. (2019) *Everything You Need to Know about Asset Performance Management* [Online]. Available at www.ge.com (Accessed 13 April 2019).

Baker, B. N., Murphy, D. C. and Fisher, D. (1997) 'Factors affecting project success', *Project management handbook*, pp. 902–919.

Balaji, S. and Murugaiyan, M. S. (2012) 'Waterfall vs. V-Model vs. Agile: A comparative study on SDLC', *International Journal of Information Technology and Business Management*, vol. 2, no. 1, pp. 26–30.

Bardyn, J.-P., Melly, T., Seller, O. and Sornin, N. (2016) 'IoT: The era of LPWAN is starting now', *European Solid-State Circuits Conference, ESSCIRC Conference 2016: 42nd*, pp. 25–30.

Bauer, M., Boussard, M., Bui, N., Carrez, F., Jardak (SIEMENS, C., Loof (ALUBE, J. de, Magerkurth (SAP, C., Meissner, S., Nettsträter (FhG IML, A., Olivereau, A., Thoma (SAP, M., Joachim, W., Stefa (CSD/SUni, J. and Salinas, A. (2013) *Internet of Things – Architecture IoT-A Deliverable D1.5 – Final architectural reference model for the IoT v3.0* [Online].

Bayes, T., Price, R. and Canton, J. (1763) 'An essay towards solving a problem in the doctrine of chances'.

Belbachir, A. N., Lera, M., Fanni, A. and Montisci, A. (2005) 'An automatic optical inspection system for the diagnosis of printed circuits based on neural networks', *Fourtieth IAS Annual Meeting. Conference Record of the 2005 Industry Applications Conference, 2005.* Hong Kong, China, 2-6 Oct. 2005, IEEE, pp. 680–684.

Belton, V. and Stewart, T. (2002) *Multiple criteria decision analysis: an integrated approach*, Springer Science & Business Media.

Benbasat, I., Goldstein, D. K. and Mead, M. (1987) 'The Case Research Strategy in Studies of Information Systems', *MIS Quarterly*, vol. 11, no. 3, p. 369.

Bennett, C. and Tseitlin, A. (2012) 'Chaos monkey released into the wild', *Netflix Blog*.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) 'The Semantic Web', *Scientific American*, vol. 284, no. 5, pp. 34–43.

Birch, J., Rivett, R., Habli, I., Bradshaw, B., Botham, J., Higham, D., Jesty, P., Monkhouse, H. and Palin, R. (2013) 'Safety cases and their role in ISO 26262 functional safety assessment', *International Conference on Computer Safety, Reliability, and Security*, pp. 154–165.

Bleakley, G., Buchheit, M., Chigani, A., Crawford, M., Durand, J., Karmarkar, A., LeBlanc, J.-P., Lin, S.-W., Martin, R., Miller, B., Murphy, B. and Schrecker, S. (2015) *Industrial Internet Reference Architecture* [Online]. Available at www.iiconsortium.org (Accessed 13 April 2019).

Blomqvist, E. (2014) 'The use of Semantic Web technologies for decision support-a survey', *Semantic Web*, vol. 5, no. 3, pp. 177–201.

Bloomfield, R., Cazin, J., Craigen, D., Juristo, N., Kesseler, E. and Voas, J. (2004) 'Validation, verification and certification of embedded systems'.

BMC (2016) *ITIL® Processes & Best Practices* [Online]. Available at www.bmc.com.

Bña, S. (2015) 'Capability Driven Development: An Approach to Designing Digital Enterprises', *Business & Information Systems Engineering*, vol. 57, no. 1, pp. 15–25.

Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R. and Selby, R. (1995) 'Cost models for future software life cycle processes: COCOMO 2.0', *Annals of software engineering*, vol. 1, no. 1, pp. 57–94.

Boehm, B. W. (1991) 'Software risk management: principles and practices', *IEEE software*, vol. 8, no. 1, pp. 32–41.

Bolt Science (2018) *Methods of Tightening Threaded Fasteners* [Online], Bolt Science. Available at www.boltscience.com (Accessed 13 April 2019).

Borgovini, R., Pemberton, S. and Rossi, M. (1993) *Failure mode, effects, and criticality analysis (FMECA),* RELIABILITY ANALYSIS CENTER GRIFFISS AFB NY.

Bouwers, E., Visser, J. and van Deursen, A. (2009) 'Criteria for the evaluation of implemented architectures', *2009 IEEE International Conference on Software Maintenance*, pp. 73–82.

Bowles, J. B. (1998) 'The new SAE FMECA standard', *Proceedings of the annual reliability and maintainability symposium*, pp. 48–53.

Brahmachary, A. (2018) *ITIL Service Operation Processes Explained* [Online] (Accessed 13 April 2019).

Brooks, F. P. (1975) *The mythical man-month: Essays on software engineering / Frederick P. Brooks, Jr*, Reading, Mass., Addison-Wesley Pub. Co.

Budgen, D. and Brereton, P. (2006) 'Performing systematic literature reviews in software engineering', *Proceedings of the 28th international conference on Software engineering*, pp. 1051–1052.

Bund (1997) *V-Modell: Entwicklungsstandard für IT-Systeme des Bundes (ESdIT)*, Bundesanzeiger.

Buntz, B. (2017) *The 9 most popular types of IIoT use cases* [Online], IoT World Today. Available at www.iotworldtoday.com (Accessed 13 April 2019).

Burstein, F. and Holsapple, C. W. (2008) *Handbook on decision support systems 2: variations*, Springer Science & Business Media.

Byun, J., Kim, B. W., Ko, C. Y. and Byun, J.-W. (2017) '4G LTE network access system and pricing model for IoT MVNOs: spreading smart tourism', *Multimedia Tools and Applications*, vol. 76, no. 19, pp. 19665–19688.

Carlin, C.-G. and Lennart, H. B. (2006) *Method for quality assurance of screw joint tightening* US7702493B2.

Casadesus-Masanell, R. and Ricart, J. E. (2011) 'How to design a winning business model', *Harvard Business Review*, vol. 89, 1/2, pp. 100–107.

Chan, F.T.S., Chan, M.H. and Tang, N.K.H. (2000) 'Evaluation methodologies for technology selection', *Journal of Materials Processing Technology*, vol. 107, 1-3, pp. 330–337.

Chapman, C. (1997) 'Project risk analysis and management—PRAM the generic process', *International Journal of Project Management*, vol. 15, no. 5, pp. 273–281.

Choi, J. (2002) 'A rule-based expert system using an interactive question-and-answer sequence', *Dept. of Geography University of Georgia*.

Chulani, S., Boehm, B. and Steece, B. (1999) 'Bayesian analysis of empirical software engineering cost models', *IEEE Transactions on software engineering*, vol. 25, no. 4, pp. 573–583.

Cisco (2017) *The Journey to IoT Value: Challenges, Breakthroughs, and Best Practices* [Online], Cisco. Available at www.slideshare.net (Accessed 13 April 2019).

Cisco / Jasper (2016) *The Hidden Costs of Delivering IIoT Services* [Online]. Available at www.cisco.com (Accessed 13 April 2019).

Clark-Ginsberg, A. (2016) *What's the Difference between Reliability and Resilience?* [Online].

Clements, P., Kazman, R. and Klein, M. (2002) *Evaluating software architectures: Methods and case studies / Paul Clelents, Rick Kazman, Mark Klein*, Boston, MA, Addison-Wesley.

Collins, T. (2014) *A Methodology for Building the Internet of Things* [Online]. Available at www.slideshare.net (Accessed 13 April 2019).

Dash, S. and Acharya, A. A. (2011) 'Cost Estimation for Distributed Systems Using Synthesized Use Case Point Model', in *Advances in Computing, Communication and Control,* Springer, pp. 161–169.

Debardelaben, J. A., Madisetti, V. K. and Gadient, A. J. (1997) 'Incorporating cost modeling in embedded-system design', *IEEE Design & Test of Computers*, vol. 14, no. 3, pp. 24–35.

Delic, K. (2016) 'On resilience of IoT systems', *Ubiquity*, vol. 2016, pp. 1–7.

DEMELO, A. and SANCHEZ, A. (2008) 'Software maintenance project delays prediction using Bayesian Networks', *Expert Systems with Applications*, vol. 34, no. 2, pp. 908–919.

Deterding, S., Dixon, D., Khaled, R. and Nacke, L. (2011) 'From game design elements to gamefulness: defining gamification', *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pp. 9–15.

Deutsch, P. (1994) 'The eight fallacies of distributed computing', *URL: http://today. java. net/jag/Fallacies. html*.

Dey, P. K. (2002) 'Project risk management: a combined analytic hierarchy process and decision tree approach', *Cost Engineering*, vol. 44, no. 3, pp. 13–27.

Dhaneshwar, S. (2017) *Insights into FOTA for Automotive* [Online]. Available at www.linkedin.com (Accessed 13 April 2019).

Dhillon, B. (2015) *Classifying Different Types of Recommender Systems* [Online]. Available at www.bluepiit.com (Accessed 13 April 2019).

Dick, B. (2002) 'Convergent interviewing. Session 8 of Areol-action research and evaluation on line', *Retrieved March*, vol. 1, p. 2007.

Din, C. and Rubio, B. (2016) 'State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing', *Journal of Network and Computer applications*, vol. 67, pp. 99–117.

Disterer, G. (2013) 'ISO/IEC 27000, 27001 and 27002 for information security management', *Journal of Information Security*, vol. 4, no. 02, p. 92.

DOT, U. S. (2007) 'Systems engineering for intelligent transportation systems', *Federal Highway Administration & Federal Transit Administration*.

Durrani, T. S., Forbes, S. M., Broadfoot, C. and Carrie, A. S. (1998) 'Managing the technology acquisition process', *Technovation*, vol. 18, 8-9, pp. 523–587.

Dvir, D., Raz, T. and Shenhar, A. J. (2003) 'An empirical analysis of the relationship between project planning and project success', *International Journal of Project Management*, vol. 21, no. 2, pp. 89–95.

Easterbrook, S., Singer, J., Storey, M.-A. and Damian, D. (2008) 'Selecting empirical methods for software engineering research', in *Guide to advanced empirical software engineering,* Springer, pp. 285–311.

Ebert, C., Gallardo, G., Hernantes, J. and Serrano, N. (2016a) 'DevOps', *IEEE software*, vol. 33, no. 3, pp. 94–100.

Ebert, C., Gallardo, G., Hernantes, J. and Serrano, N. (2016b) 'DevOps', *IEEE software*, vol. 33, no. 3, pp. 94–100.

Eclipse Foundation *Eclipse Ignite|IoT* [Online]. Available at projects.eclipse.org (Accessed 13 April 2019).

Eisenhardt, K. M. (1989) 'Building theories from case study research', *Academy of management review*, vol. 14, no. 4, pp. 532–550.

ENISA (2017) *Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures* [Online]. Available at www.enisa.europa.eu (Accessed 13 April 2019).

Eric Graves (2016) *Project Risk Management for Hardware Development: Examples of project risks* [Online]. Available at www.playbookhq.co (Accessed 13 April 2019).

España, S., González, T., Grabis, J., Jokste, L., Juanes, R. and Valverde, F. (2014) 'Capability-Driven Development of a SOA Platform: A Case Study', in van der Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M. J., Szyperski, C., Iliadis, L., Papazoglou, M. and Pohl, K. (eds) *Advanced Information Systems Engineering Workshops,* Cham, Springer International Publishing, pp. 100–111.

Eveleens, J. L. and Verhoef, C. (2010) 'The rise and fall of the chaos report figures', *IEEE software*, vol. 27, no. 1, pp. 30–36.

Farooq, M. U., Waseem, M., Khairi, A. and Mazhar, S. (2015) 'A critical analysis on the security concerns of internet of things (IoT)', *International Journal of Computer Applications*, vol. 111, no. 7.

Feigenbaum, E. A. (1977) *The art of artificial intelligence. 1. Themes and case studies of knowledge engineering,* Stanford Univ CA Dept of Computer Science.

Felfernig, A., Friedrich, G., Jannach, D. and Zanker, M. (2015) 'Constraint-Based Recommender Systems', in Ricci, F., Rokach, L. and Shapira, B. (eds) *Recommender Systems Handbook,* Boston, MA, Springer US, pp. 161–190.

Ferdinand-Steinbeis-Institut (2018) *IIC German Country Team* [Online], Ferdinand-Steinbeis-Institut. Available at steinbeis-fsti.de (Accessed 13 April 2019).

Fewell, J. (2011) *Fixed price agile projects: making the impossible possible* [Online], Paper presented at PMI® Global Congress 2011—North America, Dallas, TX. Newtown Square, PA, PMI. Available at www.pmi.org (Accessed 13 April 2019).

Fortino, G. and Russo, W. (2012) ‘ELDAMeth: An agent-oriented methodology for simulation-based prototyping of distributed agent systems’, *Information and Software Technology*, vol. 54, no. 6, pp. 608–624.

Fowler, K. R., Das, S. and Dyer, S. A. (2019) ‘A Study of the Build-Versus-Buy Decision in Developing Embedded Systems’, *IEEE Systems Journal*.

Fu, K. S. (1982) ‘Pattern Recognition For Automatic Visual Inspection’. Arlington, Wednesday 5 May 1982, SPIE, pp. 12–19.

Fuerst, C. (2001) *Web survey tool builder and result compiler* [Online], Google Patents.

Gartner Group, I. (2019) *IT Glossary* [Online]. Available at https://www.gartner.com/it-glossary (Accessed 13 April 2019).

Gasser, M., Goldstein, A., Kaufman, C. and Lampson, B. (1989) ‘The Digital distributed system security architecture’, *Proceedings of the 12th National Computer Security Conference*, pp. 305–319.

Gassmann, O., Schuhmacher, A., Zedtwitz, M. von and Reepmeyer, G., eds. (2018) *Leading Pharmaceutical Innovation*, Cham, Springer International Publishing.

Ghodsypour, S. H. and O’Brien, C. (1998) ‘A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming’, *International Journal of Production Economics*, vol. 56, pp. 199–212.

Giannopoulos, N. (2006) ‘Estimating the design and development cost of electronic items’.

Gilutz, H. (1988) ‘Automatic Optical Inspection Systems’, in Riley, F. (ed) *The Electronics Assembly Handbook,* Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 346–352.

Giray, G. and Tekinerdogan, B. (2018) ‘Situational Method Engineering for Constructing Internet of Things Development Methods’, in Shishkov, B. (ed) *Business Modeling and Software Design,* Cham, Springer International Publishing, pp. 221–239.

Glass, R. L. (2005) ‘IT Failure Rates-70% or 10-15%?’, *IEEE software*, vol. 22, no. 3.

Gope, J. and Jain, S. K. (2017) ‘A survey on solving cold start problem in recommender systems’, *Computing, Communication and Automation (ICCCA), 2017 International Conference on*, pp. 133–138.

Grandison, T. and Sloman, M. (2000) ‘A survey of trust in internet applications’, *IEEE Communications Surveys & Tutorials*, vol. 3, no. 4, pp. 2–16.

Granstrand, O., Bohlin, E., Oskarsson, C. and Sjöberg, N. (1992) 'External technology acquisition in large multi-technology corporations', *R&D Management*, vol. 22, no. 2, pp. 111–134.

Grau, N. (2013) 'Standards and Excellence in Project Management – In Who Do We Trust?', *Procedia - Social and Behavioral Sciences*, vol. 74, pp. 10–20.

Griessnig, G. and Schnellbach, A. (2017) 'Development of the 2nd Edition of the ISO 26262', *European Conference on Software Process Improvement*, pp. 535–546.

Gupta, S. (2018) *A four-phase approach to independent verification and validation* [Online]. Available at gcn.com (Accessed 13 April 2019).

Haes, S. de, Debreceny, R. and van Grembergen, W. (2013) 'Understanding the core concepts in COBIT 5', *Isaca Journal*, vol. 5, pp. 1–8.

Hankel, M. (2015) 'The reference architectural model industrie 4.0 (rami 4.0)', *ZVEI, April*.

Hasani, Z., Kon-Popovska, M. and Velinov, G. (2014) 'Lambda architecture for real time big data analytic', *ICT Innovations*, pp. 133–143.

Haugeland, J. (1985) *Artificial intelligence: The very idea / John Haugeland*, Cambridge, Mass., MIT press.

Heemstra, F. J. and Kusters, R. J. (1991) 'Function point analysis: Evaluation of a software cost estimation model', *European Journal of Information Systems*, vol. 1, no. 4, pp. 229–237.

Hemmerdinger, R. (2014) *Predictive Maintenance Strategy for Building Operations: A Better Approach* [Online], Schneider Electric. Available at www.fmmedia.com.au (Accessed 13 April 2019).

Heorhiadi, V., Rajagopalan, S., Jamjoom, H., Reiter, M. K. and Sekar, V. (2016) 'Gremlin: Systematic resilience testing of microservices', *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pp. 57–66.

Hermansson, T. O. (2016) 'Quality assured tightening of screw joints', *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 230, no. 15, pp. 2588–2594.

Herrmann, J. (2001) 'Guideline for validation & verification real-time embedded software systems', *Information Technology for European Advancement*.

Hevner, A. and Chatterjee, S. (2010) 'Design science research in information systems', in *Design research in information systems,* Springer, pp. 9–22.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) *Design science in information systems research. MIS Q 28 (1): 75-105* [Online].

Hillson, D. (2003) 'Using a Risk Breakdown Structure in project management', *Journal of Facilities Management*, vol. 2, no. 1, pp. 85–97.

Holsapple, C. W. (2000) *Decision support systems: A knowledge based approach*, Thomson Learning.

Hu, Y., Zhang, X., Ngai, E.W.T., Cai, R. and Liu, M. (2013) 'Software project risk analysis using Bayesian networks with causality constraints', *Decision Support Systems*, vol. 56, pp. 439–449.

Huang, Y. and Bian, L. (2009) 'A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet', *Expert Systems with Applications*, vol. 36, no. 1, pp. 933–943.

Hughes, A. (2018) *On Premise vs. Cloud: Key Differences, Benefits and Risks* [Online]. Available at cleo.com (Accessed 13 April 2019).

Hukerikar, S. and Engelmann, C. (2017) 'Resilience design patterns: A structured approach to resilience at extreme scale', *arXiv preprint arXiv:1708.07422*.

Idrees, M. S., Schweppe, H., Roudier, Y., Wolf, M., Scheuermann, D. and Henniger, O. (2011) 'Secure Automotive On-Board Protocols: A Case of Over-the-Air Firmware Updates', in Strang, T., Festag, A., Vinel, A., Mehmood, R., Rico Garcia, C. and Röckl, M. (eds) *Communication Technologies for Vehicles,* Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 224–238.

IIC (2018a) *IIC ITS* [Online]. Available at www.iiconsortium.org (Accessed 13 April 2019).

IIC (2018b) *IIC website* [Online]. Available at www.iiconsortium.org (Accessed 13 April 2019).

IIC (2018c) *IoT Security Maturity Model: Description and Intended Use* [Online]. Available at www.iiconsortium.org (Accessed 13 April 2019).

IIC (2018d) *Resource Hub* [Online]. Available at hub.iiconsortium.org (Accessed 13 April 2019).

IIC and others (2016) 'Industrial Internet of Things Volume G4: Security Framework', *Ind. Internet Consort*, pp. 1–173.

(2018) *Industrial Internet Consortium* [Online]. Available at en.wikipedia.org (Accessed 13 April 2019).

Infante, J. L., Karnowski, M. J., Carlin, C. and Lawson, D. C. (2007) *System with multiple path fail over, fail back and load balancing* [Online], Google Patents.

ISO (2014) *Asset Management: ISO 55000:2014* [Online]. Available at www.iso.org (Accessed 13 April 2019).

ISO (2018) *17359: Condition monitoring and diagnostics of machines -- General guidelines* [Online]. Available at https://www.iso.org/standard/71194.html.

ITZBund (2005) *V-Modell XT* [Online]. Available at www.cio.bund.de (Accessed 13 April 2019).

Jackson, M., Crouch, S. and Baxter, R. (2011) 'Software evaluation: criteria-based assessment', *Software Sustainability Institute*.

Jacobson, I., Spence, I. and Ng, P.-W. (2017) 'Is there a single method for the Internet of Things?', *Queue*, vol. 15, no. 3, p. 20.

Jaikamal, V. (2009) *Model-based ecu development-an integrated mil-sil-hil approach,* SAE Technical Paper.

Jan, S. R., Shah, S. T. U., Johar, Z. U., Shah, Y. and Khan, F. (2016) 'An innovative approach to investigate various software testing techniques and strategies', *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Print ISSN*, 2395-1990.

Janczki, M., Becker, k., Jakab, L., Grf, R. and Takcs, T. (2013) 'Automatic Optical Inspection of Soldering', in Mastai, Y. (ed) *Materials Science - Advanced Topics,* InTech.

Javeri, P. (2018) *Architectural Patterns For IoT—Event Driven Architectures* [Online]. Available at medium.com (Accessed 13 April 2019).

Jedlitschka, A. and Pfahl, D. (2005) 'Reporting guidelines for controlled experiments in software engineering', *Empirical Software Engineering, 2005. 2005 International Symposium on*, 10-pp.

Jiacheng, C., Haibo, Z., Ning, Z., Peng, Y., Lin, G. and Xuemin, S. (2016) 'Software defined Internet of vehicles: architecture, challenges and solutions', *Journal of communications and information networks*, vol. 1, no. 1, pp. 14–26.

Jones, C. (1998) 'Minimizing the risks of software development', *Cutter IT journal*, vol. 11, pp. 13–21.

Jones, G. K., Lanctot, A. and Teegen, H. J. (2001) 'Determinants and performance impacts of external technology acquisition', *Journal of Business Venturing*, vol. 16, no. 3, pp. 255–283.

Jones, R. L. (1927) *Optical Inspection System* US1722751A.

Jones, T. C. (2007) *Estimating Software Costs*, 2nd edn, New York, NY, USA, McGraw-Hill, Inc.

Joslin, R. and Müller, R. (2014) 'The impact of project methodologies on project success in different contexts'.

Kahn, R. L. and Cannell, C. F. (1957) ‘The dynamics of interviewing; theory, technique, and cases’.

Kalmar, E. E. and Kertesz, A. (2017) ‘What Does I(o)T Cost?’, *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion - ICPE '17 Companion.* L'Aquila, Italy, 22.04.2017 - 26.04.2017. New York, New York, USA, ACM Press, pp. 19–24.

Karmarkar, A., Hirsch, F., Bournival, E., Buchheit, M., Simmon, E., Sobel, W., Joshi, R., Schrecker, S., Lin, S.-W., Molina, J., Rutt, T., Miller, B., Durand, J., Didier, P., Chigani, A., Torenbeek, R., Duggal, D., Martin, R., Bleakley, G., King, A., Lembree, R., Soroush, H., Garbis, J., Crawford, M., Harper, E., Raman, K., Witten, B., Ginter, A. and Meltzer, D. (2018) *The Industrial Internet of Things Volume G8: Vocabulary* [Online], Industrial Internet Consortium. Available at www.iiconsortium.org (Accessed 13 April 2019).

Kempa, M. and Mann, Z. A. (2005) ‘Model Driven Architecture’, *Informatik-Spektrum*, vol. 28, no. 4, pp. 298–302.

Kendall, M., Stuart, A., Ord, J. K. and O’Hagan, A. (1994) ‘Kendall’s advanced theory of statistics, volume 1: Distribution theory’, *Arnold, sixth edition edition*.

Khodakarami, V. and Abdi, A. (2014) ‘Project cost risk analysis: A Bayesian networks approach for modeling dependencies between cost items’, *International Journal of Project Management*, vol. 32, no. 7, pp. 1233–1245.

Kim, H., Kang, E., Broman, D. and Lee, E. A. (2017) ‘An Architectural Mechanism for Resilient IoT Services’, *Proceedings of the 1st ACM Workshop on the Internet of Safe Things*, pp. 8–13.

Kinney, S. L. (2006) *Trusted platform module basics: using TPM in embedded systems*, Elsevier.

Kirkpatrick, M. A., Jennings, W. and Zheng, D. (2007) *Reusable online survey engine* [Online], Google Patents.

Kitchenham, B., Al-Khilidar, H., Babar, M. A., Berry, M., Cox, K., Keung, J., Kurniawati, F., Staples, M., Zhang, H. and Zhu, L. (2008) ‘Evaluating guidelines for reporting empirical software engineering studies’, *Empirical Software Engineering*, vol. 13, no. 1, pp. 97–121.

Klubnikin, A. (2016) *How Much Does it Cost to Build IoT Solution?* [Online]. Available at r-stylelab.com (Accessed 13 April 2019).

Koh, K. (2009) ‘Eliminating False Calls With 3D AOI Technology’, *EPP Europe*, no. 4, pp. 40–43.

Krafzig, D., Banke, K. and Slama, D. (2005) *Enterprise SOA: Service-oriented architecture best practices / Dirk Krafzig, Karl Banke, Dirk Slama*, Indianapolis, IN, Prentice Hall Professional Technical Reference.

Kreiner, C., Messnarz, R., Riel, A., Ekert, D., Langgner, M., Theisens, D. and Reiner, M. (2013) 'Automotive knowledge alliance AQUA-integrating automotive SPICE, six sigma, and functional safety', *European Conference on Software Process Improvement*, pp. 333–344.

Kwak, Y. H. and Stoddard, J. (2004) 'Project risk management: lessons learned from software development environment', *Technovation*, vol. 24, no. 11, pp. 915–920.

Lasi, H. (2012) 'Decision Support within Knowledge-Based Engineering-a Business Intelligence-Based Concept'.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T. and Hoffmann, M. (2014) 'Industry 4.0', *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242.

Latifi, F. and Zarrabi, H. (2017) 'A COBIT5 Framework for IoT Risk Management'.

Law, E. L.-C. and Abrahão, S. (2014) 'Interplay between User Experience (UX) evaluation and system development', *International Journal of Human-Computer Studies*, vol. 72, no. 6, pp. 523–525.

Law, E. L.-C., van Schaik, P. and Roto, V. (2014) 'Attitudes towards user experience (UX) measurement', *International Journal of Human-Computer Studies*, vol. 72, no. 6, pp. 526–541.

Lee, E., Park, Y. and Shin, J. G. (2009) 'Large engineering project risk management using a Bayesian belief network', *Expert Systems with Applications*, vol. 36, no. 3, pp. 5880–5887.

Lee, P. M. (2012) *Bayesian statistics: An introduction / Peter M. Lee*, 4th edn, Chichester, West Sussex, Hoboken, N.J., Wiley.

Leffingwell, D. (2016) *SAFe\textregistered for Lean Software and Systems Engineering*, Addison-Wesley Professional.

Li, L., Xiaoguang, H., Ke, C. and Ketai, H. (2011) 'The applications of wifi-based wireless sensor network in internet of things and smart grid', *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, pp. 789–793.

Liang, X. (2012) *Recommender System: Algorithms and Architecture* [Online]. Available at medium.com (Accessed 13 April 2019).

Lin, S.-W., Miller, B., Durand, J., Joshi, R., Didier, P., Chigani, A., Torenbeek, R., Duggal, D., Martin, R., Bleakley, G. and others (2015) 'Industrial internet reference architecture', *Industrial Internet Consortium (IIC), Tech. Rep.*

Link, G. J. P., Gill, J. and Khazanchi, D. (2017) 'Measuring adoption of open source software and practices in organizations: A pilot study', *Proceedings of the Twelfth Midwest Association for Information Systems Conference. Retrieved from http://aisel. aisnet. org/mwais2017/43*.

Link, P. (2016) 'How to become a lean entrepreneur by applying lean start-up and Lean Canvas?', in *Innovation and Entrepreneurship in Education,* Emerald Group Publishing Limited, pp. 57–71.

Loucopoulos, P., Stratigaki, C., Danesh, M. H., Bravos, G., Anagnostopoulos, D. and Dimitrakopoulos, G. (2015) 'Enterprise capability modeling: concepts, method, and application', *Enterprise Systems (ES), 2015 International Conference on*, pp. 66–77.

Lueth, K. L. (2014) *Why the Internet of Things is called Internet of Things: Definition, history, disambiguation* [Online], IoT Analytics. Available at iot-analytics.com (Accessed 13 April 2019).

Lund, D., MacGillivray, C., Turner, V. and Morales, M. (2014) 'Worldwide and regional internet of things (iot) 2014-2020 forecast: A virtuous circle of proven value and demand', *International Data Corporation (IDC), Tech. Rep*, vol. 1.

Mahalakshmi, M. and Sundararajan, M. (2013) 'Traditional SDLC Vs Scrum Methodology-A Comparative Study', *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 6, pp. 192–196.

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P. and Sheth, A. P. (2018) 'Machine learning for Internet of Things data analysis: A survey', *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175.

Mahmud, S. M., Shanker, S. and Hossain, I. (2005 - 2005) 'Secure software upload in an intelligent vehicle via wireless communication links', *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* Las Vegas, NV, USA, 06.06.2005 - 08.06.2005, IEEE, pp. 588–593.

Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A. and Yaqoob, I. (2017) 'Big IoT data analytics: architecture, opportunities, and open research challenges', *IEEE Access*, vol. 5, pp. 5247–5261.

Martin, M. (2015) *Field Service Management: The changing technology landscape* [Online]. Available at http://fieldservicenews.com/ (Accessed 13.4.2019).

Mattsson, M., Grahn, H. and Mårtensson, F. (2006) 'Software architecture evaluation methods for performance, maintainability, testability, and portability', *Second International Conference on the Quality of Software Architectures*.

Matyas, K., Nemeth, T., Kovacs, K. and Glawar, R. (2017) 'A procedural approach for realizing prescriptive maintenance planning in manufacturing industries', *CIRP Annals*, vol. 66, no. 1, pp. 461–464.

McGuinness, D. L., van Harmelen, F. and others (2004) 'OWL web ontology language overview', *W3C recommendation*, vol. 10, no. 10, p. 2004.

McNamara, C. (2009) *General Guidelines for Conducting Research Interviews. Retrieved June 13, 2017* [Online].

McSweeney, A. (2014) *Structured Approach to Solution Architecture* [Online]. Available at www.slideshare.net (Accessed 13 April 2019).

Meier, J. D., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R. and Murukan, A. (2003) 'Improving web application security: threats and countermeasures', *Microsoft Corporation*, vol. 3.

Mellor, S. J., Scott, K., Uhl, A. and Weise, D. (2002) 'Model-Driven Architecture', in Goos, G., Hartmanis, J., van Leeuwen, J., Bruel, J.-M. and Bellahsene, Z. (eds) *Advances in Object-Oriented Information Systems,* Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 290–297.

Menzies, T., Chen, Z., Hihn, J. and Lum, K. (2014) 'Best Practices in Software Effort Estimation'.

Merriam, S. B. (1988) *Case study research in education: A qualitative approach*, Jossey-Bass.

Mersky, J., Wallace, D., Fortenberry, K. and others (1993) *IEEE Guide for Software Verication and Validation Plans* [Online], IEEE. Available at ieeexplore.ieee.org (Accessed 13 April 2019).

Messnarz, R., König, F. and Bachmann, V. O. (2012) 'Experiences with Trial Assessments Combining Automotive SPICE and Functional Safety Standards', *Systems, Software and Services Process Improvement.* Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 266–275.

Meyer, S., Ruppen, A. and Magerkurth, C. (2013) 'Internet of things-aware process modeling: integrating IoT devices as business process resources', *International conference on advanced information systems engineering*, pp. 84–98.

Microsoft (2014) *Edge Server disaster recovery in Lync Server 2013* [Online]. Available at docs.microsoft.com (Accessed 13 April 2019).

Microsoft (2018) *Microsoft Azure IoT Reference Architecture* [Online], Micosoft. Available at download.microsoft.com (Accessed 13 April 2019).

Microsoft Trustworthy Computing group (2013) *Resilience by design for cloud services: A structured methodology for prioritizing engineering investments* [Online]. Available at download.microsoft.com (Accessed 13 April 2019).

Miklovic, D. (2015) *What Is Asset Performance Management?* [Online]. Available at blog.lnsresearch.com (Accessed 13 April 2019).

Miller, D. (2010) *Best practices in software vendor selection* [Online]. Available at www.slideshare.net (Accessed 13 April 2019).

Miller, S. and Fredericks, M. (2003) 'The Nature of "Evidence" in Qualitative Research Methods', *International Journal of Qualitative Methods*, vol. 2, no. 1, pp. 39–51.

MITRE (2014) *Systems Engineering Guide: Verification and Validation* [Online]. Available at www.mitre.org (Accessed 13 April 2019).

Mobley, R. K. (2002) *An introduction to predictive maintenance* [Online], 2nd edn, Amsterdam, London, Butterworth-Heinemann. Available at http://www.loc.gov/catdir/description/els031/2001056670.html%20%20%20Publisher%20description.

Mozhikunnath, R. (2016) *Verification, Validation, Testing of ASIC/SOC designs – What are the differences?* [Online]. Available at anysilicon.com (Accessed 13 April 2019).

Muhonen, T., Ailisto, H. and Kess, P. (2015) 'Standards in iot, industrial internet and condition-based maintenance', *OF THINGS/////Finland*, vol. 21, p. 56.

Mukhopadhyay, T., Vicinanza, S. S. and Prietula, M. J. (1992) 'Examining the feasibility of a case-based reasoning model for software effort estimation', *MIS Quarterly*, pp. 155–171.

Murray, A., Mejias, M. and Keiller, P. (2017) 'Resilience Methods within the Software Development Cycle', *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*, pp. 62–65.

Nadimi-Shahraki, M.-H. and Bahadorpour, M. (2014) 'Cold-start problem in collaborative recommender systems: Efficient methods based on ask-to-rate technique', *Journal of computing and information technology*, vol. 22, no. 2, pp. 105–113.

Namiot, D. and Sneps-Sneppe, M. (2014) 'On micro-services architecture', *International Journal of Open Information Technologies*, vol. 2, no. 9.

Neapolitan, R. E. and Jiang, X. (2018) *Artificial intelliegence: With an introduction to machine learning*, Boca Raton, Chapman & Hall/CRC.

Newman, S. (2015) *Building microservices: designing fine-grained systems*, O'Reilly Media, Inc.

Nielsen, J. (2010) 'Website response times', *Nielsen Norman Group*, vol. 21, no. 06.

Nilsson, D. K., Larson, U. E. and Jonsson, E. (2008) ‘Creating a Secure Infrastructure for Wireless Diagnostics and Software Updates in Vehicles’, in Harrison, M. D. and Sujan, M.-A. (eds) *Computer Safety, Reliability, and Security,* Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 207–220.

Nuamah, J. and Seong, Y. (2017) ‘Human machine interface in the Internet of Things (IoT)’, *2017 12th System of Systems Engineering Conference (SoSE).* Waikoloa, HI, USA, 18.06.2017 - 21.06.2017, IEEE, pp. 1–6.

Nunamaker Jr, J. F., Chen, M. and Purdin, T. D. M. (1990) ‘Systems development in information systems research’, *Journal of management information systems*, vol. 7, no. 3, pp. 89–106.

Ochs, M., Pfahl, D., Chrobok-Diening, G. and Nothhelfer-Kolb, B. (2000) ‘A COTS acquisition process: Definition and application experience’, *ISERN Report*.

Ogu, E. and Y.A., A. (2013) ‘Basic Concepts of Expert System Shells and an efficient Model for Knowledge Acquisition’.

Ogunsina, T. (2013) *Project Risk Management* [Online]. Available at resources.intenseschool.com (Accessed 13 April 2019).

Online Trust Alliance (2018) *Internet of Things (IoT) Trust Framework* [Online]. Available at www.internetsociety.org (Accessed 13 April 2019).

Pardo-Castellote, G. (2003) ‘Omg data-distribution service: Architectural overview’, *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pp. 200–206.

Parkinson, S., Ward, P., Wilson, K. and Miller, J. (2017) ‘Cyber threats facing autonomous and connected vehicles: Future challenges’, *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2898–2915.

Patel, P. and Cassou, D. (2015) ‘Enabling high-level application development for the Internet of Things’, *Journal of Systems and Software*, vol. 103, pp. 62–84.

Patterson, D., Brown, A., Broadwell, P., Candea, G., Chen, M., Cutler, J., Enriquez, P., Fox, A., Kiciman, E., Merzbacher, M. and others (2002) *Recovery-oriented computing (ROC): Motivation, definition, techniques, and case studies,* Technical Report UCB//CSD-02-1175, UC Berkeley Computer Science.

Peffers, K., Rothenberger, M., Tuunanen, T. and Vaezi, R. (2012) ‘Design Science Research Evaluation’, in Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Peffers, K., Rothenberger, M. and Kuechler, B. (eds) *Design Science Research in Information*

Systems. Advances in Theory and Practice, Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 398–410.

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2007) 'A Design Science Research Methodology for Information Systems Research', *Journal of management information systems*, vol. 24, no. 3, pp. 45–77.

Pérez, J.\'\i., Jimeno, J. L. and Mokotoff, E. (2006) 'Another potential shortcoming of AHP', *Top*, vol. 14, no. 1, pp. 99–111.

PMI (2009) *PMI Practice Standard for Project Risk Management* [Online]. Available at www.pmi.org (Accessed 13 April 2019).

PMI (2017) *2017 Annual Report* [Online]. Available at pmi.org (Accessed 13 April 2019).

Poole, D. L. and Mackworth, A. K. (2017) *Artificial intelligence: Foundations of computational agents*, Cambridge, United Kingdom, Cambridge University Press.

Pop, P., Tsiopoulos, L., Voss, S., Slotosch, O., Ficek, C., Nyman, U. and Lopez, A. R. (2013) 'Methods and tools for reducing certification costs of mixed-criticality applications on multi-core platforms: the RECOMP approach', *Proceedings of the Conference on Design, Automation and Test in Europe, WICERT, DATE*.

Popescu, G. H. and others (2015) 'The economic value of the industrial internet of things', *Journal of Self-Governance and Management Economics*, vol. 3, no. 2, pp. 86–91.

Porter, M. E. (1998a) *Competitive advantage: Creating and sustaining superior performance : with a new introduction / Michael E. Porter*, New York, London, Free Press.

Porter, M. E. (1998b) *Competitive strategy: Techniques for analyzing industries and competitors : with a new introduction / Michael E. Porter*, New York, London, Free Press.

Porter, M. E. and Heppelmann, J. E. (2014a) 'How smart, connected products are transforming competition', *Harvard Business Review*, vol. 92, no. 11, pp. 64–88.

Porter, M. E. and Heppelmann, J. E. (2014b) 'How Smart, Connected Products Are Transforming Competition', *Harvard Business Review*, vol. 92, no. 11, pp. 64–88 [Online]. Available at https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition.

Principles of Chaos Engineering (2018) [Online]. Available at principlesofchaos.org (Accessed 13 April 2019).

Project Management Institute (1987) ‘Project management body of knowledge (PMBOK)’.

Project Management Institute (2018) *PMI Fact File* [Online]. Available at www.pmitoday-digital.com (Accessed 13 April 2019).

ProjectCodeMeter *Differences Between COCOMO, COSYSMO, REVIC, Function Points and WMFP* [Online]. Available at www.projectcodemeter.com (Accessed 13 April 2019).

Purdy, G. (2010) ‘ISO 31000: 2009—setting a new standard for risk management’, *Risk Analysis: An International Journal*, vol. 30, no. 6, pp. 881–886.

Radwin, R. G., Chourasia, A. O., Howery, R. S., Fronczak, F. J., Yen, T. Y., Subedi, Y. and Sesto, M. E. (2014) ‘Factors influencing power hand tool fastening accuracy and reaction forces’, *Human factors*, vol. 56, no. 4, pp. 657–668.

Ramachandran, U. B. (2005) *Issues in verification and validation of neural network based approaches for fault-diagnosis in autonomous systems,* Concordia University.

Rausch, A., Bartelt, C., Ternité, T. and Kuhrmann, M. (2005) ‘The V-Modell XT Applied-Model-Driven and Document-Centric Development’, *3rd World Congress for Software Quality*, pp. 131–138.

Rehman, A. and Hussain, R. (2007) ‘Software Project Management Methodologies/Frameworks Dynamics "A Comparative Approach’, *2007 International Conference on Information and Emerging Technologies*, pp. 1–5.

Reinfurt, L., Breitenbücher, U., Falkenthal, M., Leymann, F. and Riegg, A. (2016) ‘Internet of things patterns’, *Proceedings of the 21st European Conference on Pattern Languages of Programs*, p. 5.

Remtulla, S., Joseph, E. S. and Tai, E. (2014) *Method for coupling quality control data of a subassembly of components to a main product assembly* US9842303B2.

Reyes-Delgado, P. Y., Mora, M., Duran-Limon, H. A., Rodríguez-Martínez, L. C., O'Connor, R. V. and Mendoza-Gonzalez, R. (2016) ‘The strengths and weaknesses of software architecture design in the RUP, MSF, MBASE and RUP-SOA methodologies: A conceptual review’, *Computer Standards & Interfaces*, vol. 47, pp. 24–41.

Riahi, A., Challal, Y., Natalizio, E., Chtourou, Z. and Bouabdallah, A. (2013) ‘A systemic approach for IoT security’, *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pp. 351–355.

Ricci, F., Rokach, L. and Shapira, B., eds. (2015a) *Recommender Systems Handbook*, Boston, MA, Springer US.

Ricci, F., Rokach, L. and Shapira, B. (2015b) ‘Recommender Systems: Introduction and Challenges’, in Ricci, F., Rokach, L. and Shapira, B. (eds) *Recommender Systems Handbook,* Boston, MA, Springer US, pp. 1–34.

Richter, J., Streitferdt, D. and Rozova, E. (2017) ‘On the development of intelligent optical inspections’, *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC).* Las Vegas, NV, USA, 09.01.2017 - 11.01.2017, IEEE, pp. 1–6.

Robson, C. (2002) ‘Real world research. 2nd’, *Edition. Blackwell Publishing. Malden*.

Rook, P. (1986) ‘Controlling software projects’, *Software Engineering Journal*, vol. 1, no. 1, p. 7.

Rouse, M. (2017) *Definition: CMDB: Configuration Management Database* [Online]. Available at searchdatacenter.techtarget.com (Accessed 13 April 2019).

Runeson, P. and Höst, M. (2009) ‘Guidelines for conducting and reporting case study research in software engineering’, *Empirical Software Engineering*, vol. 14, no. 2, p. 131.

S. Qanbari, S. Pezeshki, R. Raisi, S. Mahdizadeh, R. Rahimzadeh, N. Behinaein, F. Mahmoudi, S. Ayoubzadeh, P. Fazlali, K. Roshani, A. Yaghini, M. Amiri, A. Farivarmoheb, A. Zamani and S. Dustdar (2016) ‘IoT Design Patterns: Computational Constructs to Design, Build and Engineer Edge Applications’, *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 277–282.

Saaty, T. L. and Saaty, T. L. F. o. d. t. (2000) *Fundamentals of decision making and priority theory with the analytic hierarchy process*, Pittsburgh, Pa., RWS Publications.

Sahebjamnia, N., Torabi, S. A. and Mansouri, S. A. (2015) ‘Integrated business continuity and disaster recovery planning: Towards organizational resilience’, *European Journal of Operational Research*, vol. 242, no. 1, pp. 261–273.

Sami, M. (2018) *The Validation and Verification Model – The V-Model* [Online]. Available at melsatar.blog (Accessed 13 April 2019).

Sasidharan, D. K. and Kumar, N. S. (2018) ‘Full Stack Development with JHipster: Build modern web applications and microservices with Spring and Angular’.

Sassikumar, V. (2018) *Can Agile be Adopted to Handle the Challenges of IIoT?*

Schlereth, C. and Skiera, B. (2012) ‘DISE: Dynamic Intelligent Survey Engine’, in Diamantopoulos, A., Fritz, W. and Hildebrandt, L. (eds) *Quantitative Marketing and Marketing Management,* Wiesbaden, Gabler Verlag, pp. 225–243.

Schwartz, E. S. and Zozaya-Gorostiza, C. (2003) ‘Investment Under Uncertainty in Information Technology: Acquisition and Development Projects’, *Management Science*, vol. 49, no. 1, pp. 57–70.

Seaman, C. B. (1999) ‘Qualitative methods in empirical studies of software engineering’, *IEEE Transactions on software engineering*, no. 4, pp. 557–572.

Serrador, P. (2012) ‘The Importance of the Planning Phase to Project Success’.

Shahzad, B., Abdullatif, A. M., Ikram, N. and Mashkoor, A. (2017) ‘Build Software or Buy: A Study on Developing Large Scale Software’, *IEEE Access*, vol. 5, pp. 24262–24274.

Shavit, M., Gryc, A. and Miucic, R. (2007) *Firmware update over the air (FOTA) for automotive industry,* SAE Technical Paper.

Shehabuddeen, N., Probert, D. and Phaal, R. (2006) ‘From theory to practice: challenges in operationalising a technology selection framework’, *Technovation*, vol. 26, no. 3, pp. 324–335.

Sigfox (2018) *Managing the risks in your IoT project* [Online]. Available at www.sigfox.com (Accessed 13 April 2019).

Simon, H. A. (1996) *The sciences of the artificial*, MIT press.

Slama, D., Durand, J. and Morrish, J. (2015) *IIC Business Strategy and Solution Lifecycle: Managing the IIoT Value Chain Transformation* [Online], THE INDUSTRIAL INTERNET CONSORTIUM JOURNAL OF INNOVATION. Available at www.iiconsortium.org (Accessed 13 April 2019).

Slama, D., Puhlmann, F., Morrish, J. and Bhatnagar, R. M. (2015) *Enterprise IoT*, O'Reilly.

Sosnin, P. (2011) ‘Question-answer shell for personal expert systems’, in *Expert Systems for Human, Materials and Automation,* InTech.

SP800, N. (30) ‘Risk Management Guide for Information Technology Systems’, *刘伟, 张玉清, 冯登国. 一个信息系统安全风险模型—Rc 模型*, vol. 2004, p. 4.

Spraque, R. H. (1980) ‘A framework for the development of decision support system’, *Management Information System Quarterly*, vol. 4, no. 4, pp. 1–26.

Standish Group (2009) *CHAOS Report* [Online], Standish Group. Available at www.projectsmart.co.uk (Accessed 13 April 2019).

Stanton, Y. (2017) *10 reasons why your IoT project will fail* [Online], OpenSensors.io. Available at diginomica.com (Accessed 13 April 2019).

Susanto, H., Almunawar, M. N. and Tuan, Y. C. (2011) 'Information security management system standards: A comparative study of the big five', *International Journal of Electrical Computer Sciences IJECSIJENS*, vol. 11, no. 5, pp. 23–29.

Swanson, L. (2001) 'Linking maintenance strategies to performance', *International Journal of Production Economics*, vol. 70, no. 3, pp. 237–244.

Szocs, G. and others (2008) 'Combination of analytic network process and Bayesian network model for multi-criteria engineering decision problems', *Engineering Management Conference, 2008. IEMC Europe 2008. IEEE International*, pp. 1–5.

Taft, J. D. (2017) *Electric Grid Resilience Electric Grid Resilience and Reliability for Grid Architecture* [Online]. Available at gridarchitecture.pnnl.gov (Accessed 13 April 2019).

Theodoridis, S. and Koutroumbas, K. (2009) *Pattern recognition*, 4th edn, Amsterdam, London, Elsevier/Academic Press.

Thomas, M., Jacques, P. H., Adams, J. R. and Kihneman-Wooten, J. (2008) 'Developing an Effective Project: Planning and Team Building Combined', *Project Management Journal*, vol. 39, no. 4, pp. 105–113.

TOGAF (2009) *The SOA Source Book* [Online], TOGAF. Available at www.opengroup.org (Accessed 13 April 2019).

Tracy, P. (2017) *The top 5 industrial IoT use cases* [Online], IBM. Available at www.ibm.com (Accessed 13 April 2019).

Treat, T. (2015) *Designed to Fail* [Online]. Available at bravenewgeek.com (Accessed 13 April 2019).

Truong, H.-L. and Dustdar, S. (2015) 'Principles for engineering IoT cloud systems', *IEEE Cloud Computing*, vol. 2, no. 2, pp. 68–76.

US General Accounting Office (GAO) (2006) *Using Structured Interviewing Techniques*.

UX for the Masses (2010) *Usability testing hints, tips and guidelines* [Online]. Available at http://www.uxforthemasses.com/ (Accessed 13 April 2019).

Vai, M. M., Khazan, R. I., Utin, D. M., O'Melia, S. R., Whelihan, D. J. and Nahill, B. R. (2016) *Secure Embedded Systems,* MIT Lincoln Laboratory Lexington United States.

van Gerven, M. and Bohte, S. (2018) *Artificial neural networks as models of neural information processing*, Frontiers Media SA.

VDA QMC (2017) *Automotive SPICE* [Online]. Available at www.automotivespice.com (Accessed 13 April 2019).

Venable, J., Pries-Heje, J. and Baskerville, R. (2016) ‘FEDS: a Framework for Evaluation in Design Science Research’, *European Journal of Information Systems*, vol. 25, no. 1, pp. 77–89.

Verrijn-Stuart, A. A. and Olle, T. W. (1994) *Methods and associated tools for the information systems life cycle: Proceedings of the IFIP WG8.1 Working Conference ... Maastricht, the Netherlands ... 1994 / eited by A. A. Verrijn-Stuart, T. William Olle*, Amsterdam, Elsevier.

Volkovs, M., Yu, G. and Poutanen, T. (2017) ‘DropoutNet: Addressing Cold Start in Recommender Systems’, *Advances in Neural Information Processing Systems*, pp. 4957–4966.

Waclawski, E. (2012) ‘How I use it: Survey monkey’, *Occupational Medicine*, vol. 62, no. 6, p. 477.

Wagner, H. M. (1975) *Principles of operations research: With applications to managerial decisions / Harvey M. Wagner*, 2nd edn, Englewood Cliffs, London, Prentice-Hall.

Wang, Y.-R. and Gibson Jr, G. E. (2010) ‘A study of preproject planning and project success using ANNs and regression models’, *Automation in Construction*, vol. 19, no. 3, pp. 341–346.

Wątróbski, J. and Jankowski, J. (2015) ‘Knowledge Management in MCDA Domain’, 13.09.2015 - 16.09.2015, IEEE, pp. 1445–1450.

Weck, O. L. de (2009) *Fundamentals of systems engineering* [Online].

Weis, D. (2018) *5 Reasons IoT Projects Fail* [Online], Association of Equipment Manufacturers. Available at www.aem.org (Accessed 13 April 2019).

Wells, H. (2012) ‘How Effective Are Project Management Methodologies? An Explorative Evaluation of Their Benefits in Practice’, *Project Management Journal*, vol. 43, no. 6, pp. 43–58.

West, D., Gilpin, M., Grant, T. and Anderson, A. (2011) ‘Water-scrum-fall is the reality of agile for most organizations today’, *Forrester Research*, vol. 26.

Westerveld, E. (2003) ‘The Project Excellence Model®: linking success criteria and critical success factors’, *International Journal of Project Management*, vol. 21, no. 6, pp. 411–418.

Whitaker, R. (2007) ‘Criticisms of the Analytic Hierarchy Process: Why they often make no sense’, *Mathematical and Computer Modelling*, vol. 46, 7-8, pp. 948–961.

White, D. and Fortune, J. (2002) ‘Current practice in project management — an empirical study’, *International Journal of Project Management*, vol. 20, no. 1, pp. 1–11.

Winne, O. and Beikirch, H. (2013) ‘Software development effort estimation for distributed embedded-systems’, *EUROCON, 2013 IEEE*, pp. 622–629.

Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. (2016) *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann.

Wold, G. H. (2006) ‘Disaster recovery planning process’, *Disaster Recovery Journal*, vol. 5, no. 1, pp. 10–15.

Wong, D. *What Is Field Service Management and Why Should You Care?* [Online]. Available at www.salesforce.com (Accessed 13 April 2019).

WSDOT (2014) ‘Intelligent Transportation Systems’, *Systems Engineering*, vol. 1, 050-04.

Wysocki, R. K. (2013) *Effective software project management*, Hoboken, N.J., Wiley.

Xu, H., Horn Nord, J., Brown, N. and Daryl Nord, G. (2002) ‘Data quality issues in implementing an ERP’, *Industrial Management & Data Systems*, vol. 102, no. 1, pp. 47–58.

Xu, T., Wendt, J. B. and Potkonjak, M. (2014) ‘Security of IoT Systems: Design Challenges and Opportunities’, *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design.* Piscataway, NJ, USA, IEEE Press, pp. 417–423.

Yang, S. (2017) ‘IoT stream processing and analytics in the fog’, *arXiv preprint arXiv:1705.05988*.

Yardley, D. (2002) *Successful IT project delivery: Learning the lessons of project failure / David Yardley*, Boston, Mass., London, Addison-Wesley.

Yatsalo, B., Didenko, V., Gritsyuk, S. and Sullivan, T. (2015) ‘Decerns: a framework for multi-criteria decision analysis’, *International Journal of Computational Intelligence Systems*, vol. 8, no. 3, pp. 467–489.

Ye, R., Chang, M., Pan, C.-S., Chiang, C. A. and Gabayno, J. L. (2018) ‘High-resolution optical inspection system for fast detection and classification of surface defects’, *International Journal of Optomechatronics*, vol. 12, no. 1, pp. 1–10.

Yeo, K. T. and Ren, Y. (2009) ‘Risk management capability maturity model for complex product systems (CoPS) projects’, *Systems Engineering*, vol. 12, no. 4, pp. 275–294.

Yin, R. K. (2003) *Case study research: Design and methods / Robert K. Yin*, 3rd edn, Thousand Oaks, Calif., London, SAGE.

Yoshida, Y. and Nakamura, H. (1989) *Remote maintenance/supervisory system and method for automated teller machines* US5253167A.

Yun, W. Y. and Nakagawa, T. (2017) ‘Comparison Between Parallel and Standby Redundant Systems’, *Reliability Modeling With Computer And Maintenance Applications*, p. 197.

Zafari, F., Gkelias, A. and Leung, K. (2017) ‘A survey of indoor localization systems and technologies’, *arXiv preprint arXiv:1709.01015*.

Zambonelli, F. (2016) ‘Towards a Discipline of IoT-Oriented Software Engineering’, *WOA*, pp. 1–7.

Zarnekow, R. and Brenner, W. (2005) ‘Distribution of cost over the application lifecycle-A multi-case study’, *ECIS 2005 Proceedings*, p. 26.

Zarnekow, R., Brenner, W. and Pilgram, U. (2006) *Integrated information management*, Springer.

Zedlitz, J., Jörke, J. and Luttenberger, N. (2012) ‘From UML to OWL 2’, in *Knowledge Technology,* Springer, pp. 154–163.

Zhang, Y., Duan, L. and Chen, J. L. (2014) ‘Event-driven soa for iot services’, *Services Computing (SCC), 2014 IEEE International Conference on*, pp. 629–636.

Zubair, M. (2014) ‘A Hybrid Approach for Reliability Analysis Based on Analytic Hierarchy Process and Bayesian Network’, *Frontiers in Energy Research*, vol. 2, p. 201.

12 Appendix

12.1 Ignite Project Dimensions Version 1.1

Based on the original Ignite IoT project methodology (Slama, Puhlmann et al., 2015), the following proposes an update specifically to the Ignite project dimensions, based on the findings of the research presented in this thesis. This is labeled as version 1.1 in the following. For each individual Ignite project dimension, the version is noted. The source for each entry labeled as version 1.0 is the original Ignite; each entry labeled version 1.1 is a contribution from this thesis.

Also, it should be noted that IIC created a new, proprietary version of the Ignite dimensions as the foundation for the IIC online project explorer introduced in section 4. The recommendation here is to work with IIC on moving these useful changes back into the open-source version of Ignite 1.1. For copyright reasons, this cannot be included here until this is resolved.

Finally, some of the newly proposed project dimensions would require a new data entry format that differs from the standard four options per dimension. Since these are special cases that potentially require significant effort on the implementation side, they are highlighted using a light-blue background in the following section.

12.1.1 Business Model and Requirements

This is a new category, proposed as part of Ignite 1.1. The initial thinking of Ignite was that this aspect was dealt with in the phase before the project setup (see discussion in section 2.1) and that the focus of Ignite 1.1 should be more on the technical/nonfunctional project dimensions. However, one key finding of the evaluation in this thesis is that the business perspective must definitely be included in the project assessment since too many nonfunctional aspects depend on it.

12.1.1.1 Functional Viewpoint

Industry Vertical

Dimension	Industry Vertical
Version	Ignite 1.1
Description	Please indicate the industry vertical that your IoT solution is situated in.
Input	Comment: This would require a new data entry format. The user should be able to select from a list of predefined industry verticals. This could be based, for example, on the IIC vertical taxonomy.

Supported Use Cases

Dimension	Supported Use Cases
Version	Ignite 1.1
Description	Please indicate which use cases the solution must support.
Input	Comment: This would require a new data entry format. For example, the user could first select from a list of predefined use cases. These could depend on the industry selected beforehand. Second, a list of free text fields would have to be supported to allow the user to add use cases not yet available in the system.

Functional Requirements

Dimension	Functional Requirements
Version	Ignite 1.1
Description	Please provide a list of functional requirements.
Input	Comment: This would require a new data entry format. For example, an open list of free text fields could be used.

Functional Complexity

Dimension	Functional Complexity
Version	Ignite 1.1
Description	Please indicate the expected level of functional complexity.
Options	1) Very low level of functional complexity, e.g., basic remote monitoring with single device class 2) Moderate level of functional complexity, e.g., advanced remote monitoring, limited number of device classes, basic analytics 3) High level of functional complexity, including multiple sensor types, complex device hierarchy, advanced data analytics, integration with multiple backend systems 4) Very high level of functional complexity, including sensor data fusion and/or control of actuators, highly sophisticated algorithms, very high level of data diversity, complex business processes

Volatility of Functional Requirements

Dimension	Volatility of Functional Requirements

Version	Ignite 1.1
Description	Please indicate how stable the functional requirements and use-case definitions are.
Options	5) Very detailed and stable requirements and use-case definitions are available. 6) High-level requirements and use-case definitions are available, which will stabilize early in the project phase. 7) High-level requirements and use-case definitions are available but are likely to constantly change and be reprioritized. 8) Requirements and use-case definitions are only vaguely defined and are likely to constantly change and be reprioritized.

Level of Productization

Dimension	Level of Productization
Version	Ignite 1.1
Description	Please indicate the expected level of productization.
Options	1) Once-off solution that must only support one particular installation and low productization requirements 2) Basic productization required. Onboarding of new customers/assets categories can require manual changes/customizations 3) Fully productized solution, onboarding of new customers done without additional customization. Small number of well-defined regional markets supported 4) Fully multi-tenant solution with very high level of productization and support for many asset classes on a global level

Global Scale

Dimension	Global Scale
Version	Ignite 1.1
Description	Please indicate the required level of global scale supported by the solution.
Options	1) Solution must support exactly one country. 2) Solution must support a small number of well-defined countries.

	3) Solution must support a large majority of countries worldwide. 4) Solution must support most countries worldwide, including those with potentially difficult political environments.

Mission Criticality

Dimension	Mission Criticality
Version	Ignite 1.1
Description	Please indicate the level of mission criticality of the IoT solution.
Options	1) Not mission critical. In case the IoT solution is not available, the operation of the assets can continue with slightly reduced efficiency. 2) Moderately mission critical. Unavailability of system has negative impact on company performance, but operations can be continued using manual workarounds. 3) Mission critical. Failure or unavailability of the solution has severe impact on business performance, potentially leading to standstill of operations. 4) Highly mission critical. Lives depend on it.

12.1.2OT: Assets and Devices

The proposal is that Ignite 1.1 follows the naming convention from IIC and includes "OT" here.

12.1.2.1 General

Number of Assets

Dimension	Number of Assets
Version	Ignite 1.0
Description	Please indicate the number of assets that will be supported by version 1.0 of your solution in this category.
Options	5) 100s 6) 10,000s 7) 100,000s 8) Millions

Value of Individual Asset

Dimension	Value of Assets
Version	Ignite 1.0

Description	Please indicate the value of an individual asset in this asset category.
Options	1) < 100€ 2) < 1,000€ 3) < 100,000€ 4) >= 100,000€

Economic Value Add of Asset

Dimension	Economic Value Add of Individual Asset/Year
Version	Ignite 1.1
Description	Please indicate the economic value add contributed by an individual asset per year.
Options	1) < 100€ 2) < 1,000€ 3) < 100,000€ 4) >= 100,000€

Asset Complexity—Integration Perspective

Dimension	Asset Complexity—Integration Perspective
Version	Ignite 1.0
Description	Please indicate the complexity of the asset from the perspective of its interfaces (only those in scope of this project).
Options	5) Zero integration 6) Simple interface semantics 7) Moderately complex interface semantics 8) Very complex interface semantics

Asset Heterogeneity—Integration Perspective

Dimension	Asset Heterogeneity—Integration Perspective
Version	Ignite 1.0
Description	Please indicate the level of heterogeneity of the asset from the integration perspective (including versions and variants).
Options	1) 0–1 interface types 2) 2–3 different interface types 3) 4–10 different interface types 4) >10 different interface types

12.1.2.2 Processing Power

Local Business Logic

Dimension	Local Business Logic
Version	Ignite 1.0
Description	Please indicate the level of business logic required to run on the asset for this solution.
Options	1) Proxy functions only 2) Basic store and forward logic to address temporary network unavailability 3) Simple business logic, e.g., rules 4) Complex business logic, e.g., autonomous management of asset

Local Event Processing

Dimension	Local Event Processing
Version	Ignite 1.0
Description	Please indicate the level of local event processing required for your solution.
Options	1) 1 event/day 2) 1 event/minute 3) 1 event/second 4) 10,000 events/second

Realtime requirements

Dimension	Real-time requirements
Version	Ignite 1.0
Description	Please indicate the level of real-time requirements for your solution.
Options	1) Daily batch synch 2) Response within seconds 3) Response within sub-second 4) Deterministic response in nanoseconds

Local Data Management

Dimension	Local Data Management
Version	Ignite 1.0

Description	Please indicate the level of local data management required for your solution.
Options	1) <1MB stored and managed locally/year 2) <10 GB/year 3) <1 TB/year 4) > 1 TB captured per asset/year

12.1.2.3 Other HW Requirements

Power Supply

Dimension	Power Supply
Version	Ignite 1.0
Description	Please indicate the level of power supply available for your solution.
Options	1) 220V wall plug 2) Automatically recharged battery (e.g., car) 3) Large battery with moderate runtime 4) Small battery with very long runtime (no auto-recharge)

Environment

Dimension	Environment
Version	Ignite 1.0
Description	Please indicate the type of environment your assets must operate in.
Options	1) Indoor 2) Rough indoor, e.g., factory 3) Outdoor, moving, e.g., car in winter 4) Extreme conditions, e.g., aircraft, space

12.1.2.4 Lifecycle Management

Projected Lifetime of Assets

Dimension	Projected lifetime of assets
Version	Ignite 1.0
Description	Please indicate the expected lifetime of your assets (or, if de-coupled, the expected lifetime of your IIoT solution).
Options	1) < 1 year

	2) ~ 5 years 3) ~ 10 years 4) > 20 years

HW Update Constraints

Dimension	HW update constraints
Version	Ignite 1.0
Description	Please indicate the level of HW update constraints for your solution.
Options	1) All assets can be accessed by specialized technician in time without large cost overview. 2) E.g., Drive Now—technician can access all cars in a city. 3) Asset configuration must be updated by end user/high cost for bringing all assets to repair shop. 4) Impossible to update HW.

Software Update Constraints

Dimension	Software update constraints
Version	Ignite 1.0
Description	Please indicate the level of SW update constraints for your solution.
Options	1) Central access management, assets always online, sufficient bandwidth, powering down for maintenance is OK 2) Normal distributed system constraints, e.g., not all at same time (parallel versions over long period of time) 3) Very difficult—long times between updates, e.g., only user initiated (e.g., user must proactively start update) 4) Impossible to update SW

12.1.3 Communication and Connectivity

Consider following IIC proposal and move this to "systemwide challenges."

12.1.3.1 Local C&C

Technology

Dimension	Technology
Description	Please indicate the complexity of local connectivity technology.

Options	1) standard bus system 2) standard wireless 3) advanced wireless, e.g., factory floor 4) very advanced wireless, e.g., specialized antenna required, special security requirements, etc.

Required Bandwidth

Dimension	Required Bandwidth
Version	Ignite 1.0
Description	Please indicate the require bandwidth for local communication.
Options	1) 100 bytes/sec 2) 100 Kbits/sec (e.g., RS 485, RS 232, CAN) 3) 1-10 Mbit/sec (e.g., video data) 4) >100 Mbit/sec (e.g., sensor data streams)

Maximum Latency

Dimension	Maximum Latency
Version	Ignite 1.0
Description	Please indicate the acceptable maximum bandwidth for local communication.
Options	1) >10 ms (e.g., RS 232) 2) 1–10 ms (e.g., WLAN, BlueTooth) 3) Microseconds (e.g., EtherCAT, Sercos) 4) Nanoseconds (e.g., ASIC, FPGA)

12.1.3.2 Remote C&C

Technology

Dimension	Technology
Version	Ignite 1.0
Description	Please indicate the complexity of remote connectivity technology.
Options	1) LAN 2) WLAN 3) global telecom network (e.g., UMTS) 4) specialized global telecom network (e.g., satellite, proprietary wireless network, etc.)

Required Bandwidth

Dimension	Required Bandwidth
Version	Ignite 1.0
Description	Please indicate the required bandwidth for remote communication.
Options	1) 100 bytes/month 2) 100–500 Kbit/sec (e.g., GPRS) 3) 0.5–10 Mbit/sec (e.g., UMTS/LT) 4) >100 Mbit/sec

Maximum Latency

Dimension	Maximum Latency
Version	Ignite 1.0
Description	Please indicate the acceptable maximum bandwidth for remote communication.
Options	1) 90 mins (LEO, e.g., OrbComm, text messages) 2) Seconds (GPRS) 3) Milliseconds (WAN) 4) Microseconds (e.g., LAN)

12.1.4 IT: Backend Services

Proposal is that Ignite 1.1 follows the naming convention from IIC and includes “IT” here.

12.1.4.1 General

Application Strategy

Dimension	Application Strategy
Version	Ignite 1.0
Description	Please describe the impact of your new solution on the existing backend application landscape.
Options	1) No new application logic 2) Embedded new business logic into already existing core apps (e.g., ERP) 3) Small, new, self-contained application 4) New major core application (i.e., significant application with ownership of key data and processes)

Business Complexity

Dimension	Business Complexity
Version	Ignite 1.0
Description	Please describe the business complexity of the new backend solutions.
Options	1) Regular updates of 3–5 remote device readings in ERP, remote monitoring only 2) More regular updates, some alarming functions 3) One new core end-to-end process, e.g., product commissioning 4) > 5 new core end-to-end processes, e.g., commissioning, customer service, customer retention, upselling, etc.

Backend Integration

Dimension	Backend Integration
Version	Ignite 1.0
Description	Please describe the requirements for integration with the existing application landscape.
Options	1) Very simple, e.g., batch, paper forms (manual integration) 2) Basic EAI 3) SOA with 2–3 complex orchestration services 4) SOA or EAI with >20 orchestrations

12.1.4.2 Data Management and Analytics

Data Volumes/Ingestion per Day

Dimension	Data Volumes/Ingestion per Day
Version	Ignite 1.0
Description	Please describe the date volumes to be ingested and processed per day.
Options	1) <10 MB/day 2) <10 GB/day 3) <1 TB/day 4) >1 TB/day

Data Variety

Dimension	Data Variety

Version	Ignite 1.0
Description	Please indicate the level of data variety to be managed in the backend.
Options	1) 3 entity types 2) <10 entity types 3) <100 entity types 4) >100 entity types

Data Variability (Schema Changes)

Dimension	Data Variability (Schema Changes)
Version	Ignite 1.0
Description	Please describe the data variability of the new backend solution.
Options	1) static data model 2) 2 per year 3) <1 per month 4) >1 per month

Analytics

Dimension	Analytics
Version	Ignite 1.0
Description	Please describe the analytics requirements of the new backend solution.
Options	1) No analytics 2) Descriptive analytics: predefined, standard reports, simple browsing 3) Complex analytics/data mining, complex data quality controls 4) Predictive analytics, streaming analytics, CEP, advanced adaptive machine learning

Data Ownership and Control

Dimension	Data Ownership and Control
Version	Ignite 1.1
Description	Please describe the requirements regarding data ownership and control.

Options	1) Data ownership is clearly defined. Use of industry standards like use of global PaaS provider for data management is acceptable. 2) Like 1 but with some added complexity in multi-stakeholder data access management. 3) Multiple data owners require complex, multi-tenant data access controls. 4) Like 3. plus data must be hosted in data centers under full control of operator, i.e., no external PaaS.

12.1.5 Systemwide Challenges

This project area was not part of Ignite 1.0. IIC has added this project area to its own, proprietary extension of the Ignite dimensions. The IIC version adds new project dimensions like systemwide availability, reliability, safety, security, and network scale. This is consistent with the findings of Evaluation I described in section 5.

Also, IIC has removed the Ignite project area "communication and connectivity" and moved the contained project dimensions to "systemwide challenges."

12.1.6 Standards and Regulatory Compliance

This area is currently not included as a top level in the IIC version of the project dimensions. However, the results of the research done for this thesis indicate that this aspect is important and should be included on the top level.

12.1.6.1 Regulatory Requirements

Region Specific

Dimension	Region specific
Version	Ignite 1.0
Description	Please indicate the level of region-specific regulatory requirements your project deals with.
Options	5) No strict requirements 6) Low requirements 7) Medium requirements 8) Complex requirements

Industry Specific

Dimension	Industry Specific

Version	Ignite 1.1 (update)
Description	Please indicate the level of industry-specific regulatory requirements your project deals with. This should also include any mandatory company internal standards or standards imposed by technology transfer partners.
Options	1) No strict requirements 2) Low requirements 3) Medium requirements 4) Complex requirements

Technology Specific

Dimension	Technology Specific
Version	Ignite 1.0
Description	Please indicate the level of technology-specific regulatory requirements your project must deal with.
Options	1) No strict requirements 2) Low requirements 3) Medium requirements 4) Complex requirements

12.1.6.2 Standards

Technical Standards

Dimension	Technical Standards
Version	Ignite 1.0
Description	Please indicate the level of technical standards your project must deal with.
Options	1) No standards as prerequisites defined 2) Some prerequisites for M2M hardware, bus systems, application APIs, communication protocols 3) Relatively high requirements for M2M hardware, bus systems, application APIs, communication protocols 4) Very strict prerequisites for M2M hardware, bus systems, application APIs, communication protocols

Functional/Industry Standards

Dimension	Functional/Industry Standards

Version	Ignite 1.0
Description	Please indicate the level of functional/industry standards your project must deal with.
Options	1) No standards as prerequisites defined 2) Some prerequisites for M2M hardware, bus systems, application APIs, communication protocols 3) Relatively high requirements for M2M hardware, bus systems, application APIs, communication protocols 4) Very strict prerequisites for M2M hardware, bus systems, application APIs, communication protocols

New Standards

Dimension	New Standards
Version	Ignite 1.1
Description	Please indicate to what extent creation and establishment of new standards are a key requirement of this project.
Options	1) Creating or setting new standards is not a requirement. 2) Creating or setting new standards is a low-priority “nice to have.” 3) A new technical or industry standard must be created and adopted/approved by a leading standards body. 4) Like 3 but also adding as a requirement that the new standard must be widely used by other companies.

12.1.7 Project Environment

12.1.7.1 Project Environment

Timeline

Dimension	Timeline
Version	Ignite 1.0
Description	Please describe the expected timeline of your project.
Options	1) Plenty of time 2) Design to budget 3) Aggressive timeline 4) Death-march project

Budget

Dimension	Budget
Version	Ignite 1.0
Description	Please describe the budget situation of your project.
Options	1) Generous budget 2) Realistic budget 3) Optimistic budget 4) Completely unrealistic and insufficient budget

Functional Skills and Experience

Dimension	Functional Skills and Experience
Version	Ignite 1.0
Description	Please describe the functional skills and experience of the team.
Options	1) Existing team, has done similar project before 2) Like 1 but geographically distributed 3) Completely new team, individual team members have little functional experience in relevant area 4) Like 3 plus distributed

Technical Skills and Experience

Dimension	Technical Skills and Experience
Version	Ignite 1.0
Description	Please describe the technical skills and experience of the team.
Options	1) Existing team, has done similar project before 2) Like 1 but geographically distributed 3) Completely new team, individual team members have little technical experience in relevant area 4) Like 3 plus distributed

Team Stability and Availability

Dimension	Team Stability and Availability
Version	Ignite 1.1
Description	Please describe the expected stability and availability of the project team.
Options	1) All team members will be assigned full time and for the entire duration of the project.

	2) Core team will be assigned full time and for the entire project duration. 3) A significant amount of fluctuation in the team is to be expected. 4) Very high level of fluctuation expected. Also, many specialists will only have very limited availability even while they are assigned to the project.

IT Organization

Dimension	IT Organization
Version	Ignite 1.1
Description	Please indicate the type of IT organization that will be responsible for building and operating the IIoT solution.
Options	1) Small team, single location 2) Small team, multiple locations 3) Multi-tier IT organization with different responsibilities for network, database, server ops, etc. in single location 4) Multi-tier IT organization with different responsibilities for network, database, server ops, etc. in different locations

Field-Service Organization

Dimension	Field-Service Organization
Version	Ignite 1.1
Description	Please indicate the type of field-service organization that will be responsible for using the IIoT solution to provide field services to asset users.
Options	1) Well-established, existing field-service organization. IoT solution will provide some value-added services but not have huge impact on established processes and organization. 2) Small, newly established field-service organization. 3) Existing field-service organization will leverage new IoT capabilities for significant re-engineering of processes and organization. 4) New field-service organization will leverage new IoT capabilities to build processes and service organization from scratch.

Stakeholder Complexity

Dimension	Stakeholder Complexity
Version	Ignite 1.1
Description	Please describe the expected project stakeholder complexity.

Options	5) All internal key stakeholders have fully bought into the project and have similar goals. Few external dependencies. 6) Most internal stakeholders have bought into the project and have reasonably aligned goals. Some external dependencies. 7) Complex internal and external stakeholder network with partially colliding interests. 8) Highly complex internal and external stakeholder network with colliding interests.