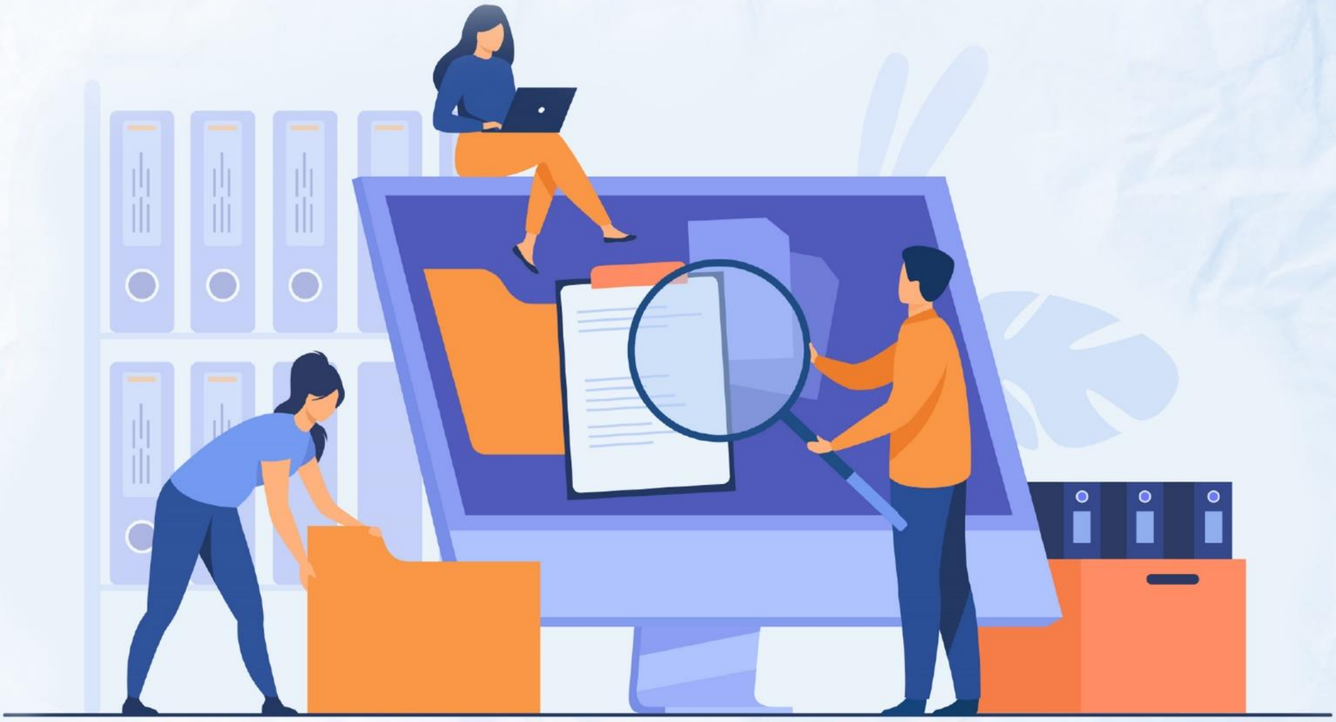
Requirements Tracing
in Agile Environments



TU Delft

2021

Requirements Tracing in Agile Environments

A management-oriented approach:
Developing a practical framework and roadmap to resolve the tensions of integrating requirements traceability in Agile environments

by

Mihai Iliuhin

to obtain the degree of Master of Science
at the Delft University of Technology
Master Thesis

Student number:	5181682	
Project duration:	01.03.2021 – 27.09.2021	
Thesis committee:	Prof. Dr. Ir. Marijn Janssen	TU Delft, 1st supervisor
	Prof. Dr. Cees van Beers	TU Delft, 2nd supervisor
	Master of Science Ricardo Matheus	TU Delft, supervisor

This research is confidential and cannot be made public until the 27th of September, 2021

Acknowledgments

This master thesis represents the final requirement for fulfilling the degree of Master of Science in Management of Technology, Department of Technology, Policy and Management at TU Delft. At the end of the master program, I feel deeply marked by fulfillment, which comes from the personal and professional satisfaction of the completed studies, which I found profoundly interesting. Therefore, I would like to express my sincere gratitude to all the people and organizations which have supported me along this path.

First and foremost, I am grateful to all my professors and colleagues for the fruitful discussions, knowledge, and shared experiences that helped me develop many soft and technical skills of high importance for my future career.

I would like to express special gratitude to my supervisors Prof. Dr. Ir. Marijn Janssen, Ricardo Matheus and Prof. Dr. Cees van Beers who supported me alongside my Master Thesis by offering valuable research guidelines and constructive feedback. Besides the important recommendations and advice on the topic, I appreciated the offered autonomy and responsibility in defining the aspects and focus for the research.

Next to that, I am thankful to all organizations and persons who participated in the conducted survey. A truly thank you for their time, valuable insights, and shared experience in regard to Integration of Requirements Traceability in their organizations.

Last but not least, I would like to thank my parents and girlfriend for their unconditional love, support and encouragement throughout this master program.

Table of Contents

1 INTRODUCTION 1

1.1 BACKGROUND 1

1.2 RESEARCH CONTEXT 2

1.3 PROBLEM STATEMENT 3

1.4 RESEARCH OBJECTIVE 4

1.5 RESEARCH SCOPE 5

1.6 RESEARCH QUESTION 6

1.7 RESEARCH OVERVIEW 7

2 LITERATURE STUDY 9

2.1 PART I - RT AND AGILE 9

2.1.1 RT introduction in Agile projects 9

2.1.2 Types of Requirements Traceability 10

2.1.3 RT methods and techniques 12

2.1.4 RT Goals and Strategies 16

2.1.5 RT - Challenges 19

2.1.6 RT Integration in Agile Environments - Challenges 20

2.1.7 Literature Stakeholder Analysis 26

2.1.8 Role of Humans in Traceability Loop 30

2.2 PART II – BPMA AND RT 31

2.2.1 Business Process Management 31

2.2.2 Business Process Management Lifecycle Challenges 34

2.2.3 Business Process Modeling Languages 34

3 THEORETICAL FRAMEWORK 44

3.1 TENSION POINTS - CORRELATION 45

3.2 RESEARCH PERSPECTIVES – COMBINING PART I AND II 46

3.3 ANSWERING SRQ I 49

4 RESEARCH APPROACH 50

4.1 DATA COLLECTION – PART I (RTI FRAMEWORK) 50

4.1.1 Face-to-Face Survey 50

4.1.2 Sampling Approach 53

4.1.3 Data Analysis BWM 54

4.2 DATA COLLECTION – PART II (BPMO ROADMAP) 58

4.2.1 Structured interview 58

4.2.2 Sampling approach 58

4.2.3 Data Analysis 58

5 ABC - CASE STUDY 59

5.1 RTI CHALLENGES AND FRAMEWORK – PART I 60
5.1.1 Product Development Methodology *60*
5.1.2 Goal Definition *62*
5.1.3 Understanding Stakeholder Needs *62*
5.1.4 Traceability Characteristics *65*
5.1.5 BWM Weights Analysis *66*
5.1.6 ABC Challenges – Summary *69*
5.1.7 RTI Framework *70*
5.1.8 Answering SRQ II *73*
5.2 ROADMAP DESIGN – PART II 74
5.2.1 RT in Agile - BPMo Design *74*
5.2.2 Answering SRQ III *83*
5.3 ROADMAP AND FRAMEWORK EVALUATION – SRQ IV 84

6 RESEARCH LIMITATIONS 86

6.1 LIMITATIONS – RTI CHALLENGES AND FRAMEWORK 86
6.2 LIMITATIONS - BPMN ROADMAP 89

7 CONCLUSION 90

7.1 ACADEMIC REFLECTION 91
7.2 MANAGERIAL IMPLICATIONS 92
7.3 FUTURE RESEARCH 93
7.4 REFLECTION ON MANAGEMENT OF TECHNOLOGY 93

BIBLIOGRAPHY 95

APPENDIX 100

APPENDIX A - SELECTION CRITERIA 100
APPENDIX B – BPMO LANGUAGES 103
APPENDIX C – ETHICS REVIEW FOR HUMAN RESEARCH 107
APPENDIX D – INTERVIEW AND SURVEY RESULTS (PART I) 109
APPENDIX E – BWM (Ξ(KSI) VALUES AND WEIGHTS) 114
APPENDIX F – INTERVIEW AND SURVEY (PART I) 117
APPENDIX G – INTERVIEW AND RESULTS (PART II) 120

List of Abbreviations

AE	Agile Environment
BPMa	Business Process Management
BPMo	Business Process Modeling
BPMN	Business Process Model and Notation
BWM	Best-Worst Method
KPI	Key Performance Indicator
MCDM	Multi-Criteria Decision-Making
MRQ	Main Research Question
MVP	Minimum Viable Product
OMG	Object Management Group
RE	Requirements Engineering
RT	Requirements Traceability
RTI	Requirements Traceability Integration
SME	Small and Medium Enterprise
SRQ	Sub-Research Question
TDD	Test Driven Development
TTM	Time To Market

List of Figures

FIGURE 1: DOT-COM BUBBLE AND AGILE WAY OF WORKING 1
FIGURE 2: RESEARCH OBJECTIVE 4
FIGURE 3: RESEARCH METHODOLOGY 7
FIGURE 4 RESEARCH STRATEGY 8
FIGURE 5: AGILE AND RT TENSION POINTS 10
FIGURE 6: RT TYPES AND TRACING 11
FIGURE 7: PROJECT GOALS 17
FIGURE 8: STAKEHOLDER USE CASES FOR TRACEABILITY INTERESTS 28
FIGURE 9: POWER AND INTEREST OF STAKEHOLDERS BASED ON DIFFERENT GOALS 29
FIGURE 10: TRADITIONAL BUSINESS PROCESS MANAGEMENT FOCUS 32
FIGURE 11: BPMA LIFECYCLE DESIGN AND RUNTIME PHASES 33
FIGURE 12: RT INTEGRATION WITH AGILE TENSION POINTS 45
FIGURE 13: RT INTEGRATION WITH AGILE METHODOLOGIES RESEARCH CONTRIBUTION 48
FIGURE 14: INTERVIEW- SURVEY DESIGN 52
FIGURE 15: STRATIFIED SAMPLING 53
FIGURE 16: COMPLEXITY ANALYSIS RALPH STACY MATRIX 60
FIGURE 17: RT TOOLS AT ABC AND EXTERNAL COMPANIES 66
FIGURE 18: RTI CHALLENGES WEIGHTS BASED ON ALL INTERVIEW SAMPLES 67
FIGURE 19: MANAGEMENT PERSPECTIVE ON RTI CHALLENGES 68
FIGURE 20: SOFTWARE DEVELOPERS PERSPECTIVE ON RTI CHALLENGES 69
FIGURE 21: RTI FRAMEWORK 70
FIGURE 22: ABC BPMN ROADMAP 82
FIGURE 23: EXTERNAL DEVELOPERS BWM WEIGHTS 88
FIGURE 24: ABC DEVELOPERS BWM WEIGHTS 88

List of Tables

Table 1: Research Sub-Questions 6
Table 2: RE in Traditional Methods and Agile 13
Table 3: Requirements Traceability Techniques 15
Table 4: Research Goals for Requirements Traceability 18
Table 5: Traceability Challenges 20
Table 6: Challenges of integrating RT in AE 23
Table 7: RT Solutions for Agile Environments 25
Table 8: Stakeholders Roles Definition 27
Table 9: Business Process Modeling Languages Categories 35
Table 10: BPMo Language Selection Criteria 36
Table 11: BPMN Flow Objects 40
Table 12: BPMN Artifacts Description 41
Table 13: BPMN Connection Symbols 42
Table 14: Environment Complexity and Project Development Methodologies 61
Table 15: Interviewees knowledge of main thesis concepts 63
Table 16: Managers and Software Developers Perspectives on RT and Agile Challenges 64
Table 17:Appendix - Table of contents 100
Table 18: Search words refinement process 101
Table 19: Search tools and their relevance 101
Table 20: Research articles selection 102
Table 21: Flowchart Components 104
Table 22: CMMN Main Components 106
Table 23: Management Interview Questions Results Summary 110
Table 24: Developers Interview Questions Results Summary 113
Table 25: BWM KSI Values and Weights 116
Table 26: BPMN and RTI Framework Interview Questions 120
Table 27: BPMN and RTI Framework Interview Results 124

Executive Summary

Scholars have studied for many years the challenges of integrating requirements traceability in Agile environments. The scholars' developed tools, frameworks, and models take a rather technical perspective on the problem. The existing literature focuses primarily on mechanisms that provide accurate and relevant information retrieval from the requirements database or analyze various techniques for creating valid links between requirements. Although the technical approach provides solutions to several scoped challenges, the scholars did not take into account the business goals, stakeholders' needs, and business processes. In practice, the company resources, the development approach, and knowledge workers' interests play a critical role in requirements traceability. As a result, multiple tradeoffs are required to develop a balanced and stable environment. In particular, the approaches are not in sync with agile development approaches, whereas the first requires documentation, the latter is focused on working software. Therefore, taking a business and management perspective on requirements traceability integration in Agile methodologies could bring new meaningful insights into solving existing challenges.

To solve existing challenges, the research was divided into two parts: 1) *requirements traceability framework* and 2) *business process model*. The first part of the study focuses on requirements traceability methods, techniques, challenges, stakeholders' needs, business processes, and other solutions discussed in the literature. The insights from the literature study are used to develop the theoretical *requirements traceability framework* that guide practitioners with the requirements traceability integration in Agile environments. The second part of the study is an extension of the framework and focuses on the topic of business process management and business process modeling. In this part is developed a roadmap detailing business processes complementary to the traceability framework and is more practical.

The research goal is to build a requirements traceability framework and a business process model that serve as a roadmap for practitioners on how to achieve requirements traceability in Agile environments.

To achieve this goal, a detailed research methodology was further developed to ensure the research's reliability and validity. The data collection for each part was done through a case study that was performed on a medium-sized company specializing in photonic integrated circuits technology named ABC. The case study is scoped to the processes related mainly to software development in the R&D and T&M departments. The results showed that stakeholders with different roles such as managers and developers have different interests in requirement traceability. As a result, there are constant tradeoffs made about which tools, methods, and processes to use for requirements traceability.

The first part of the case study resulted in the development of the Requirements Traceability Integration (RTI) framework. The framework defines several steps the practitioners should follow to strategize for traceability in the projects. Some of the framework steps are: *identification of project methodology, business and project requirements traceability goals, selecting requirements traceability techniques, stakeholders' needs, analyzing existing business processes,* and others. The RTI framework emphasizes what tradeoffs should be made in each step of the roadmap. Moreover, the process is iterative and requires adjustments if the projects are long-term based.

The second part of the research represents the development of the business process model that has been developed by considering the RT challenges at ABC. The model closely follows the steps of the RTI framework and represents an extension that provides a clear view of the tasks and processes to be followed at ABC to integrate the requirements traceability in their environment. The language (BPMN) for modeling the ABC processes was selected based on research findings from the literature study. The business process roadmap decomposes the RTI framework to individual tasks distributed among practitioners' roles involved in a project. Therefore, the business process roadmap not only emphasizes the steps to be taken to make tradeoffs between what is desirable and feasible, but also showcases what stakeholder roles should be taken into consideration for an effective requirements traceability approach.

In the last phase of the research, the framework and roadmap are evaluated. The practitioners mentioned that the RTI framework provides a good perspective on the project planning and makes an individual consider different approaches and complexities associated with requirements traceability, which will help make the necessary tradeoffs. Interviewees mentioned that if the workflow is implemented, it can solve the challenges of *documentation*, *regulatory compliance*, *improving product development lifecycle*, and *measuring project status*. Nevertheless, for the processes of *frequent requirement change control*, the practitioners mentioned that more work is needed to fit different needs within multiple projects at ABC.

The adopted managerial perspective takes a broader view on the problem of requirements traceability. It will allow the scholars to direct the research in a way that meaningful insights into stakeholders' needs and strategic traceability planning is captured. The study provides a foundation for topics such as: how business resources can influence requirements traceability, what is the level of stakeholders interests in traceability based on their role, how to make requirements traceability tradeoffs based on existing organization constraints, and how business process modeling can help improve traceability in Agile environments. Furthermore, the thesis brings meaningful data in terms of stakeholders' interests/needs in requirements traceability by using the best-worst method. The approach provides future researchers with a foundation for investigating the processes that will facilitate the workflow negotiation within a dynamic organization that aims to adopt requirements traceability.

To implement the requirements traceability at a company, it is important to start with analyzing the business goals, project goals, and available resources. The second step would be to follow the business process modeling lifecycle and investigate the “as-is” processes before considering new workflow strategies. Performing a retrospective on existing processes will already showcase several bottlenecks and possible challenges. The third step is to understand the knowledge workers' needs and interests towards traceability and assess the employees' knowledge of the concepts. The last step would be to follow the RTI framework and the business process roadmap and identify the tradeoffs and relevant processes for the company.

For future research, it is suggested to explore how to balance stakeholders’ needs, power, and interests in requirements traceability. Furthermore, to increase the validity and generalizability of future research, it is recommended to perform several case studies in organizations that activate in different industries and conduct a comparative analysis between those industries. Finally, additional research and development should be done towards the generalizability of different tasks described in the business process roadmap and to develop a set of best practices and templates for more complex steps.

1 Introduction

1.1 Background

In 1990 there was a shift in manufacturing towards the customer, which demanded products that were more reliable, innovative, cheap, and had a greater variety. As a result, the qualifying standards continuously increased, alongside the demand from emerging economies and higher living standards (*Griffiths et al., 2000*). Furthermore, rapid technological advancement in the software domain, combined with worldwide legislation amendments for an open market, has encouraged international trade, leading to rapid economic growth in many countries. Such changes have increased the competitiveness between organizations to a high level. Moreover, the investors' interest in dot-com companies grew, leading to further competition and pressure on software companies to reduce the time to market for a product (see *Figure 1*).

To address this problem, scholars and practitioners proposed redesigning the traditional methods of the development cycle by including a more iterative approach. The approach evolved from the Waterfall model as a necessity to meet the market demand *(Cohen et al., 2003) (Batool et al., 2013)*. Shortly after, the Agile Manifesto was introduced in 2001 by a collaborative work of seventeen software engineers with the end goal to provide a more flexible methodology for project development *(Beck et al., 2001)*. The Manifesto stated that Agile is all about "*working software over comprehensive documentation*," "*individuals and interaction over process and tool*," "*responding to change over following planning*," and others *(Cohen et al., 2003, pg.7)*.

However, the new Agile methodologies did not go hand-in-hand with the requirements traceability (RT) methods used by traditional project development processes. Traceability of requirements has been identified to positively influence the software products being developed, increase customer satisfaction, and reduce information asymmetry. Therefore, RT has qualified as an index of software quality (*Qasaimeh & Abran, 2013*) (*Curcio et al., 2018*). Nevertheless, the organizations that wanted to become more ambidextrous through Agile faced the burdensome task of maintaining high product quality while abandoning the traditional way of requirements engineering (RE) and RT.

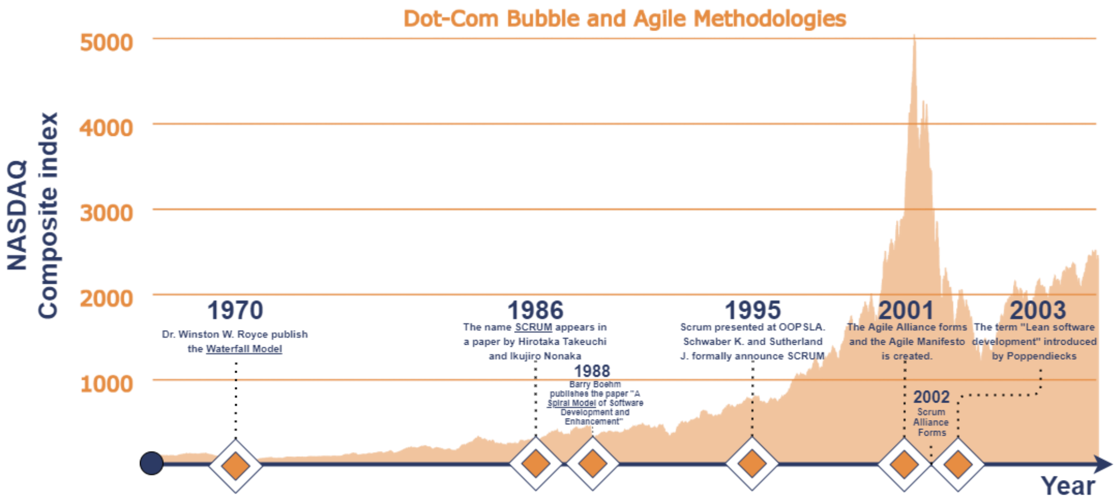


Figure 1: Dot-Com bubble and Agile way of working

After decades of research, a definite answer could not be given on how to combine the Agile way of working with RT methods most effectively. Scholars have proposed various frameworks, tools, and ideas. However, most studies tackle only a specific tension or a small group of factors that could make the integration easier. However, such analysis usually takes a computer science perspective on the problem. As a result, the studies put less emphasis on the broader picture of how RT challenges and processes are interconnected with business goals, project constraints, and stakeholders' needs.

With all the available knowledge on the topic, a different approach can be investigated by taking a management perspective on the problem of RT integration in Agile environments. The new perspective focuses on business process management (BPMa) techniques and Process Modelling Business (BPMo) languages in order to develop an integration roadmap so that practitioners can make the necessary tradeoffs easier between different interconnected RT tensions. The knowledge contained in this research will allow organizations to guide themselves through the process of integrating Agile methodologies and RT techniques based on organization-specific circumstances, rather than following a strict particular model, framework, or tool. Nonetheless, before developing such a roadmap, it is critical to understand the base concept of RT, the Agile way of working, and business process management.

1.2 Research Context

Agile Methodologies

The Agile methods are incremental and iterative and usually do not perform any extensive RE and RT on the overall architecture at the beginning of project development. While, in traditional approaches, the RE and RT represent critical initial steps to make a consistent and well-developed final product (*J. C. Lee & McCrickard, 2007*). Overall, the scholars who have studied the Agile methodologies in large organizations have emphasized that correctly remodeling the process of RE and RT represents one of the main challenges for a successful transitioning and implementation of Agile (*Uludag et al., 2018*) *(Batool et al., 2013) (Bjarnason et al., 2011).*

Agile methods are known in multiple cases to reduce problems related to dynamic environments and requirements that change fast. The methods such as Scrum, Kanban, XP, also introduce new challenges such as the complexity of striking a good balance between agility and regulatory compliance necessary for certain industries such as automotive, medical or telecommunications. Therefore, the desire to comply with industry standards while maintaining a high degree of flexibility raises the problem of *how to maintain a reliable process for RT in Agile environments*?

RT Challenges and Shortcomings

Many articles that provide a well-defined model about traceability are based on the traditional software development processes (*Gomes & Pettersson, 2007 pg.7*) (*Cleland-Huang, 2012 pg. 11-12*). On the other hand, the Agile requirements traceability models always emphasize the existing and possible implementation challenges in the industry through case studies. Furthermore, the authors usually provide a more abstract view of how traceability can be achieved via automation, ontology (semantics), machine learning, and other techniques

(*Furtado & Zisman, 2016*) (*Bjarnason et al., 2011*). The software development methods such as Scrum, Kanban, XP, Lean, Crystal, and Streamline (***hereafter called Agile methodologies***) cannot be supported by existing traditional models of RT, because the traditional require more bureaucracy, different management roles, and administrative work, which does not align with the philosophy of Agile Manifesto.

The existing articles about the traceability approaches in Agile, do not take a broader view on analyzing the tensions and challenges from a business process perspective. A more comprehensive view on the topic can help prioritize the existing shortcomings and challenges based on the industry, stakeholders, product, organization size, experience, and other metrics. Therefore, it can be argued that a practitioner would be better equipped to develop a good process of managing RT tensions in an Agile environment if a well-defined and flexible roadmap would be available as a starting point.

Business Goals and Stakeholders Needs

Numerous stakeholders involved in the daily business operations add an additional layer of complexity to the integration of Agile methodologies and RT. For an effective RT process, the stakeholders in different departments should agree on a set of communication channels and tools for system design, implementation, and validation. However, if to make the predicament that a change is required in the project development process, then the bureaucracy, lack of motivation, and personal interests of different stakeholders can become obstacles for quick adjustments and ambidexterity. As a result, this does not align with the core Agile philosophy mentioned in the above sub-chapters. For example, in a project within an organization operating in an industry such as telecommunications, there are usually involved: electronics, telecommunications and software engineers, project managers, product owners, auditors, and other regulatory inspectors. Every stakeholder speaks a different language within the project, while they all must be involved in the short iterations and feedback loops required by Agile methodologies. Furthermore, the product owner and manager should trace and understand the validation and testing of project requirements performed by stakeholders from different layers.

In a paper by *Alaa & Samir (2014)*, it is mentioned that the stakeholder's interest varies depending on the role they have within the project. The "*multi-faceted roadmap*" framework described in the paper emphasizes that the product owner is least interested in the traceability data and focuses more on quality verifications. On the other hand, Scrum masters, for example, will allocate more time in tracing requirements, analyzing design, test cases, and system performance.

The implication of different stakeholders and their divergent levels of interests regarding traceability and product development provides a good template for dividing the stakeholders into separate layers that follow unique processes. Such division in combination with a BPMo language should provide a good view of how the stakeholders should interact to serve everybody's best interest and lead to a more successful rate to meet the customer needs.

1.3 Problem Statement

The complexity to achieve a symbiosis between the Agile practices and RT depends highly on the project complexity, available resources, technology, and others. However, a study on traceability in Agile software projects conducted by *Cleland-Huang et al. (2014, pg. 5)* has shown that "[1] *Traceability research must be driven by the needs of its stakeholders, who ultimately adopt tracing solutions*…[2] *there is little prior work that examines the specific needs of the stakeholder in the traceability process and, as a result, academic researchers are left making assumptions about industry needs*…". Furthermore according to *Rempel et al. (2013, pg.1),* "*…the overall quality and mismatch of analyzed traceability suggests that an upfront-defined traceability strategy is indeed required.*

Scholars and practitioners have researched the problem for more than two decades. Still, a definite answer on how to integrate RT in Agile environments with a high efficacy could not be fully provided. The factors affecting the integration of RT in Agile environments are complex and interconnected. The practitioners constantly face difficulties on how to balance between RT and Agile way of working (see *Table 6*). Especially organizations that try to move from a more R&D phase toward complex, large-scale manufacturing processes. The studied literature emphasizes a couple of solutions to several isolated tension factors (see *Table 7*). However, very few studies tackle how the challenges are interconnected and how RT is affected by the stakeholder needs. Furthermore, the practitioners are missing roadmaps on how to integrate RT business processes in Agile effectively and how to make the correct tradeoffs in the long-run. Therefore, organizations have to analyze multiple sources and materials to understand how to design and plan such processes.

1.4 Research Objective

As mentioned in *chapters 1.2* and *1.3*, the existing literature on RT implementation in Agile mostly focuses on analyzing existing issues in the organizations, and the studies are narrowed to more technical aspects such as what are the best methodologies to establish traceability links, how to automate the traceability processes, and what tools should be used. However, this thesis proposes to take a more business perspective (management) approach on the problem of RT in Agile environments with the help of BPMa techniques and BPMo tools. Therefore, the research objective of this study can be summarized as follows:

***Develop a theoretical roadmap and business process model to enable requirements traceability in Agile environments*.**

As RT originated from the traditional software development methods, several tensions appear as described in the chapters above. Such tensions are believed to be mitigated by developing a BPMo roadmap for implementing traceability in Agile environments, as shown in *Figure 2* below.

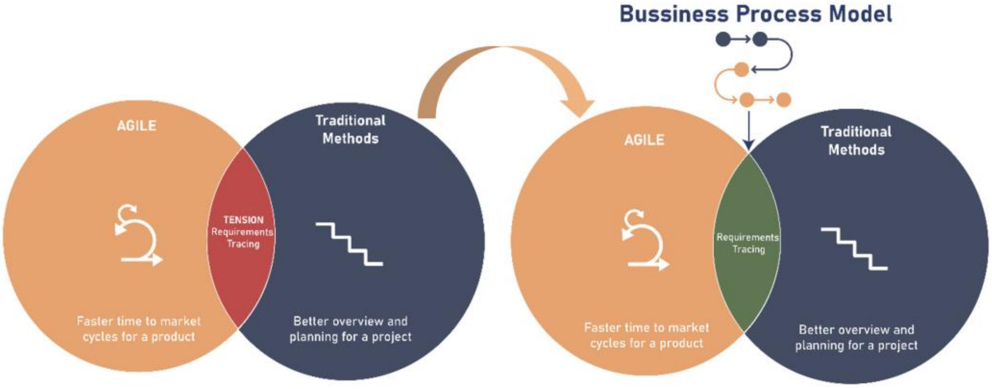


Figure 2: Research Objective

1.5 Research Scope

Research Environment

The study was conducted in a medium-sized company named ABC that activates in the telecommunication industry, developing fiber-optic communication systems based on Photonic Integrated Circuit (PIC) technology. The company has encountered rapid growth of more than 150 employees in a short period of three years due to its technology and delivering promising results in the area of 5G and PIC. The research performed at ABC is about studying the tensions of RT methodologies when these are integrated into Agile environments and how these tensions can be resolved through the context of an RT framework and business process modeling. As the company rapidly moves to large-scale manufacturing and towards releasing the high-end 5G transceiver modules to the market, many customers will look for quality assurance certifications in the areas of manufacturing, hardware, and software development. However, in a study by *Espinoza & Garbajosa (2011)*, it was mentioned that RT practices with Agile methodologies could not be performed in a traditional requirement management fashion as required by SWEBOK or ISO/IEC 12207:2008 standards. This can be explained by the completely opposing approaches required by Agile and traditional methods for documentation and collaboration, a tension frequently mentioned in the paragraphs above. Furthermore, *Espinoza & Garbajosa (2011)* highlighted that RT challenges in Agile could lead to problems in processes such as requirements change management, project estimation, and impact analysis.

Therefore, the scope of this study is limited to conducting a case study analysis at ABC R&D and Test and Measurement (T&M) departments. The T&M and R&D departments at ABC work in a dynamic/ad-hoc environment where the business processes start to slowly move from pure R&D processes to more production processes due to the plans of mass product manufacturing. The data for the study will be collected from ABC and from other organizations' R&D departments similar to the ABC size, for research generalization (transferability). The results from the ABC study case, combined with the samples collected from a more extensive population set, are believed to provide a better understanding about the business processes that could facilitate the integration of Agile methodologies and RT.

Furthermore, in current research, the concept of ***Agile environment(s)*** will often be mentioned. The Agile environment(s) is used to describe dynamic, flexible, ad-hoc environments where requirements can often change, and it is hard to define stable business processes to support the product development.

RT and Business Process Model (BPMo)

A BPMo can be applied to any process within the ABC business environment. However, in the current study, the emphasis is on developing a BPMo that describes the process of reducing existing traceability tensions when practiced with Agile methodologies. The model will focus on the key stakeholders of the environment as well as the independent variables such as requirements elicitation, project methodologies, stakeholder needs, tools, and other factors that influence the design of BPMo and RT integration.

1.6 Research Question

The main research question (MRQ) for the thesis is as follows:

Main Research Question (MRQ)
How can business process modeling enable requirements traceability in Agile environments of small and medium-sized enterprises?

The RQ is an explanatory question that aims to analyze the complexity of integrating RT methods in an agile environment. The question will be answered through carefully designed data collection methods as described in *Chapter 4*.

To more precisely define the most essential mediating and moderating variables, a set of sub-questions that be investigated are the following:

Nr	Sub-Research Questions (SRQ)	Aim
I	***What are the factors that influence requirements traceability in Agile environments?***	Gather data and get an overview on the factors such as *artifact traceability*, *granularity*, *requirements change*, *stakeholders collaboration*, *documentation*.
II	***How does a framework aiming to enable requirements traceability and reduce tension in the context of Agile environments look like?***	Select the key factors/criteria that reflect the tension and challenges when integrating RT in Agile environments.
III	***How will a business process model diagram look like in the context of the developed framework?***	Develop a BPMo so that the key factors of the framework are translated into a business process.
IV	***Is the developed framework and business process model evaluated as detailed and complete?***	To evaluate the developed framework and business process model and better understand the practitioners' needs and the direction of future research.

Table 1: Research Sub-Questions

1.7 Research Overview

This chapter provides an overview of the strategy followed for the current thesis to answer the MRQ. The layout and flow of the research are depicted in the graph shown below (see *Figure 3*). For more details on the research approach, the reader should look at *Chapter 4*.

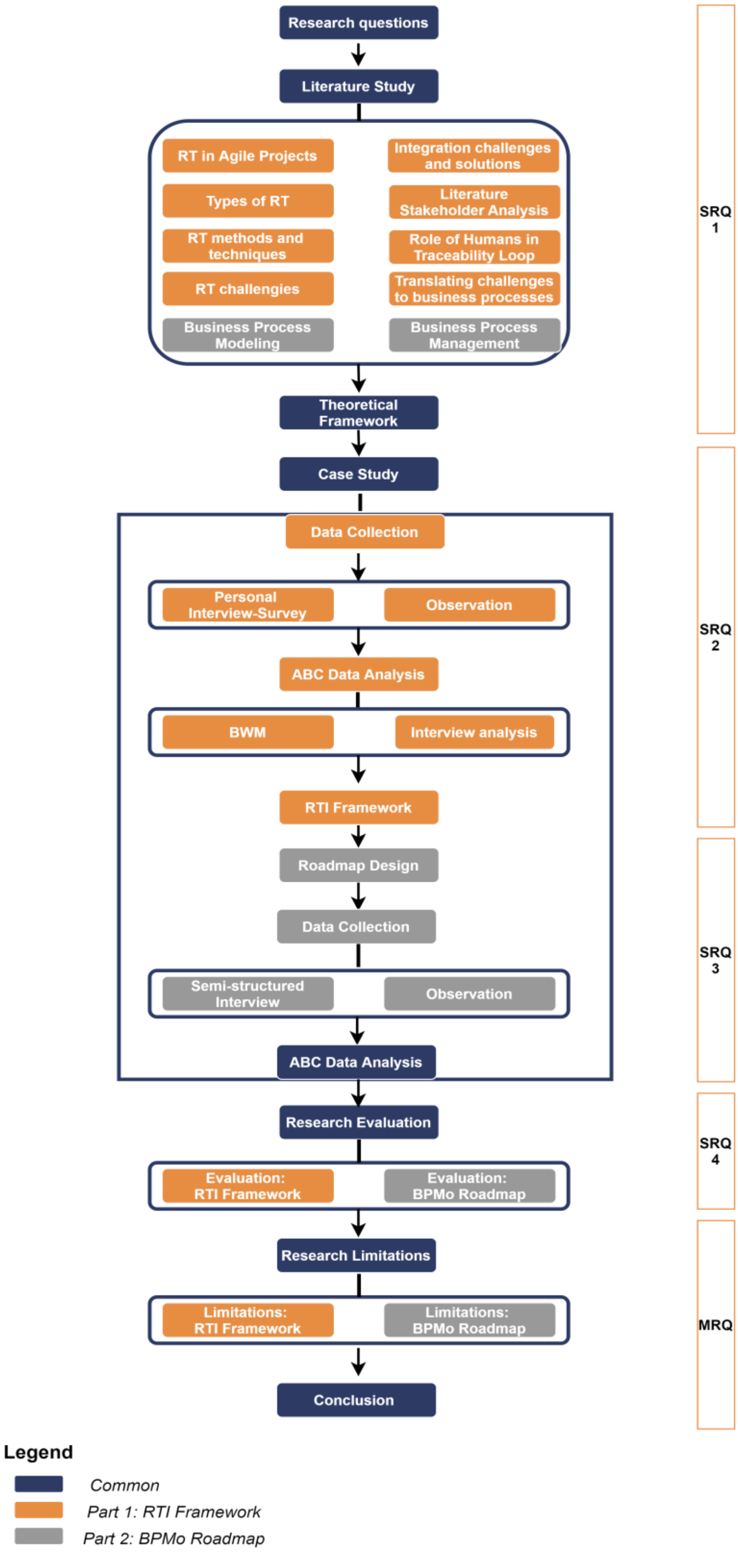


Figure 3: Research Methodology

Research Strategy

This research is divided into two key parts:

- The design of the RTI framework
- The development of BPMo Roadmap.

The separation can be observed throughout the research such as in: literature study, case study, and research evaluation (see *Figure 3*). The sub-processes described in the BPMo roadmap are directly correlated with the steps described in the RTI Framework. The two designs should always be viewed as complementary and not as separate components to answer the MRQ. The reason to divide the research into two parts is related to the different approaches taken by the framework and the roadmap. The RTI Framework will focus on requirements traceability, existing challenges, methods, techniques, stakeholders' needs, business processes, and other methods discussed in the literature study. The framework represents the theoretical perspective on the problem. On the other hand, the BPMo roadmap, is developed based on the topics such as business process management and business process modeling. It represents a design-oriented approach to the thesis. The roadmap is designed to provide a more detailed overview for practitioners on the topic of RT integration at ABC. The BPMo roadmap is the more in-depth and practical extension of the RTI Framework.

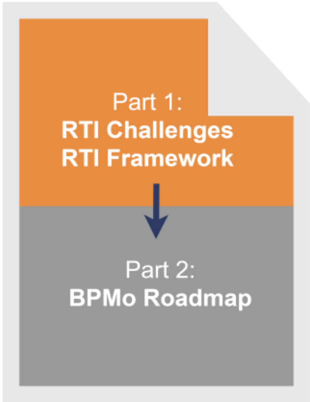


Figure 4 Research Strategy

The case study is represented by the data collection, data analysis, and development of the RTI framework, which will serve as the basis for designing the BPMo roadmap for ABC company. Further in the research, there will be one more round of data collection and analysis to evaluate the quality of the created BPMo roadmap.

To answer the research questions mentioned in *Chapter 1.6*, several strategies have been taken into consideration:

SRQ-I: The first sub-research question requires desk research to establish the main tensions related to RT in an Agile environment. The output of the desk research will serve as the basis for the theoretical framework and the development of the RTI framework.

SRQ-II/III: In order to answer the second and the third sub-questions, a study case approach will be conducted at the ABC company. This will give the possibility to investigate the phenomena of requirement traceability in agile environments in its real-life context. Furthermore, the case study will help analyze in-depth the traceability processes used by the ABC employees and their challenges (see *Chapter 4*).

SRQ-IV: The last sub-question will be answered by conducting another round of interviews with the several selected ABC employees to receive an evaluation of the RTI framework and BPMo roadmap. This evaluation will be used as guidance to understand the design limitations and also to better frame the scope for future research.

The validity threats were analyzed separately for the RTI framework and BPMo roadmap in the research limitations - *Chapter 6*.

2 Literature Study

This chapter describes two essential topics for the current research. Firstly, the chapter will describe RT types, methods, goals, and the challenges related to RT integration in Agile environments. Secondly, the focus will be on business process management, processes categorization, and modeling languages. The research on the topics mentioned above should provide a good foundation for the case study.

2.1 Part I - RT and Agile

The term "***traceability***" is referred to in this study as the ability to correlate artifacts created during the process of software development so that the end system can be described from different levels of abstractions and perspectives. Furthermore, the definition of *Gotel and Finkelstein (1994, pg.94)* is also taken into consideration which defines traceability as "*The ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases*".

RT represents a sub-category of RE and has been identified to have a critical role in positively influencing the software products that are being developed, therefore qualifying as an index of software quality (*Qasaimeh & Abran, 2013*) (*Curcio et al., 2018*). The standards such as ISO 9001:2008 (*ISO 2015, para. 8.5.2*), ISO 26262 in automotive (*Pitchford 2019, pg. 280*), medical IEC 62304 (*Regan et al. 2013, ch. 3*), have enforced traceability as a hard requirement for organizations to receive the industry quality compliance certification. Therefore, RT is necessary, in specific industries, to release the product to the market.

Besides the factors of RT mentioned above, the process also brings several other benefits and challenges, which are described in the below chapters via a more in-depth introduction to the topics such as:

- Current applicability of traceability in Agile projects.
- What types of RT exist.
- What traceability methods are being used.
- What is the future of RT integration in Agile.
- How BPMo tools can play a role in further shaping the future of Agile methodologies.
- What are the key stakeholders involved in RT.

2.1.1 RT introduction in Agile projects

Traceability in projects directly influences the development activities such as validation, impact analysis, reuse. In addition, it tackles some of the Agile implementation's main obstacles, such as lack of documentation, imprecise cost estimation of projects, constantly changing customer requirements, and others *(Curcio et al., 2018)* (*Espinoza & Garbajosa, 2011*). The RT also helps with tracking defects that appear during the software development life cycle, such as overlap between components, positively impacts the communication procedures in the team, and finally helps with transparency on the status of the project in different development phases (*Qasaimeh & Abran, 2013*) (*Spanoudakis & Zisman, 2005*).

The benefits of RT can be easily denoted in many studies. However, developing a model to integrate RT in Agile is a complex task. Therefore, it is essential to understand when the RT should be used in projects and when is the case where extensive RT can lead to project failure. To better understand the tensions and the challenges introduced by RT in Agile environments, a closer look should be taken in *Figure 5*: Agile and RT below.

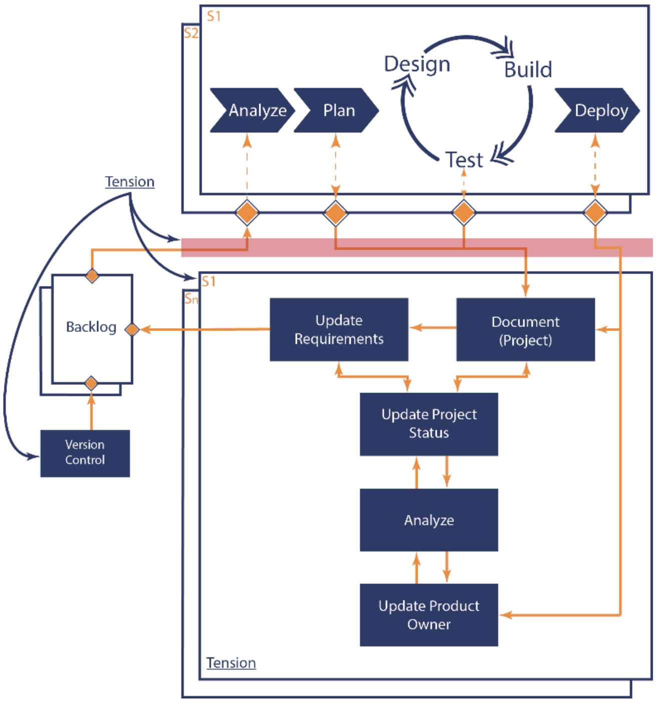


Figure 5: Agile and RT Tension Points

On top of the existing Agile iteration processes, the team should follow a set of additional operations such as "*Update Requirements*", "*Update Project Status,*" and update product "*Backlog,*" which come from a more traditional software development pattern. Such a fundamental restructuring of the product delivery process raises a couple of questions such as: *How to manage such integration*? *In what scenarios should this approach be implemented*? *Which stakeholders should be involved*?

A study conducted on several organizations by *Jakobsson (2009)* emphasized that the knowledge workers are reluctant to add additional routines to the Agile process and have decreased motivation and creativity if the administrative burden is increased. Similar findings are mentioned in a study by *Hoang Duc (2015)*, and the "*Human Factor*" is also listed as a challenge of traceability in a report by *Antoniol et al. (2006)*. Therefore, an organization should always carefully assess if the RT is required and if there are enough resources and motivation to support it.

Empirical studies state that Agile methodologies can increase project success if managed correctly (*Serrador & Pinto, 2015*). Suppose a project does not have a hard requirement on traceability, and there is no concrete answer to why RT should be included. In such scenarios, a better approach would be to follow the standard Agile methodologies, alongside the principles of *Lean* development as described by *Reis, (2011*). The cost of creating and maintaining traceability links should be lower than the expected benefit.

However, if the organization or a particular project is subjected to rigorous control from regulators to comply with a certain industry quality standard (e.g., ISO9001, ISO 26262, IEC 62304), whilst the desire to maintain agility represents a top priority. A more in-depth analysis of the trade-offs, challenges, RT Types, tools, and existing solutions should be performed. For such cases, a more detailed analysis is performed in *Chapter 5* below.

2.1.2 Types of Requirements Traceability

From the traditional methods of RE, requirements are divided into several traceability types. These RT types define the technique for how the requirements will be tested and how the customer expectations will be met. In software engineering, this represents the ability to associate software artifacts. According to *Gotel & Finkelstein (1994)*, traceability can be divided into two types: "Pre-RS traceability" and "Post-RS traceability" (see *Figure 6*). The Pre-RS traceability is concerned with the requirement's life prior to the inclusion in the requirement production (backlog). Post-RS traceability is about the requirement's life after inclusion in the backlog and how the requirement is monitored when the deployment and development work starts. The authors have emphasized that a clear distinction between pre/post RS traceability directions should be made, as the problems related to RT in practice were partly because of the lack of the distinction mentioned above.

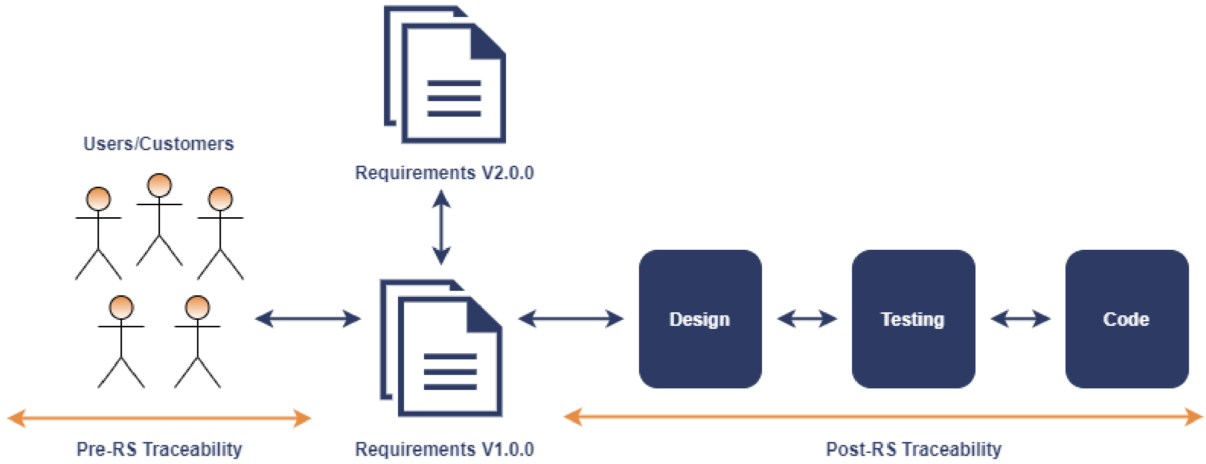


Figure 6: RT Types and Tracing

Source: (Li et al., 2002)

According to *Cleland-Huang (2012, pg. 20)*, Post-RS/Pre-RS traceability can be further divided into four types: *Forward Tracing*, *Backward Tracing*, *Horizontal Tracing* and *Vertical Tracing*.

Forward Tracing is commonly used in the software engineering context for monitoring the tracing path that usually starts from the user stories and continues to requirements backlog, design, testing, and code. *Cleland-Huang (2012, pg. 19)* defines that tracing can be primary or reverse trace link direction, and it is dependent on the specifications of the participating traces.

Backward Tracing refers to tracing links from code and design back to the requirements backlog and initial user stories. As in the case with forwarding tracing, the tracing can be primary or reverse trace link direction, and it is dependent on the specifications of the participating traces.

Horizontal Tracing refers to the decision to trace between different versions of requirements and versions of the design, testing, or code. The horizontal tracing can be both forward and backward and is at the same level of abstraction. An example of such traceability can be as follows: (i) The product owner creates a different version of a requirement that can be traced. (ii) The traces between the system design version and code where system performance is monitored. (iii) The traces of all the requirements created based on a particular user story.

Vertical Tracing – Represents the tracing of artifacts at different levels of abstraction and can also employ both *backward* and *forward tracing*. Such traceability is used to accommodate the end-to-end traceability or life cycle-wide. An example of vertical traceability can be tracing different versions of the requirements to the same version of the code or system design.

Tracing Methods

The artifact tracing requires tools and knowledge from workers on how to operate and effectively manage the requirements within the project. In literature, the scholars have defined three types of RT: ***Manual, Semi-Automated,*** **and** ***Automated*** (*Cleland-Huang, 2012, pg. 10*).

Manual Tracing – Such tracing involves a knowledge worker who initializes and monitors the process of requirement traceability. This involves actions such as traceability creation and maintenance, usually through simple actions such as edit, delete, drag, drop, and assign, found in the requirements management tools.

Automated Tracing – The method refers to the process of establishing automated techniques for requirements traceability. This involves automatic updates to the actions specified in *the manual tracing* based on changes done by knowledge workers during the process of design, testing, or user story updates. An automated *vertical* or *horizontal* tracing is very complex to achieve because it involves semantics. Currently, there are several studies on using advanced statistical algorithms to establish such traceability.

Semi-Automated Tracing – Such traceability is the most common and involves combining automated techniques and processes with the knowledge workers' interaction. An example of such a combination can be an automatic notification to the product owner when a test case has successfully finished and ticked from the product backlog a customer requirement.

2.1.3 RT methods and techniques

Agile and RT come from two opposing methodologies. Therefore, the tools and process of defining artifacts, monitoring requirements, and maintaining good status control differs. The chapters below describe the process of determining the artifacts, establishing the RT technique, and defining granularity in Agile and RT projects.

2.1.3.1 Defining Artifacts

In Agile methodologies, the "User Stories" represent the main source for developing the system requirements. The user stories are more open to interpretation when compared to the traditional methods. Therefore, it is harder to translate to more unambiguous requirements at the beginning of the project. However, this is compensated by the iterative approach introduced by Agile methodologies. In addition, the stories in Agile are used for developing user tests, which further contribute to the process of test-driven development (TDD). During the TDD process, the complete design and system requirements are further captured and linked directly to the production code *(Martin & Melnik, 2008*) (*Espinoza & Garbajosa, 2011*).

As Agile methodologies embrace constant change to requirements and its core philosophies are about rushing into software development as a priority, the RE slightly differs from traditional methods. In a paper by *Alaa & Samir (2014)*, the authors highlighted the RE process for Agile development as depicted in *Table 2* below.

RE Phase	RT - Agile Methodologies
Elicitation & Analysis	Early customer and other stakeholder involvement; Frequent communication; Defining and collaborating on user stories.
Specifications	Verifying/reviewing the quality of user stories; Developing the product backlog; Defining the sprints from backlog;
Elaboration	Quick prototyping based on the backlog key tasks; Frequent team communication; Code review sessions.
Validation	Unit and integration Testing; Review sessions/user acceptance tests; Evolutionary prototyping;
Management	(Varies based on Agile methodology). E.g., Scrum: Planning Sprints; Organizing product backlog; prioritizing epic tasks; dividing epic tasks into subtasks.

Table 2: RE in Traditional Methods and Agile

Researchers such as *Carniel & Pegoraro (2017, pg.1-2), Furtado & Zisman (2016, pg.68), Alaa & Samir (2014, pg.1)* and others, have stated that user stories are the main source to derive the first set of artifacts (ideas, requirements), that builds the product backlog and provides the first input parameters to their traceability model. The backlog is then used to develop and support the TDD process and can also be considered an artifact. According to *Carniel & Pegoraro (2017)*, the user stories must be documented in a structured way to achieve traceability in Agile environments. However, documentation represents one of the key challenges of integrating RT in Agile environments (see *Chapter 2.1.5*). Furthermore, documentation introduces new routines that some practitioners believe in decreasing creativity and that little attention is given by their colleagues who might be interested in the project design (*Jakobsson, 2009*). In the paper by *Ratanotayanon et al. (2009)*, where a more practical approach to requirements traceability is described, the authors have used the mechanism of version control in software to monitor test cases and reflect the validity of requirements. This is also supported in the paper by *Espinoza & Garbajosa (2011, pg. 63)*, which mentions that "*If continuous integration and rigorous testing are practiced, FIT-style requirements should be consistent with the produced code, through traceability support*." However, to achieve such a well-established process of monitoring, testing and continuous integration a set of RT tensions should be addressed, which vary per organization (see *Chapter 2.1.4)*.

2.1.3.2 RT Techniques

The topic of RT methods, techniques, and tools has been studied for a long time, especially in the context of traditional methodologies such as Waterfall. RT is important to maintain consistency throughout the system development lifecycle. The techniques used for RT provide a way to trace information of various forms and differ based on the number of links. These can trace the number of interconnections and the flexibility to adjust the trace links based on requirements changes.

According to *Espinoza & Garbajosa (2011)*, several traceability techniques help ensure RT during the project development, as described in the table below. It should be mentioned that the traceability methods were not only investigated from the *Espinoza & Garbajosa (2011)* paper, but rather a snowballing technique was used to define the relevant sources and information to the below-mentioned traceability methods.

Traceability Method	Description
RT Matrices	RT matrices (RTM) are the most used method to maintain trace links in software-intensive and safety-critical systems. The RTMs can vary based on the organization's needs. Nevertheless, they always contain four key components: requirement id (first column), description (module name, author name), test case id, test case description, and the status of the requirement. These usually represent a worksheet where the center is the requirement and all the possible test case scenarios with their current state and if some tests have passed or failed. Therefore, the RTMs are helpful in visualizing requirements coverage by using the test cases and communicating easier with the customer about the project status. As a result, this ensures higher quality for the developed system.
Integration Documents	The methodology provides traceability between different life cycle phases of the project or components via documents. In a paper by *Lefering (1993),* it is described that to obtain such traceability, two components are necessary. First, the "*Source Document*" contains a set of requirements of a particular element or phase and the "*Target Document,*" which requirements are linked to the source. The documents are not aware of each other's existence, but they communicate with the second component, the "*Integrator*". The integrator dictates the rules for requirements (via a "*transformation-table*") and allows user interaction to control the trace links. Such systems are rarely designed from scratch by companies focused on developing a specific product, but rather the organizations procure a finalized "*Integration Documents*" product combined with RTMs or any other tracing methodology.
Graph-based Representation	The tracing methodology is also known in the literature as "*Model-based requirements management*" and is aimed to create a visual map of requirements and how these are interconnected *(Holder et al., 2017)*. The visualization in literature research varies from UML diagrams to abstract nodes, representing visually linked requirements. The graph-based approach aims to combine all traceability management activities such as recording, identification, and information retrieval maintenance in a single framework. Such methodologies are believed to improve the validation and development of the product *(Schwarz et al., 2010)*.
Hypertext	The methodology is part of the "*semi-automated*" requirement traceability recovery process. There is a consistent problem in requirement management: if the traceability links change over time, the validity is also impacted. As a result, the data about how the system quality has changed, and the initial links are lost. Hypertext in requirements tracing uses various statistical analyses to discover terms and associate them based on their meaning. For example, in a study by *Maletic et al., (2003)* it is analyzed how a technique such as "Latent semantic indexing" can help with the retrieval of valuable information about requirements. Hypertext has become an interesting topic for scholars and practitioners because of the advances in the recurrent neural network (RNN) and natural language processing (NLP) algorithms.

Key-phrase dependencies	The key-phrase dependencies is another methodology for traceability described in a paper by *Jackson (1991)*. The technique is not based on machine learning but instead uses specific syntax and keywords pre-defined in a table. The program then looks for the keyphrases and parses them correspondingly to data in the table to retrieve the linked requirements. In the paper by *Jackson (1991)*, the author argues that the software developers can use in their code the same syntax. And this will also allow linking directly to the software code of a system. The advantages of working and developing such a system are that it does not require a large amount of data to link and define meaning from the requirements. The disadvantage is that it depends on human interaction and relies on a process to use a specific key-phrase convention that can be easily ignored if multiple cross-functional teams work on the system.

Table 3: Requirements Traceability Techniques

An interesting observation is that few authors from the studied papers have included in their design one of the RT mentioned above techniques when describing their models for traceability. Thus, when it comes to RT in Agile, *scholars, and practitioners are divided into two group*s. One of those groups relies on the studies performed on the topic of RT in the past, but carefully acknowledge its limitations in Agile context (e.g., *(Furtado & Zisman, 2016), (Espinoza & Garbajosa, 2011), (Qasaimeh & Abran, 2013), (Carniel & Pegoraro, 2017)*). Whilst other authors put very little emphasis on the past studies done on RT and classifies them as outdated for the Agile methodologies because of fundamentally opposing characteristics of the methods and roles (e.g., *(Alaa & Samir, 2014), (Ratanotayanon et al., 2009)*). However, the scholars that put little emphasis on the past studies, also took a more practical approach to develop a model for RT.

2.1.3.3 Traceability Links

In the phases of requirements elicitation and specification, the product owner and the architect, or system analyst, should know what requirements are important to trace and which are the traceability links that should be formed to deliver the end product in time with the desired customer quality. In practice, the traceability links depend on how safety-critical the system should be and what certifications are required for regulatory compliance. According to *Alaa & Samir (2014), Espinoza & Garbajosa (2011), Furtado & Zisman (2016), Qasaimeh & Abran (2013), Hoang Duc (2015), Ratanotayanon et al. (2009),* and other scholars, there are several methods which are concluded of importance when it comes to establishing traceability links within a project:

- ***Stakeholder – Requirements:*** During the requirement elicitation phase, it's important to store the information about who is the owner of a specific requirement. This will allow project managers to create feedback loops and validate the requirement with the right stakeholder. Moreover, it helps to create a map of stakeholders and assign responsibilities accordingly.

- ***Requirement – Requirement*:** Being able to trace different requirements can help better link the user stories together and understand the overall scope of the system. Such trace links can also be between different versions of the same requirement. However, in Agile environments, the requirements change quickly, and frequent adjustments lead to overhead to maintain such trace links.

- ***Requirement – System Design**:* After the requirements validation, the system design represents the next phase of product development. In traditional methodologies, failure to comply with key requirements can lead to a wrong product being delivered at the end of the project timeline. In the case of Agile development, this can represent a wasted iteration cycle. Furthermore, failing to comply with some essential requirements at this development phase can lead to incorrect budget estimations, milestone deadlines, etc. Therefore, maintaining a trace link and a history of a system design change to requirements can drastically increase the product quality.

- ***Requirement – Test Cases**:* TDD is the most common technique practitioners use to validate the system code based on the requirements. Being able to trace from requirement to test-case and backward should increase the system quality and ensure a bug-free software code (given that TDD best practices were used in the project)

- ***Stakeholder evaluation – Version:*** In the paper by *Alaa & Samir (2014),* the author identified that practitioners also value the linkage between the stakeholders' (user) feedback and the acceptance criteria. The information collected for these trace links is user satisfaction, flaws in business logic (if any), problems with the project development, and others.

- The projects vary in size and complexity based on the requirements such as safety, reliability, accuracy, performance, and other metrics. This complexity also requires other trace links to make sure the product is performing based on the specifications. Therefore, practitioners use a lot more trace links in the industry. Some of these links are described in the thesis by *Hoang Duc (2015)* and can be resumed as follows*:*
    - *User Story – System Design; User Story – User Story; Test Cases – Source Code; Test Case – System Design; System Design – System Design; Source Code – Source Code.*

Even though the trace links are well defined and argued in different papers, the most complex seem to be the "*Stakeholders – Requirements* " and " *Requirements – Test Cases*". As described by *Jarzębowicz & Weichbroth (2021)*, the iterative nature of agile methodologies creates extra complexity in constantly measuring and tracking the non-functional requirements in the system.

2.1.4 RT Goals and Strategies

When it comes to RT goals, there are usually four perspectives to be explored. There are RT business goals, project goals, stakeholders' goals, and research goals. This sub-chapter will analyze business, project, and research goals perspectives. The stakeholders' interests are described in more detail in *Chapter 2.1.7* below.

In the paper by *Rempel et al. (2013, pg.1)*, the authors argue that many researchers believe traceability rarely happens in an ad-hoc manner and is usually explicitly planned upfront. But their findings clearly emphasize that it is quite the opposite; many practitioners follow no explicit traceability strategies. However, the process of determining all suitable trace paths is technically complex and involves multiple stakeholders' interests. As a result, this causes the discrepancy between the existing traceability, reported traceability, and the actual applied development process. Nevertheless, the authors derived a traceability strategy model, as shown in *Figure 7* below (for more details, see *Rempel et al. (2013, pg.1)*).

The framework represents a detailed view of the decision-making process that practitioners should take before traceability goals are defined. However, the perspectives given by the framework are only on the project level; no frame of reference is provided by the authors when it comes to business goals.

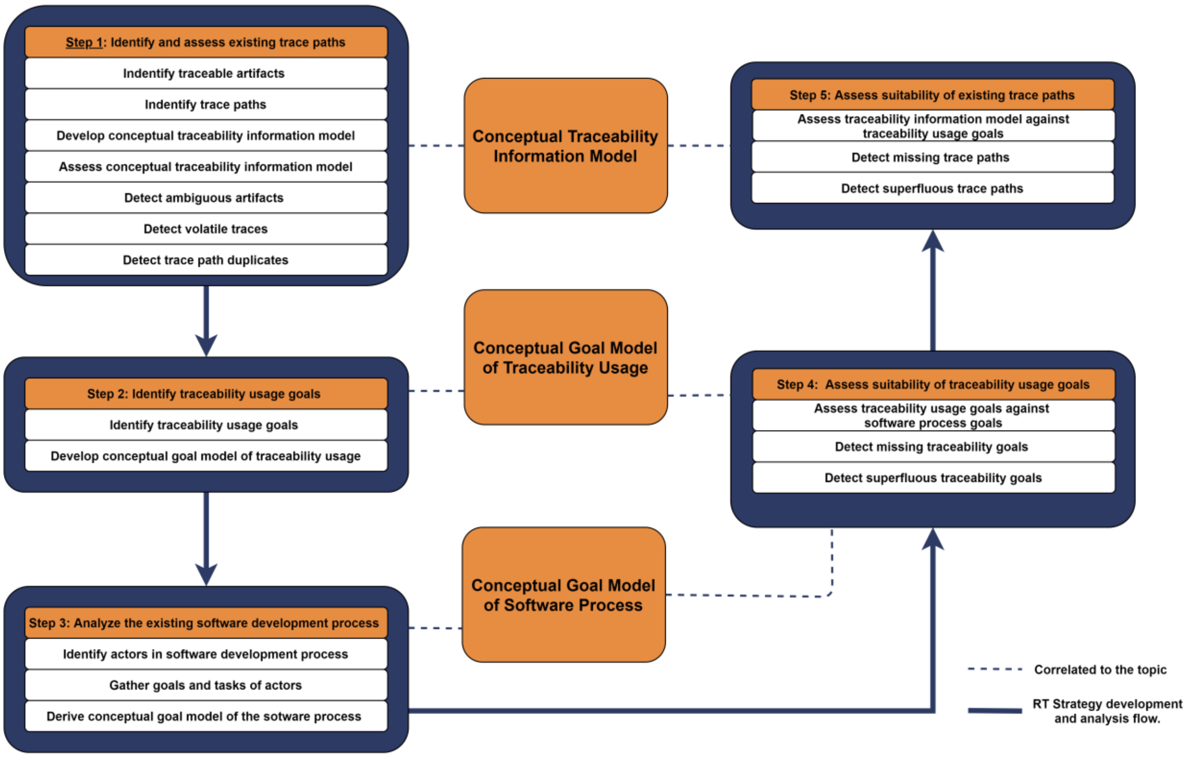


Figure 7: Project Goals

Source: Rempel et al. (2013, pg.1)

To build further on the idea, the business perspective on the topic can be derived from the paper by *Cleland-Huang et al. (2014, pg. 3)*. The author introduces "*a goal-orientated perspective*" on the challenges experienced by researchers when investigating the topic of RT. These challenges are categorized into a set of goals. Each goal is defining the focus and the challenges that are analyzed under a particular purpose, as described in the table below.

Traceability Goal	Research Perspective (Cleland-Huang et al. (2014))	Business Perspective (Adjusted)
Purposed	The research aims to identify and better understand what traceability is needed and how the tasks are connected to RT and support the stakeholder needs.	The knowledge workers' roles and their workflow, needs, and interests play a critical role when selecting a specific traceability method or tool to be used within the context of the organization.

Cost-Effective	The studies are aimed to analyze what is the ROI for using a specific traceability technique. The researchers should look at different traceability decisions and product life-cycle to understand the cost and benefit.	The management estimates its budget for a particular project and defines what tools and methods for traceability should be implemented. The team looking at a cost-effective solution should also develop strength and weakness matrices for different methods.
Configurable	The research for configurable systems focuses on dynamically identifying traceability links that could potentially be more relevant for stakeholders' needs whilst being at the same time semantically reach and relevant.	From the business perspective, a configurable system is the one that can be adjusted first to the existing business traceability processes in different departments and projects. However, the system can also be tailored to individual stakeholder needs.
Trusted	The articles that are based on the *Trusted* goal focus on investigating and developing models and tools that bring confidence in their information retrieval techniques. The research focuses on systems where the stakeholders have complete confidence in traceability.	A trusted system is viewed from two perspectives. First, it is looked at from the maturity level and if long-term support is available. The second perspective is the accuracy provided by the tool/method on traceability. Every organization has a different threshold for trust.
Scalable	The studies focusing on the *Scalable* goals should investigate how the artifacts can be traced on different granularity levels across all the organization boundaries.	The business goal matches the research goal in this context. However, scalable requirements tools and processes vary based on the needs of each organization.
Portable	The goal of portability can be attained in the research by investigating standards, policies, and possible exchanging formats that can accurately integrate information across different projects so that the system can be reused and merge across the organization.	In business, a solution that attains the goal to be portable is usually the one capable of integrating with other business processes without too much overhead. Furthermore, a portable solution can the one which processes align with business planning and satisfy the management thresholds for KPIs
Valued	The research aims to develop a traceability method that involves the needs and workflows of each stakeholder. The method should support the technical and business area of the project.	A valued system does not create an additional administrative burden while supporting the existing business processes flawlessly.

Table 4: Research Goals for Requirements Traceability

Combining the papers of *Cleland-Huang et al. (2014, pg. 3)* and *Rempel et al. (2013, pg.1)* with a business perspective interpretation provides a new layer of abstraction that better integrates the business goals with the project goals. For example, now it is possible to go through the five steps of *Figure 7*, while taking the perspective for the system to be *Cost-Effective.* In such a case, the processes such as "*Identify traceability usage goals*" and "*Identify trace paths*" will be answered with the goal of *Cost-Effectives* in mind.

2.1.5 RT - Challenges

Very often, in practice, the traceability is conducted in an ad-hoc manner. The vague traceability processes cause the RT benefits to go unnoticed or not fully realized. To improve the traceability processes and to create real industrial-strength solutions, a group of researchers has met in 2005 at the International Conference on Software Engineering and launched the "*Center of Excellence for Software Traceability*" to investigate *"the Grand Challenges of Traceability*" *(Giuliano Antoniol et al., 2017)*. The initiative to create such a group led to a wide range of topics and challenges discussed by scholars and is further elaborated in this sub-chapter.

In 2017 the scholars had announced a new meeting on "*Grand Challenges of Traceability*". During the event, the participants agreed on two critical areas that could improve the research quality on RT and help tackle the traceability challenges in the upcoming years. The first area was about the lack of datasets on traceability. According to *Giuliano Antoniol et al. (2017*), such scarcity is caused by the organizations that impose restrictions on scholars to share the datasets or even the investigation results. The second challenge is the lack of researched tracing techniques adopted in industrial practice, followed by the absence of datasets from those real-world implementations. Nevertheless, besides the difficulty of investigating the RT topic, the researchers are confronted with multiple technical, strategic, and business-related problems and knowledge gaps raised by practitioners when integrating RT practices in their organizations. Some of the existing and future planned research alongside challenges are briefly summarized in *Table 5* below:

Research Topic	Challenge	Description
Trace Strategizing		
Evaluation of Traceability	**Measuring Traceability Utility**	The scholars are still investigating what an effective traceability utility value is and how this should be measured. The discussions on the topic varied from measuring the recall and precision of requirements from the tools to measuring the value from tasks and how much more effective and quick they are accomplished and understood by the project players.
Traceability Queries and Strategies for the Requirements Engineering Domain	**Develop TIM models to improve RT navigation and analysis.**	The practitioners are using a multitude of tools for documenting, maintaining, querying different types of requirements. This represents an obstacle because the information is scattered, and it needs to be consolidated from all the data sources to have meaningful insights. An analysis performed on the queries can help create information traceability models (TIM) for practitioners. Nevertheless, no definite answer on how to improve the RT navigation by business analysts and engineers could not be given and is currently still being researched *(Malviya et al., 2017)*

Best of Both Worlds: Synthesizing the Human and Method Sides of Requirements Tracing	**Develop flexible RT tools that adapt based on human interaction.**	The scholars argue that simply focusing on improving traceability tools or processes followed by humans is not enough to achieve good RT. Instead, a better perspective looks at how stakeholders interact with different tracing tools and the optimum value for trace automation and manual interaction. Therefore, the researchers should also focus their analysis on human interaction data, improving the RT research and data quality in the long term *(Niu et al., 2017)*.
Trace Link Creation and Evolution		
Using Deep Learning to Improve the Accuracy of Requirements to Code Traceability Too Little for Big Data?	**Develop effective and flexible IR systems.**	The technique to perform accurate information retrieval (IR) is still a challenge in RT research. However, using commits, code comments, and documentation to fetch into a neural network such RNN can improve the accuracy of IR models. Especially if RNNs are used with existing external databases where developers post questions and code *(Zhao et al., 2017) (Hayes et al., 2017).* The proposal of such research topics in 2017 gives insights that even with RT techniques such as RT matrices, integration documents, hypertext, the practitioners still have difficulty maintaining good RT processes.
Trace Link Usage		
Software Engineers' Information Seeking Behavior in Change Impact Analysis	**Harvest data for an accurate Impact Analysis Assessment**	Many engineers do not consider traceability as a beneficial process. Researchers make such observations for many years. As a result, this creates difficulties for performing change impact analysis (CIA). However, each stakeholder is only interested in some specific variables of CIA, and the methods to perform such analysis is preferred to stay flexible *(Borg et al., 2017)*.

Table 5: Traceability Challenges
Source: (Giuliano Antoniol et al., 2017)

In *Table 5* above are described only a very small sample of the existing challenges and future research that should be conducted on RT. There are still many unknowns on measuring, maintaining, and promoting RT practices within an organization effectively. However, the complexity rises significantly if the RT practices are integrated with Agile methodologies (see *Chapter 2.1.1*). In such a case, the stakeholders change their perspective, start to behave strategically, both methodologies have opposing philosophies, and existing tools usually do not support both processes. Moreover, Agile methodologies are complex to integrate within the product development lifecycle when quality and regulatory standards represent a hard requirement.

2.1.6 RT Integration in Agile Environments - Challenges

In the chapter above, several RT challenges were analyzed in order to show the complexity that scholars and practitioners are currently facing. However, in this sub-chapter, the concept of

RTI challenges will be introduced. The RTI challenges are referred to as challenges related to RT integration in Agile environments. Because the requirements in Agile environments are expected to change quickly, the practitioners are faced with a certain degree of uncertainty during the product development. As a result, this also increases the complexity when compared to the traditional RT described in textbooks.

Many scholars who conducted literature studies based on RT integration in Agile have identified critical challenges which influence the accuracy and efficacy of their models and frameworks. This chapter compiles the tensions of integrating RT in Agile described in fourteen selected articles. A semantic analysis was performed so that all the identified variables are grouped into fifteen challenges emphasized in most of the papers related to RT and Agile. Also, most RT challenges are included in the RTI challenges, which are further described in the table below.

Critical Areas	Description	Articles
Identifying Granularity Level	Traceability brings the difficulty of determining the correct granularity of an artifact in the project. Moreover, a challenge relies upon defining the key information to be traced; and developing a meta-model to correctly store the links between artifacts. Granularity represents the traceability depth of the requirements that should be implemented in a model. The granularity cannot be static and depends on multiple variables such as the project complexity, team experience, size, user stories, confidentiality agreements, etc. According to *Espinoza & Garbajosa (2011)*, a user story can be mapped to several requirements, and tests become essential to support accountability of the top-level artifacts. But how much depth can be traced back to a single user story is important to consider.	(G Antoniol et al., 2006); (Espinoza & Garbajosa, 2011); (Hoang Duc, 2015)
Maintaining and Scaling the Number of Requirements	The challenge is correlated to "Link Semantics" and "Tools" and is described in the papers as a problem on effectively and accurately query large numbers of traceability links and artifacts.	(Furtado & Zisman, 2016); (G Antoniol et al., 2006); (Cleland-Huang et al., 2014)
Automating Traceability Links	The tools that provide traceability are already complex and have multiple management control models. Automating such tools and processes that follow is complex and invokes techniques such as machine learning, TDD pattern, and others.	(Hoang Duc, 2015); (G Antoniol et al., 2006); (Espinoza & Garbajosa, 2011);
Frequent Requirement Change Control	In Agile environments, one significant difficulty is constant and frequent requirement change. Therefore, the RT implementation should be designed with the flexibility to keep pace with the incoming changed requirements. However, this requires more resources and time.	(Lin & Chen, 2019)

Impact Analysis after Requirement Change	As there is no knowledge about a fully automated RT system, many entries are still done manually. This leads to errors in explicitly documenting the relations between requirements and estimates such as effort, system quality, and others. As a result, an accurate analysis is relatively hard to achieve in an Agile environment.	(Lormans, 2009); (Curcio et al., 2018); (G Antoniol et al., 2006)
Administrative Burden	As RT comes from traditional methods, it might require some additional heavy routines on top of the Agile way of working. Even though such routines can be minimized, achieving compliance from the employees represents a significant challenge.	(Lormans, 2009); (G Antoniol et al., 2006); (Jakobsson, 2009);
Regulatory Compliance	RT is critical to have industry-standard certification for any project. In Agile environments, measuring and ensuring traceability represents a challenge for regulators as well as for enterprises.	(Qasaimeh & Abran, 2013);
Measuring Project Status	Failing to accurately measure a project's existing progress and status through artifact links can lead to inappropriate architecture. Scholars have emphasized that in practice, the status attribute of every requirement is not used, which leads to the lack of accuracy on estimations. A more complex problem is created when managers try to benchmark the project status in Agile environments.	(G Antoniol et al., 2006); (Curcio et al., 2018); (Lormans, 2009)
Tools	RT differs per organization based on size, complexity, industry, regulations, etc. Therefore, the commercially available tools to tackle the RT problem can be costly, too complex, too simple, don't cover all the cases, and others. Furthermore, integrating such a tool in AE brings additional requirements such as flexibility, automation, transparency, uniformity, consistency, and others. Therefore, finding, developing, adjusting such a tool for each organization's needs represents a big challenge and costs expensive resources.	(Curcio et al., 2018); (Hoang Duc, 2015); (Lormans, 2009); (G Antoniol et al., 2006);
Costs	Integrating RT in AE brings extra costs from new routines, processes, tools, training, and maintenance.	(Jakobsson, 2009); (Curcio et al., 2018); (Ramesh et al., 2010); (Cleland-Huang et al., 2014)
Product Development Lifecycle	Agile methodologies are focused on quick product development interactions and involving the customer in the early stages of the project. Introducing in the process RT with new routines can impact project performance and time to market (TTM) for a product.	(Curcio et al., 2018); (Espinoza & Garbajosa, 2011)

Tracing Across Organization Boundaries	Complex projects can involve multiple parties. This can be affiliated divisions of an organization or external contractors that collaborate. As a result, next to the complexity of managing Agile methodologies among distributed teams or large projects, the employees should also focus on maintaining traceability of the artifacts and versioning of the product backlog.	(Hoang Duc, 2015); (Furtado & Zisman, 2016); (G Antoniol et al., 2006);
Workers Motivation	The knowledge workers who are accustomed to the Agile way of working often see the additional processes and routines, such as traceability, to directly impact their creativity and performance with little to no gain. RT benefits are long-term rather than short-term. Therefore, if an employee does not notice a direct correlation with KPIs in a short period of time, then the RT routines tend to be ignored.	(Curcio et al., 2018); (Jakobsson, 2009); (Mader et al., 2009); (Lin & Chen, 2019);
Workers Knowledge and Skills	The introduction of RT in Agile can require training for employees to get accustomed to the new tools and methodologies. Furthermore, a company might lack workers who are familiar with RT or lack significant experience. As a result, this represents a big challenge to successfully integrate new processes that require skills, certification, and knowledge.	(Curcio et al., 2018); (G Antoniol et al., 2006); (Lormans, 2009); (Furtado & Zisman, 2016)
Documentation	Agile methodologies are people-centric (as mentioned in *chapter 1.1*). The process of documenting the project progress and development is believed to produce low value for meeting the customer needs. RT quality is directly impacted by documentation. Therefore, tension arises between flexibility and RE routines. A challenge, which many organizations try to balance based on the project complexity and industry standards.	(Ramesh et al., 2010); (Furtado & Zisman, 2016); (Laux, 2019); Ratanotayanon et al. (2009)

Table 6: Challenges of integrating RT in AE

Literature RT solutions for Agile environments

This chapter further builds on the results from *Table 6* above. The goal is to analyze existing solutions in the literature and assign them to the defined challenges from *Table 6*.

Solution	Paper Title	Tackled Challenge	Article
Trace++	Trace++: A Traceability Approach to Support Transitioning to Agile Software Engineering	Identifying Granularity Level; Measuring Project Status;	(Furtado & Zisman, 2016)
Traceability Meta-Model (TmM)	A study to support agile methods more effectively through traceability	Identifying Granularity Level; Automating Traceability Links; Impact Analysis after Requirement Change; Administrative Burden	(Espinoza & Garbajosa, 2011)
Audit Model for ISO 9001	An audit model for ISO 9001 traceability requirements in agile-XP environments	Regulatory Compliance;	(Qasaimeh & Abran, 2013)
Language extended lexicon (LEL)	The impact of using a domain language for an agile requirements management	Identifying Granularity Level; Impact Analysis after Requirement Change; Maintaining and Scaling the Number of Requirements;	(Urbieta et al., 2020)
Traceability Web	Effective requirements traceability: Models, tools, and practices	Measuring Project Status; Costs;	(Kirova et al., 2008)
TraceMan	A non-invasive approach to trace architecture design, requirements specification, and agile artifacts	Costs; Identifying Granularity Level; Tracing Across Organization Boundaries; Workers Motivation	(Antonino et al., 2014)
Echo	An agile approach to capturing requirements and traceability	Costs; Identifying Granularity Level; Tools; Costs; Frequent Requirement Change Control;	(C. Lee et al., 2003)
TmM - Just in Time Traceability	Towards requirements reuse by implementing traceability in agile development	Product Development Lifecycle; Automating Traceability Links; Costs; Administrative Burden;	(Elamin & Osman, 2017)
Domain Specific Requirement Traceability	Agile requirements traceability using domain-specific modeling languages	Product Development Lifecycle; Maintaining and Scaling the Number of Requirements; Frequent Requirement Change Control; Documentation;	(Taromirad & Paige, 2012)

AMME	Agile modeling method engineering	Identifying Granularity Level; Maintaining and Scaling the Number of Requirements; Frequent Requirement Change Control;	(Karagiannis, 2015)
TraceMan (Extended)	Lightweight traceability for the agile architect	Costs; Impact Analysis after Requirement Change; Tracing Across Organization Boundaries; Workers Motivation; Requirements; Frequent Requirement Change Control;	(Gayer et al., 2016)
SPL and XP	Extreme product line engineering: Managing variability and traceability via executable specifications	Costs; Frequent Requirement Change Control; Workers Motivation; Impact Analysis after Requirement Change;	(Ghanam & Maurer, 2009)
Multi-Faced Adoption Roadmap	A multi-faceted roadmap of requirements traceability types adoption in Scrum: an empirical study	Identifying Granularity Level; Workers Knowledge and Skills; Impact Analysis after Requirement Change;	(Alaa & Samir, 2014)

Table 7: RT Solutions for Agile Environments

The analysis of existing solutions in the literature provides a good overview of the rather technical approach taken by the scholars to solve the problems of traceability and RT integration in Agile environments. Solutions such as “Echo”, “TraceMan (Extended)”, “SPL and XP”, “Trace++” and others, discuss the problem from a computer science perspective and focus on developing algorithms, frameworks, and tools to maintain traceability in dynamic and complex environments. For example, in “Echo” study, the authors used the “Eclipse” integrated development environment (IDE), to develop their tool. However, there are two problems with such an approach: 1) the IDE could become outdated and, as a result, irrelevant for practitioners 2) the scope is limited to solve only some RTI challenges such as granularity and trace links. Nevertheless, the RTI problem is broader and more complex, as described in the chapters above.

However, it should be mentioned that in studies such as “Multi-Faced Adoption Roadmap”, “TraceabilityWeb”, “AMME”, “Audit Model for ISO 9001” the focus is shifted towards product development, responsibility division, quality assessment analysis, stakeholder needs, and other aspects. Therefore, in the literature, there is a tendency to study the problem of RT from different perspectives. Nevertheless, no studies were found to take a management perspective and focus on measuring the stakeholder interests in traceability, optimizing and analyzing existing RT processes in organizations and studying the tradeoffs and negotiations required to be made by an organization given the limited resources and existing workflows.

2.1.7 Literature Stakeholder Analysis

Many projects involve stakeholders and teams with unique goals, interests, and values, that act based on their role, power, and constraints. The differences between the teams can cause different tension points and provide a good balance that can increase the product quality. Overall, the project managers, architects, developers, and others, usually agree about a set of processes to ensure that the customer demands are met, and the final product is delivered on time. However, the most significant complexity and tensions do not arise during the project planning. It usually manifests itself further in the project development, when a substantial effort should be made to maintain and adapt the initial project processes to the changing requirements. How often the requirements change depends entirely on the complexity of the environment in which the project is built to operate. Nevertheless, to reduce such tensions, the stakeholders try to balance between a more mechanistic process such as RT and the more organic one such as the Agile methodologies.

One of the approaches to balancing stakeholders' tensions was researched by *Alaa & Samir (2014)*. In their study, the authors found that the practitioners are only interested in specific requirements traceability links. These links are *stakeholder – user story*, *user story – test cases*, and *user story – acceptance criteria*. Therefore, to reduce the tensions between the stakeholders in the RT and Agile environments, each project player must be responsible only for the traceability links they find essential. However, to support such a process, it is required to use an RT tool capable of arranging the information and requirements based on the stakeholder role and needs in the project. The authors further suggest that somebody with the role of project owner will trace the least data because they are only interested in the product quality. On the other hand, the Scrum masters trace more data and are explicitly interested in test cases, stakeholder requirements, and project design/architecture data. As a result, the developers will trace acceptance criteria and stakeholder requirements more often than test cases *(Alaa & Samir, 2014)*. Nevertheless, organizations often create their own definition and responsibilities that come with each role. For example, in the *Alaa & Samir (2014)* paper, the product owner will not trace product data. However, from practical observations, the product owner role is often involved in product backlog tracking and updates. Therefore, for the current research, a clear definition per role is provided in the table below.

Role Name	Description
Customer	The customer represents a person who has a clear overview of the business needs and operational constraints of the product to be developed. The customer provides priorities on requirements and guidance during the development. The interest of the customer is about product quality and having at any moment in time a realistic view of the project status.
Product Owner	The product owner role has the responsibility to manage the product backlog and requirements database, discuss with the customer about the priority of requirements, develop the project plan, create the product development team and ensure transparency into the upcoming work.

System Architect	The system architect role is about validating the requirements discussed by the customer and product owner, design the top-level system architecture for the product desired by the customer, update the requirements database and backlog, work together with developers to co-create the system components.
Scrum Master	The scrum master makes sure the update the Sprint backlog, enforce Scrum rules, and plan the daily scrum and sprint meetings.
Developer	The developer is responsible for designing the system components described by the system architect, view assigned open requirements, test system requirements, and keep up to date the implementation status.
Reviewer (Quality Assurance Administrator)	The reviewer has the responsibility to assess if the organization standards and best-practiced are followed during the product development.

Table 8: Stakeholders Roles Definition

Stakeholders Goals and Interests

In the paper "*Acquiring Tool Support for Traceability*" by *Cleland-Huang (2012)*, the author emphasizes the importance of stakeholders' goals, interests, constraints, and their role in the project development when introducing new traceability processes and tools in the organization or a project. Such changes represent a frequent occurrence that comes from a scope change or new requirements. However, the author brings forward the idea that not all stakeholders' needs can be met. Usually, to stay flexible, the needs of the stakeholders should be prioritized unless there is an infinite budget and timeline for a project or business opportunity. The paper by *Cleland-Huang (2012)* provides a broader view on the topic of process change and managing a traceability tool. The author identifies an important stakeholder by looking from such a perspective named "*negative stakeholder*". Such stakeholders are the ones who neither want a new requirements management system nor want to change the existing requirements and processes. They usually should identified through an analysis and can take any position in a power-interest matrix model.

The most important tasks that stakeholders should undertake in an RT and Agile process match the *Alaa & Samir (2014)* and *Cleland-Huang (2012)* findings. *Figure 8* illustrates several roles during the project development and their interests and activity regarding RT. An interesting observation is that in the paper of *Alaa & Samir (2014),* the topic was analyzed from the perspective of RT and Agile, whilst in the *Cleland-Huang (2012)* paper, the perspective was only on how to integrate and new traceability tool within an organization. However, both authors reached the same conclusion when it comes to the roles and their interests in the tasks to be performed in an environment where traceability is a critical requirement.

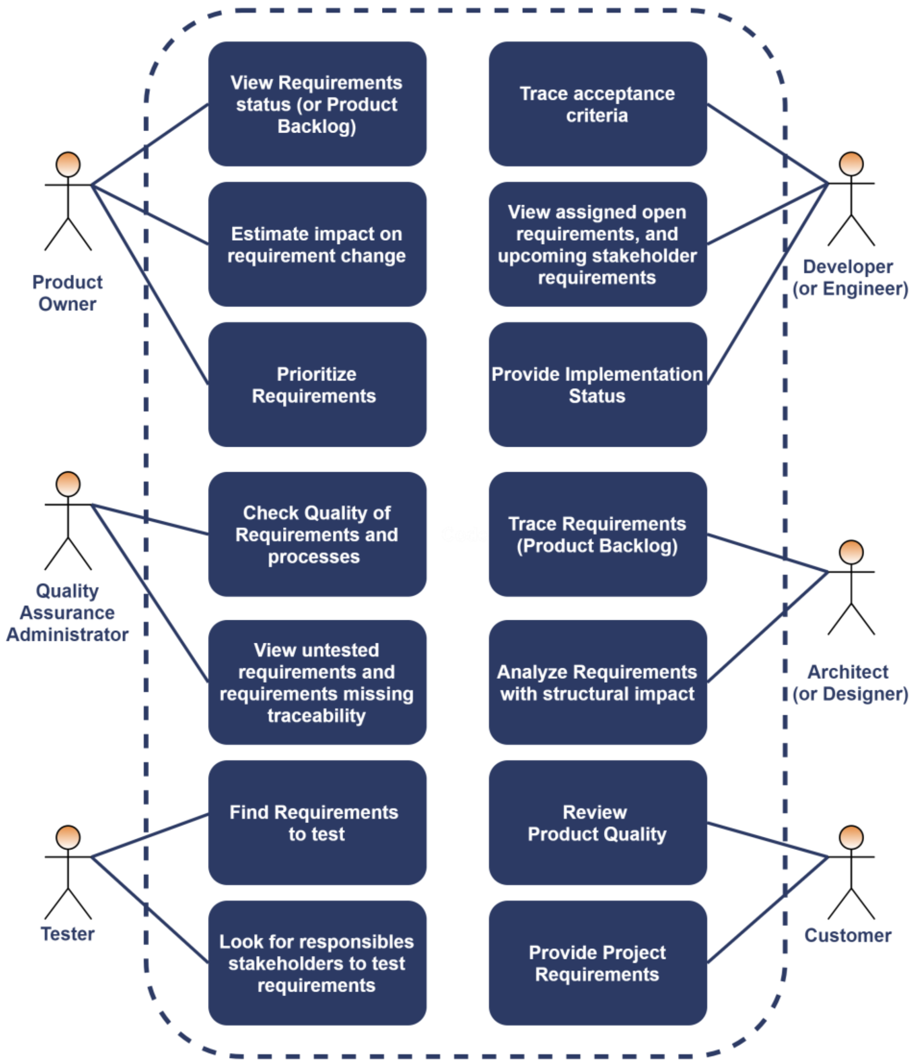


Figure 8: Stakeholder Use Cases for Traceability Interests

source: (Cleland-Huang, 2012)

Such distribution in tasks and interest of project players provides a clear picture of how to distribute and develop the initial processes to reduce tensions between the involved parties. However, *Figure 8* does not answer the problem when all the stakeholders' needs are impossible to attain.

Overall, the studies conclude the power and traceability interests of product owners, architects, quality assurance managers, system analysts, and others. Usually, the roles follow a hierarchical top-down approach in terms of power and interest in the traditional methodologies and Agile environments. *Figure 9*: Power and Interest of Stakeholders based on different goals clearly illustrate how the stakeholders can be distributed (PD circles). Nevertheless, there can be slight deviations from the diagram based on how the organization is internally structured (e.g., flat, hierarchical, organic).

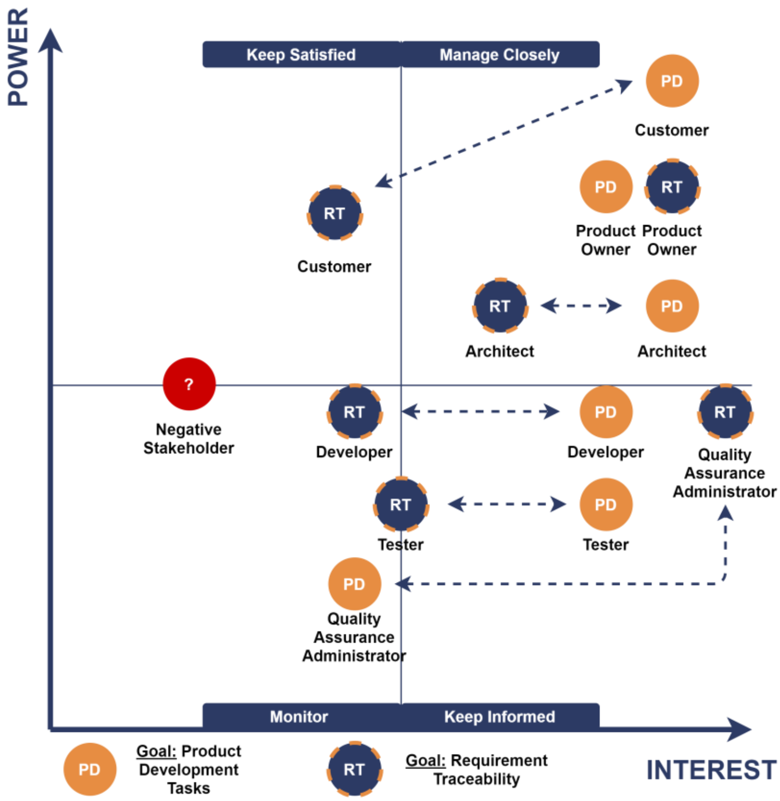


Figure 9: Power and Interest of Stakeholders based on different goals

If the goal changes from "*Product Development*" to "*Requirements Traceability,*" then *Figure 9* shows an entirely different distribution of stakeholders (RT circles). The RT represents a part of product development, and in the literature is mentioned that it can positively impact the project status. However, the stakeholders' interests and power vary from the "*Product Development*" goal. Such a difference is expected since the RT goal already enforces a specific process for stakeholders and product development. As a result, the diagram provides a good visualization of where the key tension points between stakeholders can appear. For some stakeholders, both the power and interest increase, while for others, it decreases. Therefore, to find a good balance between stakeholders, it is important to be informed about each player position within the project and how it changes with the introduction of new processes and tools.

2.1.7.1 Regulatory Inspectors and Auditors

RT does serve to enhance transparency when structured disclosure about the system is requested. Furthermore, it provides a roadmap to understand the system's integrity and safety when this is in doubt *(Kroll, 2021)*. An easily traceable system is less likely to be subjected to unauthorized changes and cause any safety or security issues. Traceability in software projects is a key criterion for auditors to certify organizations to a specific software quality standard. The auditors represent a stakeholder on the outside layer of project development and usually verify the processes enforced by quality assurance administrators within an organization. The ISO 9001, ISO 90003, ISO/IEC 12207:2008, or SWEBOK represent some of the standards for which organizations try to obtain certifications. However, such standards were developed with the traditional methods in mind and are less flexible in accepting an alternative implementation such as Agile with RT.

Moreover, the differences between the legislation and methodologies practiced by organizations leave the regulators with room for interpretation for specific articles and paragraphs specified in the legislation/standards. As a result, an organization should always elaborate with the local authorities or an expert of a particular market on the exact requirements for certifications. Such conversations can save the company precious resources by not implementing unnecessary administrative routines that reduce company productivity or completely abandon Agile methodologies.

In the paper "*An Audit Model for ISO 9001 Traceability Requirements in Agile-XP Environments*" by *Qasaimeh & Abran (2013)*, the author addressed how the Agile methodologies should be audited for ISO9001 certification. The paper describes a model that auditors can use as a guideline to check if certain organizations using Agile in their projects comply with the traceability requirements of ISO 9001 standards. The authors have developed a model containing two criteria: "*Engineering criteria*" and "*Management criteria*". These criteria grade the model based on nine "*Yardstick*s", each measuring a unique feature of RT. Some examples of "*Yardsticks*" can be summarized as: proper structure for user stories, RT encompasses entire product life cycle, artifacts relationships, traceability links are well defined, and others. Such models can be effective for practitioners to self-assess compliance with the ISO9001 standard while not losing extra resources on unnecessary details.

2.1.8 Role of Humans in Traceability Loop

The RT processes are currently heavily dependent on human actions. The *manual tracing* of requirements has shown to be unpleasant and demotivating. Therefore, scholars and practitioners continuously improve the tools, methods, and processes for automated and semi-automated testing. Thus, for Agile methodologies, which are believed to be people-centric and very flexible, the traceability solution only relies upon automated and semi-automated tools. In a *semi-automated tool*, the analysis starts with setting up the tracing tasks by selecting from user stories and backlog the high and low-level artifacts with a specific tracing technique (see *Chapter 2.1.3.2*). Afterward, the tool generates a graphical representation of the links (e.g., traceability matrix (RTM)). The practitioner carefully analyzes this matrix and adds the missing traces or refactors the incorrect links. The benefit of the human-software tracing is higher accuracy, but it can introduce the cost of high effort when the tool fails to establish the traceability links correctly.

For the Agile way of working, even minor inconsistencies in the tool can be seen by knowledge workers as an administrative burden with no benefit. Therefore, the reliability and versatility of the traceability tools should represent one of the key metrics in an Agile environment. However, a study conducted on human behavior and traceability tools by *Cleland-Huang (2012, pg. 260)* show that high accuracy of TMs "*does not lead to better analyst performance in semi-automated tracing tasks do not always lead to better results*". Furthermore, the author mentioned that an equally important indicator besides high accuracy is the ***human-computer interaction (HCI)***. The author says that HCI is important because, with the poor UI capabilities of a tool, the analyst is prone to more errors and less motivated to further continue with traceability. Therefore, a new direction of traceability research can be on developing front-end services that increase analyst performance. Such an approach is different from what currently exists in the literature because it focuses more on human nature than statistical analysis, algorithms, and requirements link semantics.

2.2 Part II – BPMa and RT

2.2.1 Business Process Management

For many projects, the organizations follow a set of business processes that represent the core activities that should be accomplished to meet the customer's requirements or the milestones proposed by the project managers. According to *Jander (2016)*, business process management (**BPMa**) is a topic that is being intensively researched because it represents a combination of best practices, technologies, and methods that can maximize the performance of the organization. BPMa can be used to target specific or multiple departments within a business entity so that KPIs are improved through effective cooperation, clear process visualization, and possible automatization. According to *Weske (2007),* BPMa can be defined as "*a set of activities performed in coordination in an organizational and technical environment*. *The activities jointly realize a business goal*". The business goal can vary based on the organization's needs. However, in the current research context, the goal could be to effectively integrate RT processes with Agile methodologies.

The core of BPMa is about how to effectively structure the business processes to reduce human error and arbitrary decision-making. However, not all the processes can be structured to the degree that they are flawlessly maintained and that no impediments to the organization's goals are encountered. In many cases, businesses are confronted with decisions where they have to rely on the training of their employees or on good judgment to reach the initial established goals of an organization or project (*Jander, 2016)*. Furthermore, the methodology focuses on a specific business process to improve, and usually, the processes vary in terms of structure, collaboration, repetition, and complexity. According to *Leymann & Roller (1999),* the business process can be categorized based on how high or low is the ***business value***, ***degree of repetition,*** and ***degree of structuring***.

- The "*business value*" describes how important is the business process for the organization. The processes with a high "business value" are usually at the core of an organization and directly contribute to the overall value the business brings to its customers and partners.

- The "*degree of repetition*" refers to the number of times a process is repeated within a company. For example, business processes aimed at mass production are usually given a high mark of repetition. Such processes generally receive a very large budget for the design phase because the expenses are shared between repetitions. On the other hand, the processes that receive a lower score for repetition are only used once, and their form remains unchanged until the specific goal is achieved. According to *Leymann & Roller (1999),* the processes with a low repetition score are usually designed to facilitate the collaboration between participants.

- The author described the "degree of structuring" as a process with a high degree of detail for most activities and execution constraints. A high value of structuring usually requires little to no decisions to be made by the stakeholders. The outcomes and steps to be taken are already clearly elaborated in the business process. A structured process requires no independent judgments by the employees and therefore minimizes the chances of human error. Such processes are also usually very repetitive across projects and departments.

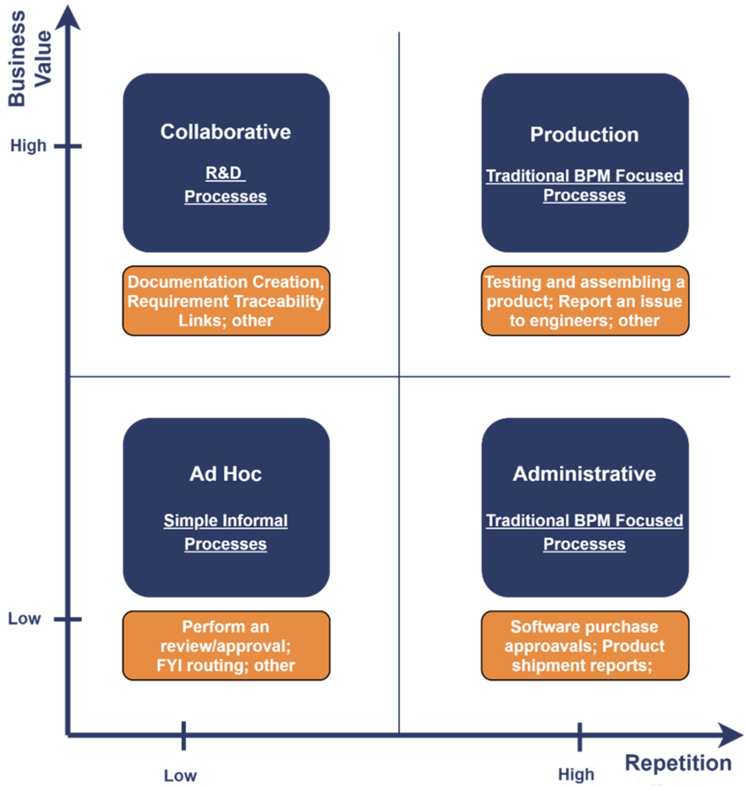


Figure 10: Traditional Business Process Management Focus

Source: Leymann & Roller (1999)

Furthermore, according to *Leymann & Roller (1999)*, the processes can be divided into four classifications: “Collaborative”, “Production”, “Ad Hoc,” and “Administrative”. Each of these classifications has specific strategies and design patterns for process development. For example, the “Collaborative” processes are used for scenarios with a high degree of business value. Still, they are designed specifically for a type of environment that creates relatively low repetition and structure. In the case of RT processes in Agile environments, it can be argued that these can be categorized as “Collaborative,” especially if the process is in its initial phases. RT processes are constantly challenged by events were external judgment or experience of employees should play a part in the decision-making process. As a result, the practitioners use a more iterative approach to business process design, as shown in *Figure 11*. The BPMa lifecycle model (shown below) is elaborated in detail in the research of *Jander (2016)* and *Razavian et al. (2016)*. The authors describe the model as an important tool for evaluating the business processes and workflow management for different organizations. However, the BPMa lifecycle has different starting points. If a process has already been designed, the managers should start the process improvement from the “*process identification*” phase. If the process was not yet elaborated and implemented, then the lifecycle starts from the “*process design*” phase.

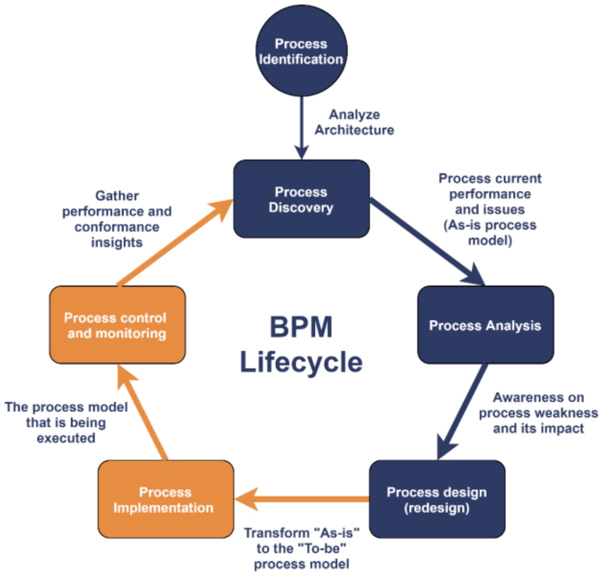


Figure 11: BPMa Lifecycle Design and Runtime Phases

The BPMa lifecycle is described in the literature as follows:

- In the paper by *Razavian et al. (2016),* the BPMa lifecycle starts from the "*Process Identification*". In this phase, the managers overview what processes are relevant and how the processes are correlated. The result of this step is a list of processes and a top-view visual diagram to show the relationships between them.

- The "*Process Discovery*" phase is where the content of the processes starts to be elaborated. The managers discuss the roles of stakeholders, the resources, and the responsibilities that have to be fulfilled. Furthermore, a description is done about how the data such as documents, requirements updates are communicated between tasks and how different processes should be linked.

- The next phase is "*Process Analysis,*" where the current performance of the process is being evaluated. In this phase, the weaknesses and bottlenecks of the process are assessed. Furthermore, if certain quantitative measures for the process are available, an analysis of different KPIs is also performed.

- The "Process Design (redesign)" action list differs based on the business processes status. According to *Jander (2016)*, in the case of initial process design, the managers in this phase should develop strategic planning to achieve a strong linkage between business goals. This phase requires a business process modeling language that can clearly describe the process behavior while sustaining the strategic planning of the business. In the case of "*process redesign,*" *Razavian et al. (2016)* highlights that this phase should be focused on implementing the solutions discovered during the process analysis. The managers can evaluate several solutions based on their complexity and the impact on the KPIs.

- "*Process Implementation*" is the first runtime phase where the developed business model is tested within the organizational context. The new or changed process might require changes or process automation. If the automation changes directly affect the human factor, usually this is accompanied by many tension points and extra expenses that go into negotiation moments. However, the managers typically take into account such outcomes beforehand and create the implementation plan accordingly.

- The second runtime phase is "*Process control and monitoring,*" in which the management team can analyze information about the processes that are running in the real environment. The monitored information can be the start and duration of a task, interaction of the participants with the workflow process, and the errors, bottlenecks, issues encountered over time. All the collected data can be used as an input to a new BPMa lifecycle, eliminating design flaws and improving performance.

2.2.2 Business Process Management Lifecycle Challenges

Based on *Jander (2016)* and *Razavian et al. (2016)* studies, the BPMa lifecycle phases are susceptible to bad design practices due to several factors that practitioners should consider. According to *Jander (2016),* when practitioners develop business processes, they tend to focus on "*Administrative*" or "*Production*" processes which are very mechanistic in nature. However, very little attention is given by the researchers to the development of "*Collaborative*" processes, which entail a certain degree of flexibility when designed, implemented, and monitored. The author argues that a more flexible process can be achieved if a goal-oriented approach is taken. "*Workflow model agility*", "*Strategic Operational Cohesion,*" and other goals examples should be used for every step in the BPMa lifecycle such as "*Analysis*", "*Design*", "*Implementation*".

From *Razavian et al. (2016)* perspective, every BPMa lifecycle phase comes with cognitive biases that lead to bad design. The biases can introduce an incomplete process model because participants have only partial knowledge about the system operation or the number of participants in the process discovery is limited. Furthermore, the stakeholders can have separate views on what processes should be changed and which ones have good performance indicators. Moreover, stakeholders such as managers, product analysts, product owners have in mind a map of which processes are wrong and should be changed. As a consequence, omitting the complete process analysis phases and failing to discover the real bottlenecks and system weaknesses.

2.2.3 Business Process Modeling Languages

In the business process management field, besides identifying business processes running within an organization, it is equally important to acknowledge the procedure of defining the language needed to build the models to support the business strategy and goals. Therefore, a process modeling language represents the key elements that should be used to understand better the business processes and strategic design workflows that can be automated and improve the project or business KPIs.

Process modeling languages aim to support contrasting demands of two groups of users *Jander (2016)*:

- The ***business users*** mainly focus on the top-level business processes, strategies, and goals of the project or organization they are activating. The user is expected to have the expertise, different perspectives, and good understanding of the business processes within the organization the user is performing its daily activities.

- The ***technical users*** are characterized to have expertise in the technical fields of IT software systems used to collaborate, design, and support business processes.

Therefore, because of the difference between the two groups, the process modeling languages are known to be divided into three categories: *informal*, *formal*, and *semi-formal*. Each of these categories are used based on how diverse the audience is, how many details are wished to be conveyed, or portray how the processes communicate and interrelate. Based on the *Jander (2016)* study, the usage of each category is described in the table below:

Category	Description
Informal	An example of informal language is the natural language used to describe the business processes, usually in a written form. The informal language describes the business processes on a very abstract level and does not have any semantics. Therefore, when conveyed to a specific user in a group (business, technical), the messenger and the receiver can interpret it differently. The benefit of using informal language is to reach a large audience of technical or non-technical stakeholders involved in a process design.
Formal	The formal language is the modeling language used to design well-defined business processes and workflows. The formal language has strict semantic definitions for each of its components. This means that the language has a single way of information or process interpretation. Furthermore, the formal languages require model designers to know the semantics and develop valid execution processes for IT-based systems.
Semi-formal	The semi-formal language represents a conveyed combination of informal and formal language aspects. The designers use it to develop less known processes in natural language and the processes that are clearly defined in the formal language. This helps the designer not be overwhelmed by forcing himself to formalize executions subjected to change from the start. Examples of such languages are BPMN and CMMN described in the sub-chapters below.

Table 9: Business Process Modeling Languages Categories

In practice, the researchers have identified that practitioners use many language variations and semantics to describe their business processes. According to *Jander 2016*, one representation of the modeling language can be a *textual representation*, for which the designer should use specific text editors which support the semantics of the language. However, a more common approach found in practice is the *graphical representation* of the language. The graphical representation of information has a significant advantage over the textual representation by providing really quick overviews of the business processes. Moreover, graphics are usually more intuitive and pleasant to people, as it requires less analysis to understand the process flow and the correlations that exist.

2.2.3.1 Language Selection

Selecting the correct language for mapping internal business processes does not represent an easy task. Each organization has its own set of requirements and approach towards process mapping. Therefore the language should be able to support the business goals and the workflow of the company employees. According to *Lu & Sadiq (2007)*, the BPMo languages (graph-based or rule-based) can be analyzed through the five key criteria: *Expressibility*, *Flexibility*, *Adaptability*, *Dynamism, and Complexity*. Furthermore, the research of *Pereira & Silva, (2016)*, which builds on the findings of *Lu & Sadiq (2007)*, and several other researchers, have introduced ten additional criteria for the process of language selection. Several of these criteria were not selected for further description due to the substantial similarity between their concepts. For example, criteria such as *Usability* and *Readability* mentioned in the research of *Pereira & Silva, (2016)* are included in the definition of *Complexity* in the table below.

Language Criteria	Description
Expressibility	The possibility the language offers to represent different organizational use cases. The language clearly can express the process requirements and process modeling. It uses clearly defined structure, execution, business processes, information transactions, data management, etc.
Flexibility	Flexibility represents the capability of the language to execute the business process based on a partially specified model, where all the additional details are added at runtime.
Adaptability	Adaptability is about the capability of the language to allow the creation of process workflow that can react to certain ad-hoc circumstances and enable the execution of one or several processes simultaneously.
Dynamism	Refers to the ability to be able to modify the business process workflow when the processes evolve or should be replaced. The evolution can be referred to as a process improvement or something more complex such as complete process innovation.
Complexity	This criterion helps to analyze the difficulty to design/model, analyze and deploy in production the mapped processes. For example, the complexity can be increased or reduced due to availability or lack of tools or support.
Formality	Refers to the property of the language to reduce the ambiguities in the business process model interpretation and to be accurate in description while keeping the formal aspect in place.
Versatility	Versatility is about the suitability of the language for different tasks. For example, is the language suitable only for analysis of the processes and documentation, or can it also be designed to be executable and deployed on different machines?
Universality	Universality represents the level of awareness (high or low) about the language among the knowledge workers and the level of support from the users in terms of implementation in the business environment.
Concision	The ability of the language to describe different characteristics/angles of specific business processes using a smaller set of components.
Tools Support	The concept refers to the availability of tools to support the main workflow and semantics of the language.
Ease of Learning	Refers to the amount of work required to get acquainted with the language and master the key components to developing industry-standard designs.
Collaborative Work	The criterion is about the property of the language to support collaborative processes (work). An example can be system architecture design or any type of meeting.

Table 10: BPMo Language Selection Criteria

The criteria defined in the literature provide a good base to categorize and select a particular language for process modeling. However, the relevance of the criteria and the weights given to each language in practice is less prominent when there is no desire from employees to introduce any type of change in their workflow. Nevertheless, the criteria could help to communicate better the concrete reasons to refute a particular language selection for process mapping.

Practitioners and scholars use multiple languages and platforms to describe their business models. Furthermore, there is no definitive universal tool that satisfies the needs of all users. There are numerous languages due to each user's background difference and different concepts and processes the user wishes to describe or automate. However, it is of crucial importance to know and understand the usage, advantages, and limitations of several existing languages before selecting a particular one to describe the business operations, processes, or system. In the following sub-chapters, three languages will be emphasized: Flowcharts, BPMN, and **CMMN**. These languages can describe and help to automate ideas with different levels of complexity, flexibility, formality, and other criteria that might be important for specific stakeholders.

2.2.3.2 Flowcharts

Flowcharts represent one of the earliest modeling languag that was introduced by *Gilbreth & Gilbreth, (1921)*. The flowchart language was quickly accepted as a means of communicating different business processes and system behaviors among various organizations. The language was later adopted by the American Society of Mechanical Engineers as a standard across many industries. Over time, the practitioners have introduced slight variations to the language in terms of components that are used to describe a particular process *(Jander, 2016)*.

Because of its visual simplicity, the flowcharts require little to no experience in process modeling in order to convey a certain workflow, algorithm, or describe the process of solving a particular task. In addition to the largely used flowchart components (see *Table 21, Appendix B* – BPMo Languages), there are several which can help to design more complex workflows. These components are "*or-junctions"* and *"summing-junctions,"* which, as in the "Decision*"* flowchart component (see *Table 21*), helps enrich and direct the process flow. The language also offers annotations for different data storage methods such as "*databases"* and "*stored data,"* which refers to hard drives, memory cards, or any other storage devices. Furthermore, the language can describe data access methods such as "*Sequential Access Storage"*. Also, the designer could provide data input methods such as "*Manual Input"* of data or describe a "*Manual Operation"*. As a result, it can be argued that the language provides all necessary components to describe simple business processes and requires little time to get accustomed by knowledge workers.

The introduction of Flowcharts at the beginning of BPMo chapter was done due to the language's simplicity. The Flowcharts are most commonly adopted and known by knowledge workers who fulfill different organizational roles. In SMEs that practice the Agile way of working and do not have any methodology for describing their business and system processes, Flowcharts can represent the first step towards business process modeling standardization. According to *Wiemuth et al. (2017)*, standardization of business process modeling has increased workers' productivity and helped reduce systematic errors. Consequently, this allowed the knowledge workers to further plan for applications which include automatization of different systems and processes within the organization or specific project.

Flowchart Disadvantages

The first disadvantage of the flowchart is the fact that the language instructions and elements are written in the natural language, which as a result, cannot be interpreted by IT systems. Unless, when complex natural language processing and machine learning algorithms are applied. The lack of executable flowchart models is not a mistake by design. Instead, the language was specifically developed to assist the workers in communicating system or business processes to other workers. Furthermore, the computer systems that are currently capable of running complex execution models on the local PC or remote servers were not available when the language was developed *(Jander, 2016)*.

The language is also lacking key components such as events. Events are referred to items that are part of the process which can be triggered by external stimulus and can change the workflow of a particular process. Furthermore, suppose an organization would like to move to process automation. In that case, the business processes designed using flowchart diagrams will have to be translated and redesigned in more syntax-rich languages such as BPMN. As a result, this could increase the cost and introduce steep learning curves for the knowledge workers.

2.2.3.3 BPMN

The BPMN is part of semi-formal languages that can target audiences with different knowledge backgrounds when communicating the business processes. The language was developed by the Business Process Management Initiative (BPMI), which at a later stage was taken over by the Object Management Group (OMG) *(Jander, 2016)*.

The informal part of BPMN is represented using graphical visualization. It is usually a **non-executable** model used for portraying a clear process flow necessary to maintain the business's daily operations or project timelines. The informal BPMN can be developed as a means of documentation to which every stakeholder can refer at any moment in time. Therefore, to ensure clear communication among stakeholders, a universal business process communication language should be adopted in the first place. As a result, it can be argued that for R&D or small organizations, non-executable BPMN can be the first step towards bringing more value to the customers and the business itself. The non-executable BPMN is also more simplistic in terms of the technical infrastructure required to design the processes. It can usually be developed either by *business* or *technical* users, which primarily design the model using symbols widely used in practice *(Recker, 2010)*.

According to *Aagesen & Krogstie (2015, pg. 238)*, several case studies have shown that the core construct of the BPMN language is known by the organizations. However, the practitioners tend to disregard the extended set of components that provide a richer meaning and understanding. Furthermore, the knowledge workers also tend to neglect the standardized rules of BPMN to fit the local needs, such as improving readability or simplifying the modeling and design tasks. Nevertheless, it should be mentioned that the case studies were conducted in the years 2008 and 2011. In the above-mentioned timeline, the BPMNs 1.0/1.2 were still used by organizations where the diagrams have to be mapped from “valid” BPMN to BPEL (Business Process Execution Language), which the engines could interpret and execute. However, the model was missing concrete process execution semantics. Therefore, as the language was mainly used to communicate internally, and the execution was a complex task, the practitioners made changes as they saw fit. With the introduction of BPMN 2.0 and the improvements it brought to the execution semantics, events, choreography diagrams, and appearance of intuitive design tools, the process designers were put into an environment to follow a more standardized pattern for the BPMN design so that the engine can interpret the results. Hence, it can be argued that the risk of tailoring the BPMN standards should be less common with BPMN 2.0. However, the workarounds can still represent a threat for organizations using the non-executable models. Furthermore, in the *Aagesen & Krogstie (2015)* book, there is no data regarding the standardization metrics and values achieved with BPMN 2.0, and the research available on the topic is limited.

The **executable** BPMN requires a closer collaboration between the business and technical users. It represents a platform composed of complex components such as local or remote servers, data analytics tools, content management services, user rights administration, databases, and others. The decision to use an executable BPMN in the daily business operations or in a particular project should be supported by a thoroughly carried analysis for the return on investment, available knowledge capital of the company, existing business processes, potential improvements, and future perspectives. The decision should also be accompanied by extensive research using the BPMa lifecycle and the BPMa categorization diagram as analysis tools.

Language Components

With the release of BPMN 2.0 in January 2011, a set of new features were added to the language. Therefore, to provide a better overview of the BPMN main components, these were categorized into four main groups: *Flow Objects*, *Connecting Objects*, *Pools and Swimlanes*, and *Artifacts*.

Flow Objects

The flow objects are composed of three key components: *Events*, *Activities* and *Gateways* (see *Jander (2016)*, for more details).

Component	Description
Events	These represent the trigger to start, modify or end/complete a process. Events can be further divided into three types: *Start*, *Intermediate* and *End*. The Start and Intermediate events can include additional symbols which describe in more detail what is the trigger for the event. For example, it can be a message, timer, error, conditional, escalation, and others. 
Activities	The activities represent specific tasks performed by a person or by a particular system (hardware or software). The activities can be divided into four types: *Tasks*, *Transactions*, *Event sub-processes*, and *Calls*. It should be noted that all the activities can be generic or triggered by a particular event. – **Tasks** represent single actions that occur in the business processes. Some examples of tasks can be updating backlog, creating RTM, writing documentation, etc. Overall the tasks represent actions that are performed by the process participants and are usually the most common in BPMN diagrams. – **Transactions** are a sub-process component that constitutes a payment process. The transactions verify if all participants have completed their parts of the transaction before finalizing a particular sub-process.

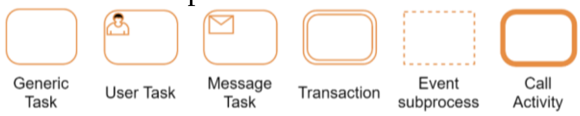
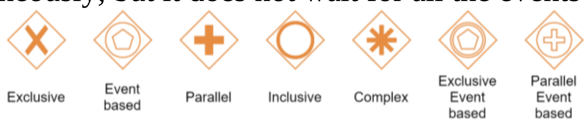
	− **Event sub-processes** are triggered by a start event and differentiate because the component does take part in the business process's regular flow. These can be interrupting and non-interrupting − **Calls** are activities that are external to the process definition and allow the creation of process definitions that can be reused. 
Gateways	The gateways should be viewed as decision points that can adjust/modify the process flow based on certain events or conditions. Gateways do not represent decisions; neither should these be viewed as components that make decisions. Instead, the gateways dictate the flow of the process. BPMN 2.0 differentiates between seven different types of gateways: *Exclusive*, *Event-based*, *Parallel*, *Inclusive*, *Complex*, *Exclusive Event-based*, *Parallel Event-based*. − **Exclusive** gateway routes the sequence flow to one of the outgoing branches of the flow. It waits for the incoming branch to complete before triggering the outgoing branch. − **Event-based** gateway represents a branching point where the alternative paths are followed based on the events that occur. − **Parallel** gateways ensure that all the outgoing branches are activated at the same time. When the merge occurs, the gateway waits for the incoming branches to complete before activating the outgoing flow. − **Inclusive** gateway is one of the most commonly used in BPMN designs. When splitting, one or multiple branches can be activated. The incoming branches should all be complete before the sequence flow continues. − **Complex** gateways are used less frequently than other gateways. However, in specific scenarios, these can simplify the need to use a combination of multiple gateways. An example can be when a parallel gateway starts multiple flows, but only several are required to complete to trigger the outgoing flow. − **Exclusive event-based** gateway is used to start new process instances when the alternative paths are determined by events instead of conditional flows. An exclusive gateway can be used when the decision about some alternative approach should be taken by someone external to the process (e.g. customer). − **Parallel event-based** represents a combination of the event-based and parallel gateways. It allows for more than one process to happen simultaneously, but it does not wait for all the events to arrive. 

Table 11: BPMN Flow Objects

Artifacts

The artifacts in BPMN are used to bring an additional level of detail to the designed diagrams. The artifacts can be used to represent data from different perspectives as well as to add notes that describe the process better. There are three types of artifacts: *Data Objects*, *Group Objects*, and *Annotation Objects*.

Component	Description
Data Object	The components help to represent business data that is being interchanged in a BPMN process. Data objects can be attached to different associations which can denote if the data needs to be stored, collected, or processed. The data object can be of five types: data *Input*, *Output*, *Collection*, and Data *Storage*. – **Data input** can be seen as an additional requirement necessary to complete a certain task. If the task to be completed is data-dependent, this is usually shown with the data input object. – **Data output** is usually used when a particular process or task generates data. An example can be the process or task of developing the product backlog or conducting a survey. – **Data Collection** symbol is used when there are multiple documents and other datasets that should be handled by the process. An example can be a collection of requirements from different sub-systems, invoices, etc. – **Data Storage** is used when the data collected from the process should be temporarily stored for later use. The data storage component can be accessed at any moment in time during the process. It basically represents a database. 
Group Object	The group object is straightforward as these are only used in the process flow to organize tasks that have significance to the process or structure the overall BPMN design for better readability. 
Text Annotations	The designers use the annotation object to convey a certain message about a particular task or processor to increase the readability and understanding for the other stakeholders involved in the business process design. 

Table 12: BPMN Artifacts Description

Pools and swimlanes

A distinctive feature of BPMN is the ability to clearly divide the process flow and tasks based on the participants' roles and other possible external collaborators. In the context of BPMN, the pools are used to define the major participants in a collaborative process. Furthermore, because each participant has its own process flow and states, BPMN guidelines do not allow to connect direct *sequence flow* between elements located on different pools (see table below).

A pool can be further divided into swimlanes. The swimlanes can be used to specify a particular group of users who are in charge of a sub-process. For example, a can pool can be an organization, and the swimlanes can represent different departments within that organization. Using swimlanes helps to clearly define who is accountable (responsible) for some part of the process. As opposed to the pools, the *sequence flow* between the swimlanes is allowed. However, this usually takes place via the *Message flow* components. Furthermore, the swimlanes can be subdivided with additional lanes which are nested.

Connecting Objects

Similar to other graphical-based languages, there are components in BPMN that show the relations, associations, and the sequence and message flow of its components. The connecting objects represent lines that provide a clear view of the order of activities to be performed. There are three symbol types: *Sequence Flow*, *Message Flow,* and *Association Flow*.

Component	Description
Sequence Flow	The sequence flow shows the main order of activities that the model performs. It is graphically represented as an arrow. The direction the arrow points represents the sequence flow for the business process. The sequence can be default or conditional. 
Message Flow	The symbol shows the messages that flow across the pools. The message flows are used to connect different tasks between the swimlanes. (see the sub-paragraph below) 
Association	The association symbol shows the relations between the artifacts and the flow objects. The symbol is also used to connect data stores, data objects, and text annotations. It can be directional or bidirectional. 

Table 13: BPMN Connection Symbols

2.2.3.4 CMMN

In the paper by Wiemuth et al. (2017) is mentioned that formalization and standardization of process have shown to increase productivity and allowed to broaden the capability to apply automation systems to a larger extent. However, the highly standardized processes designed through languages such as ***BPMN could create difficulties for developing more dynamic processes due to the language semantics***. Practitioners and scholars have raised concerns that the BPMN 1.0/2.0 language lacks flexibility when it comes to design processes that account for ad-hoc and dynamic events. Such events usually require a non-standard procedure to be followed by stakeholders.

To address the issues raised for the BPMN language, in 2014, OMG has released the Case Management Model and Notation (CMMN) 1.0. In 2016, OMG introduced CMMN 1.1, which brought better visual differentiation between connectors, text annotations, the possibility to add extensions, and many other technical fixes. The language was developed to address the problem of designing models for unpredictable processes. However, CMMN was not designed with the intention to replace the BPMN.

One of the key differences between BPMN and CMMN is in the workflow. According to *Marin (2016)*, CMMN represents a declarative language where the modeler describes "what" is allowed or forbidden in the process. On the other hand, BPMN can be described as an imperative language where the modeler describes "how" to proceed in each step of the process. Furthermore, the system (model) developed using BPMN provides a clear roadmap on how to achieve the business goal, and the model uses the workers to achieve the initially established business goal. Contrarily, the case management systems, such as those developed using CMMN, knowledge workers take complete responsibility to accomplish the business goal and use the model as a tool to achieve the goal *(Marin, 2016)*.

The case management systems rely on the knowledge worker's expertise to deal with a particular event. However, as in BPMN, the CMMN language provides a set of components that help the workers better communicate and improve the case management process. These components are briefly described in *Table 22, Appendix B* – BPMo Languages.

CMMN Disadvantages

CMMN was introduced by OMG with the initial goal to fill in the gap in BPMo languages when it comes to case management and ad-hoc processes. In the beginning, the language was supported by practitioners and viewed as a valuable technique to improve the organization's productivity. However, as several companies developed tools and created language engines to support the CMMN semantics, the practitioners became more reluctant when it came to the implementation of CMMN in the organization processes.

In an article posted by *Deehan, (2020)* titled "*How CMMN never lived up to its potential.*", the author described how Camunda, an organization specializing in BPMo languages such as BPMN, decided to leave the CMMN support behind after several years of development and a couple of releases. The reason behind the leave was related to the lack of traction from organizations when a functional engine was presented by the Camunda organization. In addition, the practitioners were reluctant to switch to CMMN because of the burdensome task of learning a new language, which was characterized by a steep learning curve. Furthermore, the author argues that the many processes considered ad-hoc or unpredictable, the practitioners were still identifying repetitive elements that could be modeled using BPMN. Another disadvantage of CMMN, as *Deehan (2020)* argued, was that the practitioners were missing an explicit event modeling and labeling, which made the CMMN processes often hard to read.

3 Theoretical Framework

The viewpoint adopted in this study is that many SMEs that integrate RT with the Agile way of working will face several tension points because of opposing processes required by the two methods during the project development. Furthermore, this study analyzes if the challenges can be managed using a business process modeling approach.

The extensive literature analysis has shown that the tensions points are not limited only to the integration challenges of RT in Agile, but also extend to the stakeholders' interest in traceability, alongside their roles and power within the project (*see Chapter 2.1.7*). Furthermore, RT processes face challenges such as developing accurate information retrieval algorithms and developing accurate impact analysis assessments (*see Chapter 2.1.5*). The tension is also increased when the managers have to decide what traceability technique to use, which traceability links to form, and what will be the granularity of RT (*see Chapters 2.1.1, 2.1.2, 2.1.3*). Moreover, the decision-makers have to consider the business and project goals, project complexity, and as a result, design a business process model that can support the product development, but not further increase the administrative burden and project complexity. However, the task of designing a good BPMo comes with its challenges. Some of these challenges are: standardizing on a BPMo language in the organization, identifying the key processes, maintaining a flexible system, and selecting a BPMo language capable of describing and support all the business needs (*see Chapters 2.2.1 and 2.2.3*).

The organizations that try to enable RT and the Agile way of working are faced with a complex problem. However, not all organizations are equal. They don't have the same mission, objective and are not operating under the same regulations and governmental control. As mentioned by *Blaauboer et al. (2007)*, adopting RT is essentially a matter of choice, whether to trace requirements or not during the development. Such decisions are made from analyzing the available alternatives, which through the irrevocable allocation of resources, is expected to have revocable resources such as time, money, innovation, and others. Therefore, an organization might not face all the challenges and tension points described in the literature since the return on investment to implement a complex RT and Agile process might be negative. As a result, the management might opt for a more simple technique to reach their target goals.

Considering all the complexity and tension expected from integrating RT methods with the Agile way of working, one of the theoretical framework's goals is to emphasize how the tension points are correlated. Furthermore, attention is placed on the lack of research on the topic of how to identify which tension points are the most critical for the business or project that is being developed. This is because the business missions, values, and needs, are different and so are the combination of the RT and Agile integration challenges faced by them. The analysis from *Chapter 2.1*, helped further to identify the details of the existing research gap. Therefore, it is possible to conclude that very little research was done on developing a requirements traceability integration (RTI) framework. *Therefore, this study focuses on developing a clear RTI framework and BPMo roadmap to facilitate the RT and Agile integration process at ABC and any similar SME.*

3.1 Tension Points - Correlation

From the literature study conducted in *Chapter 2.1,* several tension points have been identified when it comes to integrating RT with Agile methodologies. These tensions were studied by scholars using different perspectives. For example, in the papers by *Cleland-Huang (2012)* and *Alaa & Samir (2014),* the RT in an Agile environment is analyzed through the perspective of the stakeholder roles and their interests (*see Chapter 2.1.7*). It is concluded that developing a clear roadmap for the stakeholder interest in traceability can facilitate in the end overall RT process in every environment. Every stakeholder will not be subjected to tasks characterized by them as burdensome or irrelevant. However, if there is an increase in the project complexity, this can directly add new challenges such as *Costs*, *Tracing Across Organization Boundaries*, *Maintaining and Scaling the Number of Requirements*, and others. These new challenges can directly affect the stakeholders' collaboration and add extra administrative burden because of an increase in hierarchy and process management. Therefore, stakeholders' interests in traceability cannot be assessed separately from the project complexity (see *Figure 12*). The project complexity can be defined based on the *certainty/uncertainty* in the technology to be developed and the *agreement/disagreement* in the project requirements (*Stacey, 2007*).

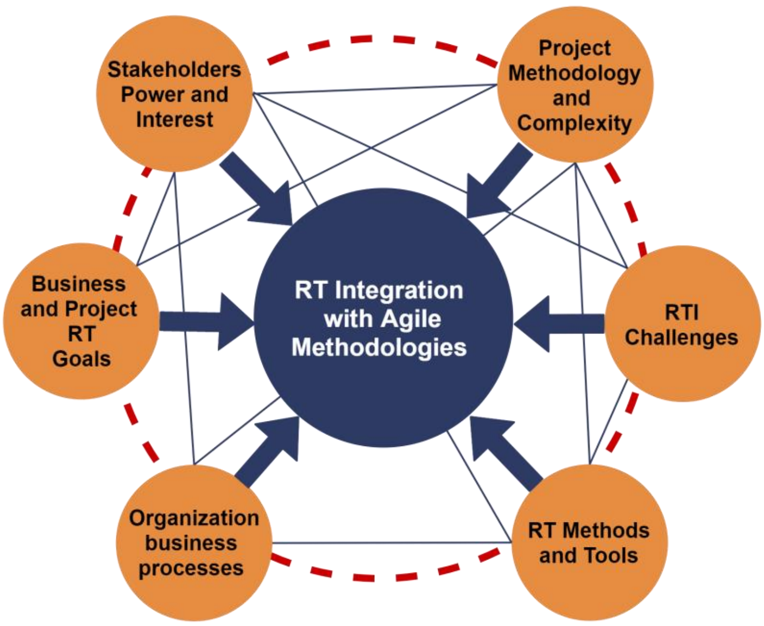


Figure 12: RT Integration with Agile Tension Points

In a complex project where RT and Agility are a priority, the management team should carefully assess which RT links, granularity level, and techniques are the most appropriate to meet their traceability goals. As mentioned in the papers by *O. Gotel et al. (2012)* and later *Cleland-Huang et al. (2014)*, there are several traceability research goals such as: *Cost-Effective*, *Scalable*, *Valuable* and other. These research goals, as mentioned in *Chapter 2.1.4*, can also be translated into business goals. However, if the managers do not prioritize their RT business goals correctly, this can lead to an over-engineered traceability process. For example, a traceability system designed to be *Scalable* and *Cost Effective* will have a different level of complexity when compared with a traceability system that should be *Portable* to other projects. As a result, an incorrect assessment of the traceability business goals can introduce complex RT methodologies in the project, bringing new RT challenges and further increasing project complexity. In addition, the traceability techniques affects stakeholders' interests and brings new RT and Agile integration challenges such as a decrease in *Worker Motivation* or requires new *Knowledge Workers and Skills* to maintain new complex processes (see *Chapter 2.1.5*).

Such scenarios can be triggered from any of the tension points represented in *Figure 12.* As previously described, these can create a chain reaction that makes the organizations' processes or product development more complex. Despite the intertwined correlation shown in *Figure 12* above, it is still possible to identify where a chain reaction starts and a possible end. However, when these occur in complex real-world environments, it can take a long time to see the complexity of the project increasing. The changes compound over time, and when an audit is performed to see what decision caused issues within the project, it is very hard to trace. Processes, requirements, documents, and others are changed over time by many stakeholders who usually follow their interests and goals. In SMEs with a very flexible way of working, the chain reaction phenomena can have very destructive implications on the projects. As a result, it can be argued that ad-hoc decision-making, when it comes to RT, should be carefully assessed and removed on the process level rather than focusing the RT strategy on specific challenges, tools, and methodologies.

RT Research in Practice

The strong correlation between tension points shown in *Figure 12* makes it hard to rely on a single framework or model to solve the problem of integrating RT processes with Agile methodologies. The researchers cannot anticipate all the challenges, chain reactions, and possible environments in their studies. However, this does not mean that targeted research on specific challenges will not reduce the problem complexity. In fact, it can be argued that is quite the opposite. It is rare for an organization to face all the RT and Agile integration challenges at one specific moment in time, if the management team went through a guided RTI process and clearly defined their business and project RT goals. As a result, the research papers such as *Antonino et al. (2014)*, *C. Lee et al. (2003)*, *Gayer et al. (2016)*, which focus on specific challenges like *Measuring Project Status*, *Costs*, *Tools*, *Worker Motivation* and other, contribute significantly to researchers and practitioners because the papers provide insights and perspectives on how organizations can solve such challenges. However, the drawback of the studies mentioned above is that the proposed solutions involve specific tools and frameworks that might not be generalized to similar organizations that use a different toolkit and processes for RT in their Agile environments. Furthermore, the above metnioned research is also more subjected to phenomena of time depreciation because the tools, frameworks used in the industry change quickly.

3.2 Research Perspectives – Combining Part I and II

The research performed in the literature study chapter provided five key perspectives on the problem of integrating RT in Agile workflow:

- The first viewpoint is that the integration of RT and Agile way of working is not only limited to the differences between RT mechanistic approaches and the Agile “just enough (flexible)” approach. Instead, the tensions to integrate the two methodologies into a business or project processes extends to the design and definition of business and project goals, stakeholders' interests, selecting the correct project methodology, project complexity, and already existing RT technical challenges described in *Chapter 2.1.5*.

- As described in *Chapter 3.1* above, the tension points are strongly correlated and can trigger chain reactions in the project processes, raising complexity without any trace for the origin point. Moreover, the symptoms of an incorrect process design for integrating RT and the Agile way of working does not manifest in the short term unless there is a small organization, project, and team.

- Many researchers have focused their studies on a particular group of challenges to analyze and solve regarding the RT integration in Agile workflow (see *Table 6*). However, no studies were found to combine and calculate the weights for the challenges based on the practitioners' perspective. Furthermore, no guidance or roadmap has been developed to highlight which RT methodologies, techniques, and challenges a manager should consider for an effective RT process in the environment where the project or business is being operated. A researched gap that is further explored in the following chapters.

- Based on the study of *Leymann & Roller (1999)*, a business process can be tagged as *Ad-hoc*, *Administrative*, *Production*, and *Collaborative*. These are categorized based on how much business value the process brings and how *Structured* and *Repetitive* the process is. The process of RT in Agile environments can be tagged as *Collaborative* because it brings high business value but is less structured and not designed to be the same for every project.

- Categorizing the RT and Agile integration process as *Collaborative* allows proceeding to the BPMa lifecycle design. However, the process of design should be supported by an appropriate BPMo language. The language selection should be based on the level of pre-structured processes. From the literature analysis, two BPMo languages have been investigated: BPMN and CMMN. This can also relate to the HCI concept mentioned in *Chapter 2.1.8,* which refers to the aspect that human interaction with the computer should be user-friendly, since otherwise, with poor UI capabilities of a tool, the analyst and process manager is prone to more errors and less motivated to further continue with the action or process.

Research Perspectives

The research overview provided above depicts the findings, existing challenges, and the research gap covered in the current thesis. This helps to frame this research and clearly visualize where a significant scientific and societal contribution can be made with the following case study. To address the challenges and the research gap described above, two new components were added to the *Figure 13* shown below. The first block is the "RTI Framework". This component aims to support the decision-makers to carefully analyze their goals and environment before developing a process to sustain RT in Agile environments. This component embeds the idea that many challenges can be addressed effectively by the practitioners if a structured RTI design approach is taken beforehand. Because such an approach will reduce the number of challenges to tackle, concrete solutions could be applied. As previously mentioned, chain reactions that can increase project complexity over time, usually come from decision-makers that might have a lack of understanding of how the tension points are inter-connected, what information should be traced, what processes should remain flexible, and other rigid. Therefore, it is argued that answering these questions beforehand should reduce the number of tensions and help develop a more structured process with the help of a BPMo language.

The data collection and analysis results described in the upcoming chapters will be used to delineate the key components and processes of the RTI framework. The framework has the primary goal to serve as RT pre-planning roadmap and provide answers about which are the key RT factors to be considered by the knowledge workers in project development phase. Furthermore, the RTI Framework will provide guidance on how RTI challenges should be balanced in an environment where there are different points of view, interests, and biases. Such guidance could be provided based on a BWM analysis conducted with practitioners for the fifteen challenges identified in the literature.

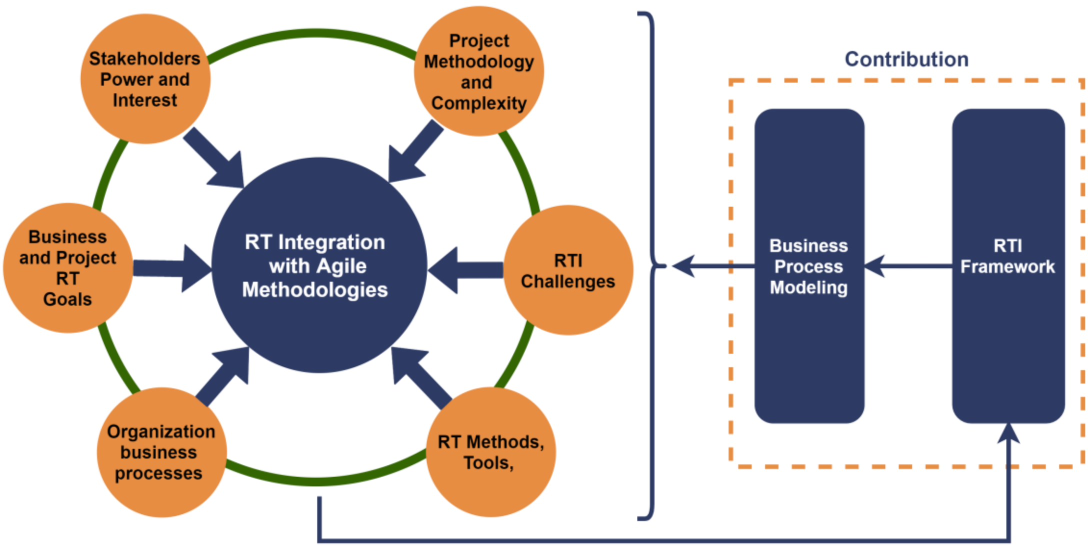


Figure 13: RT Integration with Agile Methodologies Research Contribution

The analysis and data collection performed for the RTI Framework will be directly connected to the business process management and modeling component. The RTI framework components will be translated to one-on-one processes using a BPMo language such as BPMN. Therefore, the BPMo roadmap and RTI Framework should be viewed as complementary components which are aimed to showcase the trade-offs to be made between what is desirable and feasible given limited resources and the need to stay agile.

The BPMo part of the research is explorative and will be less susceptible to generalizability compared to the RTI framework. The business modeling will be based on the processes and business goals at ABC, which represents an SME moving from the product R&D phase to large-scale production. However, certain BPMo processes like Agile methodologies (e.g., Scrum) and TDD are usually designed very similarly across multiple businesses since these are implemented based on pre-existing industry standards to receive certification (see *Chapter 2.1.8*). The *Business Process Modeling* component is believed to help further maintain, monitor, and suppress the tensions during RT integration with Agile methodologies.

3.3 Answering SRQ I

RQ I: ***What are the factors that influence requirements traceability in Agile environments?***

The current study concludes that fifteen key challenges can influence the integration of RT in Agile environments. However, it is further argued that it is of rare occurrence when all challenges would manifest at once at a single point in time. Nevertheless, if the organization has not developed any standardized business process model for RT, different teams might develop their own traceability patterns. As a result, all fifteen challenges could manifest in one organization, but these challenges are different per department and team.

Furthermore, an important observation is that these fifteen challenges cannot be separated from other tension points such as stakeholders' needs and interests, business and project RT Goals, project methodology definition, and RT techniques. All these factors are interconnected and can raise project complexity if not managed correctly. Therefore, the process of integrating RT in Agile environments cannot be viewed only through the perspective of single challenges. The factors influencing requirements traceability can be on a different level of abstraction.

4 Research Approach

This chapter will describe the research methods and data collection approach used to achieve the research objective from *Chapter 1.4* and answer the research questions from *Chapter 1.6*.

Introduction to Case Study Approach

The case study approach was selected for this research to understand better how the identified tension points from integrating RT processes with Agile methodologies might be operating in the existing ABC environment and how the challenges might be diminished. According to *Sekaran & Bougie (2016)*, case studies implicate deep, contextual analyses of comparable situations in other organizations. Therefore, choosing the appropriate case for analysis, understanding, and correctly translating the dynamics to one's situation is critical for successful problem-solving. Considering that many companies prefer to guard their proprietary data, it is not easy to find many authentic case studies *(Giuliano Antoniol et al., 2017)*. In order to protect the data of the company chosen for conducting the case study, the decision was made to use the name ABC instead of the actual name of the company. The ABC company was considered as a perfect environment for the scope of the research, which is the tensions of RT methodologies when integrated into Agile environments (R&D and T&M departments). This company is moving rapidly from the R&D focus to a more extensive scale production approach while trying to integrate the mechanical processes in the Agile way of working.

4.1 Data Collection – Part I (RTI Framework)

After the desk research on the topic of RT, data collection represents the next step to develop the theoretical framework. This chapter described the methods such as data sampling, results validation, and how the generalizability and validity threats will be counter-measured.

4.1.1 Face-to-Face Survey

The data needed for this research will be collected through a face-to-face survey (also called interview-survey), a survey method utilized when a specific target population is involved *(Sincero, 2012)*. The purpose of conducting a personal interview-survey is to explore the knowledge workers' perspectives and gather more and deeper information on the researched topic. There are several reasons why a combination between survey and interview data collection methods is the most suitable for this research. First, it is essential to mention that case studies provide qualitative rather than quantitative data analysis and interpretation *(Sekaran & Bougie, 2016, pg.55)*. The quality is achieved by researchers through the opportunity to ask questions which are more detailed, open-ended, complicated or technical *(Doyle, 2005)*. The survey data collection part of this interview-survey method represents a hard requirement for applying the Best-Worst method in data analysis.

Considering the pandemic circumstances, the face-to-face survey will be conducted via online communication channels, and the aim will be to have only personal survey-interviews. However, in the case of any technical difficulties, telephone interviews will be conducted.

Face-to-Face Survey: Advantages and Disadvantages

One of the main advantages of the face-to-face survey is the fact that the researcher can clarify the doubts of the interviewee on a particular question. In other words, the methodology gives the possibility for the researcher to follow up with another question or explanation that will provide more clarity or will allow to obtain more relevant information. Another characteristic specific only to personal survey-interviews is the big use of non-verbal clues (facial expression, body language) to understand better the interviewer's reaction or opinion about a specific topic. Usually, personal interviews are characterized by a higher response rate when compared to surveys, which is an important factor to consider when there are a limited number of potential respondents. This is also the case for the current research.

There are, of course, certain disadvantages of the personal interview-survey. First, there are mentioned the high costs of conducting personal interviews in the literature, which will be mitigated using the online communication channels. The same solution will be applied to geographical limits, which is characteristic of personal interviews. In the case of online personal interviews, there are only two downsides and namely: response bias and confidentiality difficulty. Therefore, the interviewees will be informed about the known issues of the online interviews, and the option to retrieve from the interview at any moment in time will be provided.

Telephone Survey: Advantages and Disadvantages

As previously mentioned, if there will be any technical problems to perform an online, face-to-face survey, the telephone survey will be the alternative.

The main advantages of a telephone survey are low costs, less discomfort of face-to-face for respondents, and less time consuming, which makes it also more efficient. At the same time, the telephone interview is limited to only verbal clues and has a lower response rate when compared to personal interviews.

4.1.1.1 Pre-test

To verify the comprehensibility, validity, and time needed to complete the face-to-face survey, a pre-test is required as the first step of the survey. The pre-test will be provided to 3 persons from the target group (see *sub-Chapter 4.1.2*). The feedback provided by the participants of the pre-test is then used to update the questionnaire. Then the improved survey can be used to collect data from the entire target group, such as software developers and managers of ABC.

4.1.1.2 Final face-to-face survey

After the pre-test phase, the final face-to-face survey will be conducted with the target population (see *sub-Chapter 4.1.2*). The questionnaire participants will be asked to respond in their role as employees of ABC company. All the respondents will be assured of confidentiality.

To attain the respondents, an invitation to the interview will be sent using the university email account to avoid the email being perceived as spam mail. The knowledge workers will also be persuaded to participate by briefly explaining the aim of the research and its impact on the company at the beginning of the survey, or in the email.

Furthemore, by organizing/grouping the questions logically in appropriate sections and providing instructions on how to complete each of them, will help the respondents answer the questions without difficulty. The survey will be structured in 3 parts: *Establishing credibility and rapport*, *Funneling*, and *BWM*.

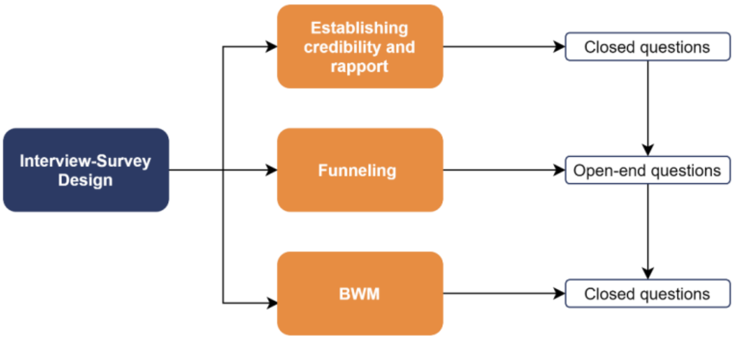


Figure 14: Interview- Survey Design

Establishing credibility and rapport

In order to gather honest answers from the participants to the interview, the interviewer should build rapport and trust with them. In other words, the researcher should be able to make the respondent sufficiently comfortable to answer truthfully without fear of any consequences. In this respect, the interviewer should state the aim of the interview and assure confidentiality. However, establishing rapport with the interviewees may not be easy. Some knowledge workers could be suspicious that the researcher in on the management's "side" (to propose a reduction of employees or to increase the workload) *(Sekaran & Bougie, 2016)*. Therefore it was decided to start the face-to-face survey by introducing the purpose of the research and how the respondent was selected to be one of those interviewed. In this regard, the respondents were informed that this survey aims to better understand the tensions faced by organizations when combining requirements traceability practices with the Agile way of working. In addition, the interviewees were told that the data collected will be used to identify and compare the literature findings with the existing practical challenges.

On the other hand, the researcher is also interested in this first phase of the face-to-face survey to obtain some relevant information about the interviewee regarding his/her experience, function, and department.

Funneling

At the beginning of a semi-structured interview, it is highly recommended to ask open-ended questions to obtain a broad idea and get some impressions about the current situation in the organization: tools, challenges, responsibilities. From the answers on the broad questions, more focused questions can be asked, as the researcher guides the respondents' through the interview and takes notes of some potential key challenges relevant to RT in an Agile environment. This transition from broad to more specific subjects is called the funneling technique *(Sekaran & Bougie, 2016)*. The technique will allow to get a better grasp on interviewee perspectives, challenges and frames of thinking.

BWM Survey

The BWM survey will be used to analyze and understand the stakeholders' interests and encountered tensions in regards to RTI challenges within their working environment. The data collected should provide insights on how practitioners rate different RT challenges, and which ones are the most important and least important to them based on the role fullfilled within a project. The results will be used to contribute to the development of the RTI Framework.

The final part of the data collection is the survey which consists of two closed questions regarding the "*most tension factor*" and the "*least tension factor*", for which the interval scale is used. This survey design was chosen to satisfy the requirements of BWM.

After choosing the most and the least tension factors, the respondent should compare the tension of the other fourteen factors regarding the most/least one. This approach is called pairwise comparison. In the designed survey, the *Likert* scale is used to examine how strongly subjects agree or disagree with statements on a nine-point scale with the following anchors: *1* refers to an equal amount of tension, and *9* refers to absolutely more tension, and vice-versa (see *Chapter 4.1.3* for more details).

4.1.2 Sampling Approach

Sampling is the process of selecting a sufficient amount of correct elements from the studied population so that a study of the sample and an understanding of its characteristics make it possible to generalize such properties to a larger population set *(Sekaran & Bougie, 2016)*. The sampling steps considered in this research are shown in the figure below.

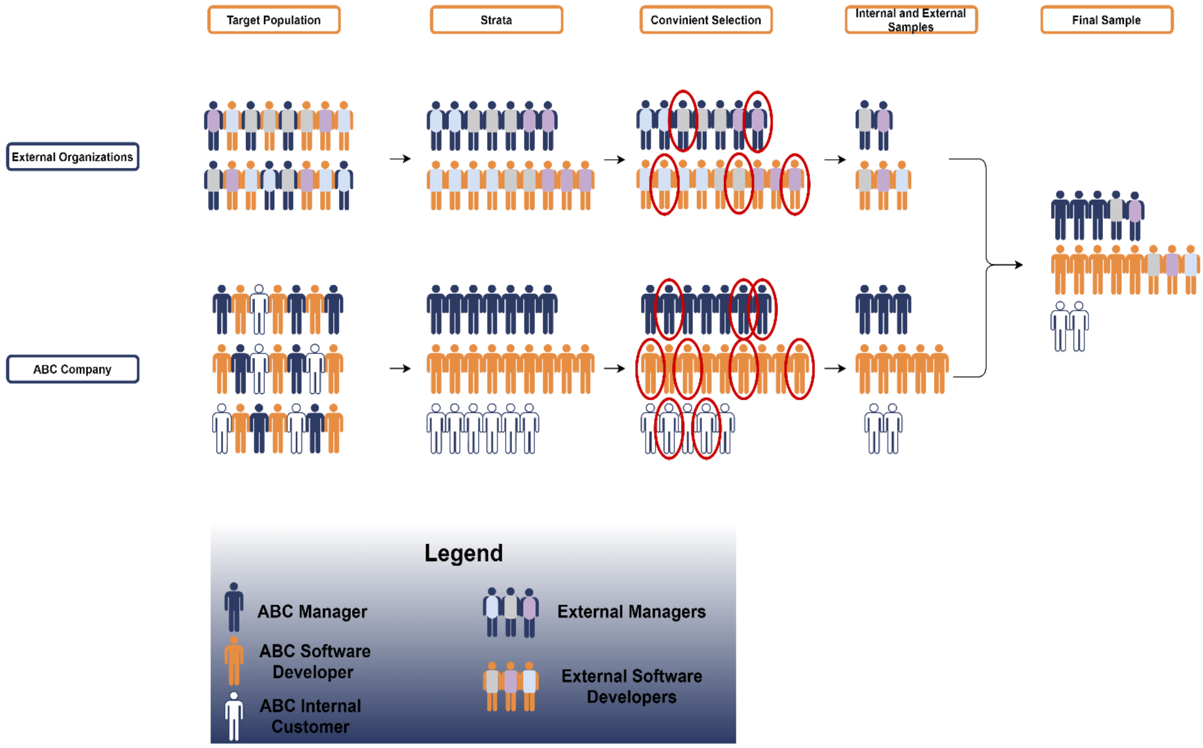


Figure 15: Stratified Sampling

Sampling begins with precisely defining the target population *(Taherdoost, 2016)*. The target population of this research consists of all the ABC employees involved in software development projects, who have the role of managers (product managers, product owners, quality assurance managers), engineers and software developers, and the customers of ABC who use internal software products. While sampling helps to estimate population parameters, there were identified subgroups of elements within the population that are expected to have different parameters on a variable of interest to the study *(Sekaran & Bougie, 2016)*. The process of stratification was applied in order to identify the groups that are relevant, appropriate, and meaningful in the context of the research. The population was divided into mutually exclusive groups: managers (product managers, product owners, quality assurance managers), engineers and software developers, and the customers of ABC who use internal software products. The convenience (non-probabilistic) sampling approach was used for selecting the representatives of each group for collecting the data, since it implies fewer rules to follow, gives the possibility to collect data quickly, and it is a cost-free methodology.

In order to generalize the study, the target population was extended outside the ABC boundaries. The intention is to add managers and software developers from three similar companies and compare them to the results from the ABC population.

Considering that the number of persons who are part of the target population is very limited (around 30) and this study is part of the qualitative research it was decided that a sample of 13 of persons would be appropriate for the *Part I* of the study.

4.1.3 Data Analysis BWM

The data collected will be analyzed using "The best-worst method" (BWM), which is a multi-criteria decision-making (MCDM) method built up by *Dr. Jafar Rezaei in 2015*. BWM is used to rank several options, which are analyzed based on certain multiple criteria. In this way, the best solution is chosen.

The benefits of BWM:

- **More reliable pairwise comparison**
  After identifying the best and the worst options, there is a precise range of the evaluation, leading to a more consistent pairwise comparison.
- **An effective strategy in mitigation of anchoring bias**
  The BWM is a so-called consider-the-opposite strategy. This means that the pairwise comparison is based on the best and worst (opposite) alternatives in a single optimization model.
- **Higher chance for a compromise on a solution**
  BWM is originally a non-linear model, which brings multiple solutions (compared to a linear model). In this way, the flexibility is increased in a group-decision making environment.
- **Data (and time) efficient method**
  The pairwise comparison methods are either single vector methods or complete matrix-vector methods, both types having their weaknesses. The main downside of the methods based on one vector only is that the consistency of the provided pairwise comparisons cannot be checked. The methods based on full matrix-vector do not encounter this problem, but there are other issues as confusion and inconsistency, which occur because of too many questions being asked. The BWM stays just in the middle with the two

vectors approach, which gives the possibility of checking the consistency and at the same time being the most data and time-efficient method.

Pairwise comparison

The goal of an MCDM is to select the most important alternative depending on a set of decision-making criteria. Weights are assigned to the criteria based upon a pairwise comparison between the criteria *(Rezaei, 2015)*. In the current research, BWM will determine the factor that creates the "most tension" within the ABC organization when it comes to integrating requirements traceability processes alongside Agile methodologies. Pairwise comparison between the RT challenges enables us to assign weights for factors in each dimension, which can be associated with the importance of the factors. The most important advantage of BWM over the other MCDM is its pairwise comparison approach, which requires less comparison, leading to higher consistency to derive the weights *(Rezaei, 2015)*. Instead of comparing between each value, first the most important and the least important factor of fifteen RT challenges identified in *Chapter 2.1.6* are determined. Based upon this reference factor, the rest of the challenges from the list are subsequently compared. The process of determining the weights of the factors in the set of RT challenges is divided into five steps:

Step 1

The set of decision criteria is determined: RTI challenges with Agile methodologies composed of 15 factors (c_1, c_2, c_3, c_4,…,c_{15}). See *Chapter 2.1.6.*

Step 2

The expert (respondent of the survey) determines the challenge from the set of criteria (c_1, c_2, c_3, c_4,…,c_{15}), which creates the most and the least tension when integrating requirements traceability processes alongside Agile methodologies.

Step 3

Respondents determine the preference of the “most tension factor” over all the other criteria (in regard of the tension it creates when it comes to integrating requirements traceability processes alongside Agile methodologies) using a number between 1 and 9 (1 refers same level of tension and 9 refers to absolutely more tension). These comparisons result in a best-to-other vector.

e.g. RTI challenges with Agile methodologies $C_B = (c_{B1}, c_{B2}, c_{B3}, c_{B4},\ldots, c_{B15})$, where B: Most tension factor

c_{Bj} indicates the expert's preference on how much more tension is created by B when compared to the tension created by j from the RTI challenges with Agile methodologies dimension (c), evidently $c_{BB} = 1$.

Step 4

Similarly, respondents determine how much more tension is created by all the other challenges when compared to the tension created by the “least tension factor” using an interval between one and 9 (one refers to an equal amount of tension and nine refers to absolutely more tension). These comparisons result in an others-to-worst vector.

e.g. RTI challenges with Agile methodologies $C_W = (c_{1w}, c_{2w}, c_{3w}, c_{4w}, \ldots, c_{15w})$, where W: Least tension factor

c_{jw} indicates the expert's preference on how much more tension is created by j when compared to the tension created by W from the RTI challenges with Agile methodologies dimension (c), evidently $c_{ww} = 1$.

Step 5

The last step implies the calculation of the optimal weights for the set of factors: RTI challenges with Agile methodologies ($w_{c1}, w_{c2}, w_{c3}, w_{c4}, \ldots, w_{c15}$). A solution can be found when the maximum absolute difference for all j is minimized for the following set (C in our case) $\{|w_B - C_{Bj} w_j|, |w_j - C_{jw} w_w|\}$ *(Rezaei, 2015).*

Formulation of the solution minmax

$\{|w_B - C_{Bj} w_j|, |w_j - C_{jw} w_w|\} j$

$\Sigma w_{Cj} = 1$

$w_{Cj} \geq 0$, for all j

This formulation can be translated to a linear programming problem: min ξ_L

$|w_B - C_{Bj} w_j| \leq \xi_L$, for all j

$|w_j - C_{jw} w_w| \leq \xi_L$, for all j

$\Sigma w_{Cj} = 1$

$w_{Cj} \geq 0$, for all j

The model above has been formulated as a linear problem, which has a unique solution. The solution to this model is the optimal weights for the tension created by the challenges of RTI with Agile methodologies ($w_{c1}, w_{c2}, w_{c3}, w_{c4}, \ldots, w_{c15}$).

For the linear model of BWM, ξ_L is considered a consistency indicator of the comparisons, and values of ξ_L closer to zero show a higher consistency level *(Rezaei, 2015).*

Reliability and Validity

Reliability refers to the consistency of the measurement. Reliability shows how trustworthy the test score is and includes two components: *stability* and *internal consistency*.

Stability of measures refers to the ability of the results to remain the same over time. For ensuring the stability part of the reliability in this paper, the test-retest method will be performed (only for a couple of interrogates). However, because of the limited time available for the research, the stability will not be tested.

Internal consistency refers to the homogeneity of the items in the measure that tap the construct. The higher the internal consistency, the more reliable the survey is. As previously mentioned, one of the advantages of BWM is that it provides the possibility to check the consistency of the provided pairwise comparisons, using ratios of the relative importance of criteria in pairs estimated by a decision-maker, from the two evaluation vectors (*Liang et al., 2020*). For the linear model of BWM, ξ_L is considered as a consistency indicator of the comparisons, and values of ξ_L closer to zero show a higher level of consistency *(Rezaei, 2015).* Reliability is an essential prerequisite of validity, which will be explained below.

A measure cannot be valid unless it is reliable, but just because a measure is reliable, it is not necessarily valid. Validity shows how a specific measure is suitable for a particular situation. If the results are accurate according to the situation, explanation, and prediction of the researcher, then the research is valid. If the method of measuring is accurate, then it will develop in valid results.

In order to ensure the collection of valid data in the survey, the pre-test will be executed *(see Chapter 4.1.1.1).* Several modifications, including clarifying and simplifying some of the languages, can be made. Extra response categories and scale items can also be added.

Generalizability

Generalizability refers to the range of applicability of the research results from one organizational setting to the others *(Sekaran & Bougie, 2016, pg.22)*. The broader the scope of applicability of the findings generated by the study, the better the research. In the current study, the scope extends to R&D and T&M departments of technical SMEs based in the Netherlands. The large scope makes this research of a very high value and usefulness to its users. For making the generalization possible, the research sampling design has to be reasonably developed. Therefore, it was decided to extend the data collection to other ABC similar-sized Dutch companies. This sampling design will allow the comparison of the results from different organizations and, in this way, it will increase the research validity and, therefore, its scientific value.

4.2 Data Collection – Part II (BPMo Roadmap)

4.2.1 Structured interview

In order to evaluate the quality of the designed BPMo roadmap, it was decided to conduct several structured interviews. Structured interviews are the most appropriate method of data collection "*when it is known at the outset what information is needed*" to collect *(Sekaran & Bougie, 2016, pg. 208)*. In the case of this study, the primary purpose is to identify how and if the designed BPMo roadmap for ABC company would help in diminishing the main challenges identified during the BWM analysis. Moreover, the interviewees will be asked for suggestions for improvements to the designed BPMo roadmap.

Taking into consideration the pandemic circumstances, the interviews will be conducted virtually but still in the face-to-face format. During the interviews, the graphical representation of the BPMo will be showed and explained to make the roadmap components clear to the practitioners.

4.2.2 Sampling approach

Considering the time constraints for the roadmap analysis, the *Convinience* sampling approach will be used. Furthermore, only three interviews will be conducted that will cover three types of stakeholders: one manager, one software developer, and one architect.

Since the BPMo roadmap was designed especially for ABC organization, all the interviewees will be employees of this company. The practitioners that will be selected for the interviews are from the same population that was used for the face-to-face survey approach described in the chapter above.

4.2.3 Data Analysis

The data from the collected qualitative interviews begins with an analysis on set of transcripts generated from the conducted interviews *(Sheppard, 2020, pg. 256)*. However, for the generation of transcripts data, a specific process will be followed as described below:

- First, a complete overview of the BPMo roadmap will be given to the practitioners.
- Afterward, several questions will be asked about the roadmap and the company processes. Detailed notes will be taken during the interviews, including both verbal and nonverbal responses.
- Finally, the notes will be read and transformed into a conclusion regarding the quality of BPMo roadmap for ABC company. Furthermore, with the help of practitioners, several methods of improvement for the roadmap will be identified. These improvement suggestions will be used as an input limitation analysis and future research discussion.

5 ABC - Case Study

This chapter will describe the results of the data collected from the interviews at ABC company and from several external companies. The data analysis was based on ten interviews rather than the initially planned sample of thirteen interviews as described in *chapter 4.1.2 Sampling Approach* The variation of three interview samples was due to the lack of responses received from the practitioners to participate in the study. Therefore, the analysis performed in the sub-chapter below is based on three ABC managers, one ABC customer, three ABC developers, and three external developers. The data collection process was followed as described in *chapter 4.1*. However, the "telephone interview" was omitted since all the respondents agreed to conduct a face-to-face survey.

All of the respondents were subjected to the same interview-survey process (see *Chapter 4.1*) with the final goal to identify how the practitioners describe their environment based on project development methodologies, what traceability techniques are used in their team, what challenges are faced with RT and Agile way of working, and in the end to define how much weight each practitioner attributes to the 15 challenges described in the chapter of literature study. The weights calculation and distribution were based on BWM steps described in *chapter 4.1.3*. The reliability of the results was assessed based on the ξ_L value and the raw data available in *Table 25* appendix.

The data analysis in this chapter provides several new insights on stakeholder roles, environment complexity, and challenges weights, which were not mentioned in the studied literature and provided a better overview and understanding of how RT should be integrated into different environments. The results from the analysis are used to build upon the concepts of the RTI framework described in *Chapter 3, Theoretical Framework*. The RTI framework can be utilized to facilitate the decision-making process for selecting the appropriate RT methodologies and techniques for ABC.

5.1 RTI Challenges and Framework – Part I

5.1.1 Product Development Methodology

At the initial phases of project development, practitioners are constantly faced with the question of which development methodology to use so that the best performance, quality, and quick product lifecycle development is achieved. In the context of the current research goal and theoretical framework, analyzing such a question is highly important. According to *Serrador & Pinto (2015)*, implementing Agile methodologies in the project has increased the satisfaction of the product owners and contributed to an overall better product at the end of the cycle. However, for larger projects, *Jørgensen (2018)* argues that the Agile methodologies have created more tension compared to the more traditional way of working. Furthermore, Agile methodologies are designed for flexibility to operate and develop the product in very dynamic environments. However, how effective such a flexible method will fit in projects where the technology and requirements are tested for decades. The product development follows a more linear approach, with COTS components and well-defined business processes. As a result, it can be argued that each project should be analyzed individually, and there is no one methodology that fits all the needs.

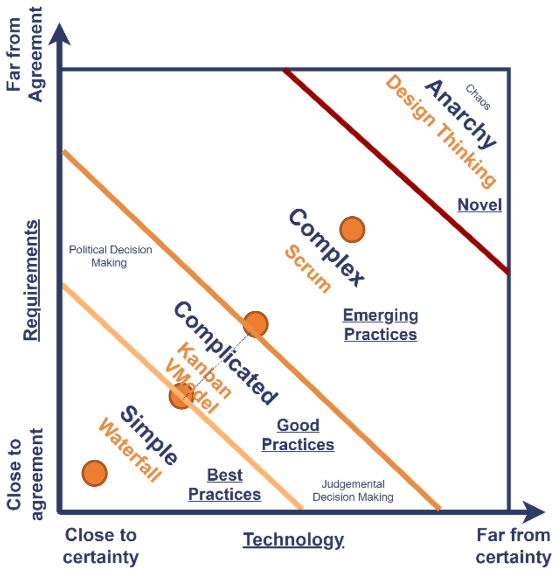
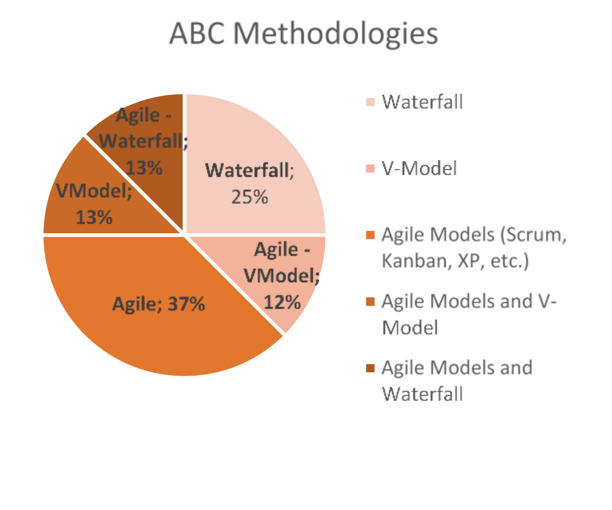


Figure 16: Complexity Analysis Ralph Stacy matrix

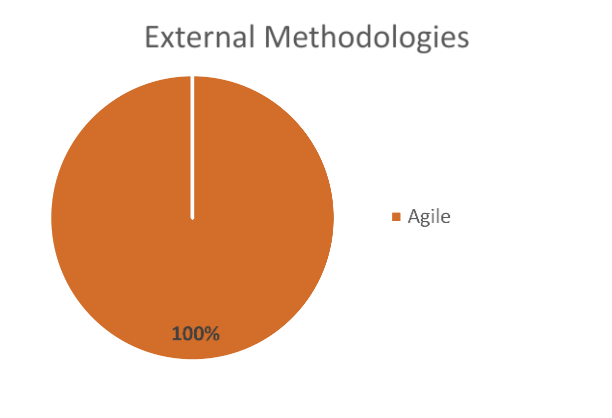
According to *Stacey (2007)*, system complexity can be analyzed through two components: "***Requirements** close to an agreement, or far from the agreement*" and "***Technology** close to certainty, or far from certainty*". Analyzing the project in these dimensions can help understand which product development methodology is the most appropriate for the project. As a result, providing the practitioners with a more robust decision-making tool to analyze and select a particular method. In *Figure 16*, Ralph Stacey matrix shows that the closer to certainty and mature the technology is, and closer to an agreement the requirements are, we deal with a more simple environment. An environment where things a more static is best suited for traditional ways of development. However, as the uncertainty increases and the requirements change frequently, the environment because more complex, and as a result, our product development methodologies should also change.

When the environment of the ABC company was analyzed by asking, "*which methodologies best represent the processes in the project the knowledge worker is involved*", the practitioners did not give only one definite answer. A vast spectrum of methodologies was used in different projects. This is represented by circles in the *Figure 16* above. Furthermore, the practitioners have stated that there is a mix between two methodologies like Agile (Scrum) with V-model or Waterfall with Agile (Kanban). An interesting finding is that the methodology used is not closely linked to the company department. Nevertheless, Agile was mentioned four times in the context of R&D, and Waterfall was mentioned twice in the context of the Manufacturing department.

In contrast, external software engineers have described their followed process as being Agile. The engineers mentioned that there is a combination of methodologies such as Scrum and Kanban depending on the project. The differences can be due to the tensions observed at ABC, where the company is trying to keep the R&D processes to further develop the product quality. At the same time, the management team focuses on business processes necessary for large-scale production and manufacturing, not only for software but also for hardware components. However, the external results were based on developers working mainly with products that were still in the R&D phase, and no large-scale production was mentioned.



Methodology (Responses)	Stakeholder Roles	Departments
Agile (3)	Software Engineer; Architect; Chief of Staff	Production; Test and Measurement; R&D
Waterfall (2)	Quality Assurance; Software Engineer	Production; Test and Measurement; Manufacturing;
Agile and Vmodel (1)	Software Engineer	Test and Measurement; R&D
Vmodel (1)	Project Manager	Test and Measurement;
Agile and Waterfall (1)	Project Manager	Manufacturing; Test and Measurement; Production



Methodology (Responses)	Stakeholder Roles	Departments
Agile (3)	Software Engineer;	Production; R&D

Table 14: Environment Complexity and Project Development Methodologies

The results from *Table 14* slightly change the perspectives adopted in the theoretical framework (*chapter 3.2*). The initial thought was the each SMEs have unique requirements for their RT. Therefore, they can usually adapt their processes with a more Agile way of working and deploy the design for multiple projects in different departments. However, *Stacey's (2007)* study shows that this does not always represent the best decision because the complexity varies, and with the complexity the product development methodology should change. But the most important factor is that internally at ABC, the requirements agreement or technology certainty can vary so much per department and project that each team may have unique traceability goals and require different traceability characteristics. But as previously mentioned, the focus of the study is on the Agile environments within organizations. From the data analysis is possible to observe that the Agile departments are mainly R&D and Production. Nevertheless, slight variations exist for specific projects in both departments towards a more Agile and V-model or Agile and Waterfall approach.

5.1.2 Goal Definition

In the *Theoretical Framework* of the study was mentioned that the lack of RT goals within an organization or project could lead to chain reactions that can increase the complexity in the long-term to very high levels. It can be argued that rarely a practitioner will weigh equally, at a single moment in time all the RT goals mentioned in *Chapter 2.1.4*. Therefore, to design a better RT business process, the practitioners should prioritize their traceability goals.

From several observations made at the ABC, it was possible to conclude that a clear goal definition for RT was never agreed upon among practitioners who follow the same interest and work in the same departments. Furthermore, by analyzing the answers to the open-ended interview questions, it was possible to remark that the employees are looking towards a **Configurable, Trusted**, and **Portable** system. The interviewees mentioned that a significant amount of tension in their work is created by the fractured tools across the organizations and that there is a "[1] *lack of a full system that integrates everything"*, and that the "[2] *approvals for certain requirements change, takes time and is not efficient*".

Therefore, it can be debated that if the decision-making process for the RT practices contained the negotiation rounds to define the key traceability goals then, a more structured point of view would have emerged among stakeholders, and a common system would have been adopted in the early stages of ABC product development. And as a result, the business goals of ABC could also greatly benefit from a systematic approach to requirements and the tool used to support the business operations.

5.1.3 Understanding Stakeholder Needs

Several research articles which were analyzed in the literature study conclude that the needs, interests, and constraints faced by employees in their role as a software developer, manager, analyst, and others, are of key importance when the traceability processes and tools are changed or new ones are introduced (see *Chapter 2.1.7*). Furthermore, the analysis completed in the theoretical framework showed that stakeholders could directly influence increasing project and RT complexity if their interests are not met, or the RT processes do not support their workflow, which varies per role.

This sub-chapter examines the interests and needs of the stakeholders at ABC when it comes to RT and the Agile way of working and how these relate to those found at external companies. The gathered data analysis should further emphasize if the employees' workflow matches the one described in the literature and if the challenges faced by different roles are caused by an incorrect assessment of traceability needs and inaccurate modeling of business processes based and human biases (see *Chapter 2.2.2*).

Before proceeding to the challenges and interests of stakeholders in RT and Agile, some preliminary questions were asked to understand the interviewees' knowledge base about the concepts such as Agile, RE, and RT. From the results shown in *Table 15* below, it is possible to see that the actors on average rated their knowledge on the main topics slightly above average. However, an interesting observation is that the interviewees with a lower value for *years of relevant experience* assessed their understanding of particular topics with a score on average of two to three points higher compared with interviewees who have more experience in the field. Furthermore, the practitioners have generally agreed that Agile methodologies efficiently respond to change quickly, promote communication with the team, and deliver *higher customer satisfaction*. Although, contradicting results were found in the literature studies of *Beck et al. (2001)*, *Serrador & Pinto (2015)* for the topics of *increasing team members' motivation* and being *people-centric rather than process-centric*. Practitioners have neither agreed nor disagreed with the topics mentioned above, but Agile methodologies, on the contrary, are highly advocated to promote such processes and values.

The compiled results on knowledge assessments provide an insight that the practitioners are not biased towards a methodology or practice and assess their knowledge as balanced, but with plenty of room to learn and explore new concepts and theories that probably are still unknown. Moreover, such equitable results on the knowledge assessment bring the opportunity to explore further the challenges faced by practitioners.

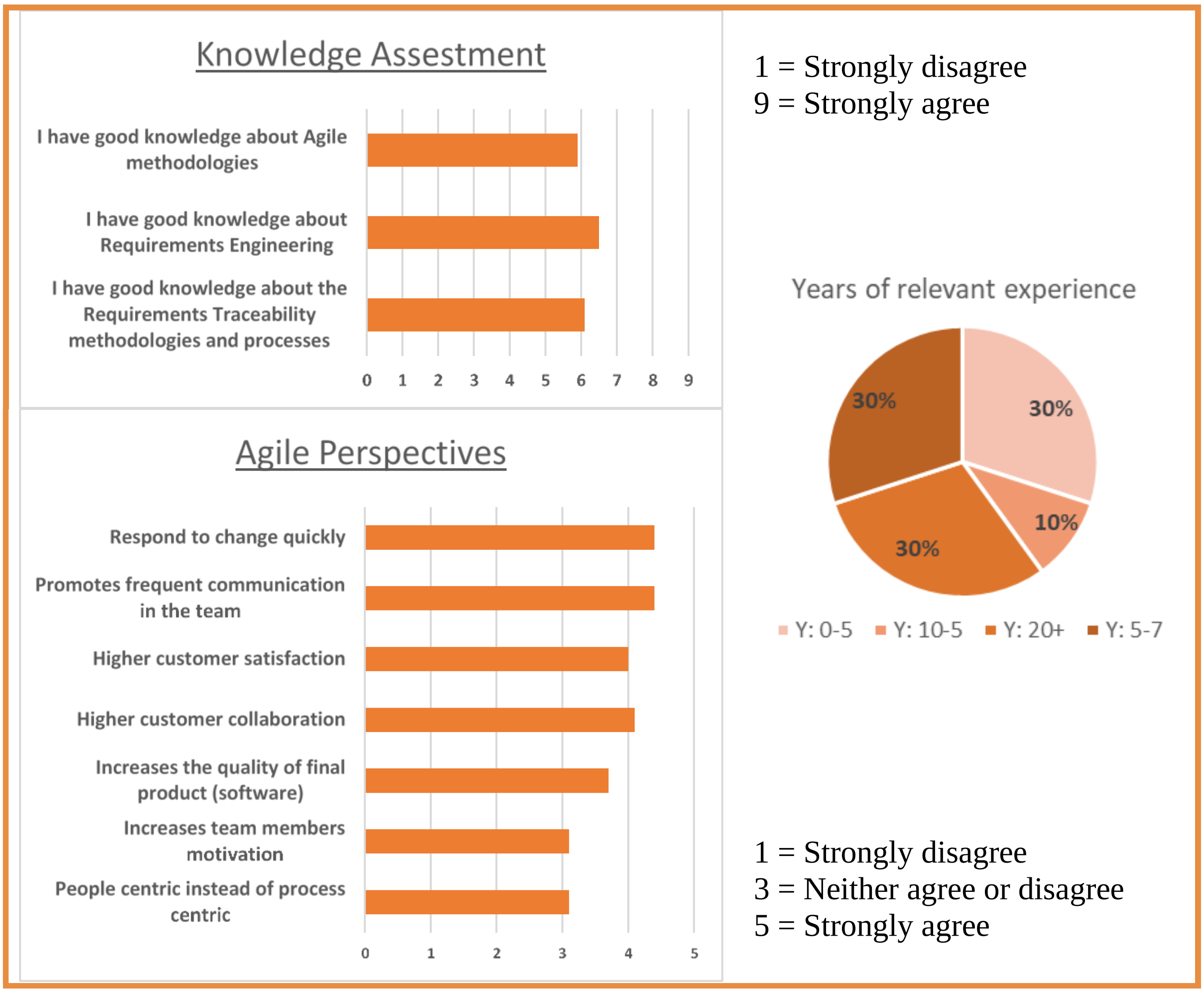


Table 15: Interviewees knowledge of main thesis concepts

To better understand the interests and constraints faced by stakeholders at ABC and at the external organizations, a set of questions were asked about *what challenges/problems the organization faces when it comes to integrate or maintain the RT processes alongside the Agile way of working*, or if they are familiar with any *active projects currently within the company for improving the RT integration with Agile way of working*. To further explore gathered data, the practitioners' perspectives were divided based on two roles: **project managers** and **software developers**.

Managers Perspective	Software Developers Perspective
[Project Manager] Increased and hard to manage backlog due to constant changes in the product requirements since a big part of product design is still in the R&D phase. [Project Manager] There is no product owner to keep track of requirements and no person to translate the customer requirements to functional and non-functional requirements. [Project Manager] There is no process and responsibility assigned to do prioritization of the requirements. [Quality Assurance Manager] There is no process or tool implemented to help with impact assessment analysis on a requirement change. Therefore, no visibility is ensured, and system quality is diminished. [Customer] The requirements capture process is missing entirely, and there is a lack of a tool to monitor the status of the requirement.	[Internal] The requirement capturing tool is missing; if the requirements were changed at some time, the blame is set on the developer for not understanding the requirement. [Internal] The frequent requirement change rate overwhelms the requirement traceability log and makes it unusable. This decreases the traceability appeal to the team members, and the process ends up being avoided by the engineering staff. [Internal] A tighter control on requirements change at the start of the project will be beneficial. However, few processes have been implemented to support such a change. [External] Rapid succession of scope changes can lead to muddied traceability as a result and decrease the product quality. [External] No standard chapters for documentation, any architect and developer can document the project in its own way. [External] Even if all the employees followed the same course for Agile (green belt). Every person insists on their own way to implement the Agile process and use the RT tools in a different way and process.

Table 16: Managers and Software Developers Perspectives on RT and Agile Challenges

The compiled challenges from the table above clearly distinguish how different roles within the project focus on different workflows and are interested in different RT parameters. Therefore, the results resemble a close match with the studies by *Alaa & Samir (2014)* and *Cleland-Huang (2012).* Managers are interested in prioritizing requirements and ensure a structured backlog. The quality assurance actor is interested in impact assessment and requirements information retrieval. The customer is focusing on product quality but has no visibility in the case above on the requirement status. On the other hand, the developers are overwhelmed by the traceability log since they are interested in tracing acceptance criteria, viewing assigned open requirements, and performing development.

Table 16 provides two more insights on the RT in dynamic environments. Firstly, the missing processes of RT at ABC create tension between different stakeholders' roles. As mentioned by an internal software developer, the missing process of monitoring requirements might result in misunderstandings and arguments between different roles. As there is no description of the actual functional or non-functional requirement, there is no party to be made responsible if a requirement was not delivered as expected. The second insight comes from an external developer that emphasizes that even though the employees follow the same training to implement Agile methodologies and RT practices, they still view the business processes modeling and implementation differently from the rest of the colleagues. From these findings, it can be argued that product development methodologies and RT practices should be accompanied by semantic-based solid modeling languages that define a clear implementation process. Nevertheless, such results increase the validity of the study by *Razavian et al. (2016)* on how cognitive biases negatively impact the BPMa lifecycle.

An interesting observation is that several interviewees mentioned some key challenges related to RT integration with Agile methodologies before introducing them to the BWM analysis. Some examples are "frequent requirement change control", "impact analysis after requirement change", "worker motivation to trace the requirements," and others. A strong correlation was observed between what was initially described by several practitioners and what later was selected as the "most tension" and "least tension" in the BWM survey. It can be argued that it gives further validity to the challenges defined in the literature and what practitioners experience in their daily work.

When asked about the existing solutions to address the RT challenges, the managers had more extensive knowledge on the topic. They mentioned integrating several tools, switching from a fractured tool base to a more structured one and others. However, the developers were less aware of upcoming changes and proposed solutions. This points to a lack of communication between different roles- a tension and a challenge that is separate from the process of RT integration in Agile environments but directly influences and increases the complexity of integration. It can be argued that these observations further validate the theoretical framework, which emphasizes that the tension points strongly correlate, and the process of RTI should be investigated as a wicked problem as described and defined by *Rittel & Webber (1973)*.

5.1.4 Traceability Characteristics

RT is a complex process that should be managed within organizations to increase product quality and customer satisfaction. Even though RT as a process has been implemented in many organizations successfully over the years, the scholars are still faced with a set of RT challenges such as: developing more accurate information retrieval models to help with project analysis, introduce flexibility and adaptiveness in RT tools to improve the HCI, support the management team with query parameters that give a better overview on the impact assessment on requirement change, and others (see *chapter 2.1.5*). However, until an RT tool is developed that fits all the organizations' needs, each management team is responsible for carefully designing their RT business processes to support their product development. This requires that the actors understand traceability methodologies and techniques and set boundaries to what is essential to their business goals.

To assess how practitioners tackle the RT challenges and understand what parameters and techniques are used in their daily operations to monitor requirements, an open-ended question was asked about *the organization's current tools and methodologies for requirements traceability*? From the received responses, it was clear that the practitioners felt more comfortable discussing the tools rather than starting a conversation about requirements traceability techniques, traceability links, and what granularity level is actually important to trace in their current projects. Despite the above-average results achieved for the selfassessment on the *RT methodologies and processes* in *Table 15* above, the interviewees were more conservative to elaborate on the above-mentioned topics. However, many practitioners did mention the workflow of RT in their projects. The RT flow usually starts from the product datasheet specified by the customer. Then the requirements are discussed and translated into additional functional and non-functional requirements. These requirements are then stored in different tools, and each project manager is responsible for translating them and manage their product backlog. The workflow described by practitioners only refers to one type of RT, which is *Forward Tracing*, mentioned in *chapter 2.1.2* and described in reasonable detail in the *Cleland-Huang (2012, pg. 20)* paper.

Nevertheless, one interviewee noted that an RT technique used is RT matrices. However, a problem faced is that the requirements specifications do not match the tasks necessary to accomplish the requirement. That very little traceability and analysis is done on the topic.

Another interesting insight on the topic of RT in Agile environments is provided by the *Figure 17* shown below. The graph provides an overview of all tools used at ABC to perform RT and monitor product development status. The toolkit mentioned by each practitioner highly correlates with their role. For example, software engineers at ABC noted that their traceability tools are Jira, Git, Confluence, Gitlab, and Asana. On the other hand, the project managers and product owners referred to PDM, Smartsheets, Arena, and Asana. Such a fractured overview of the tools can represent a symptom of the long R&D period the ABC processes were subjected to. Nevertheless, bringing the tool to a more standardized package should start with several negotiation rounds between different teams to compromise. A long-term operation with a fractured eco-system can drastically decrease the organization's product quality and ambidexterity.

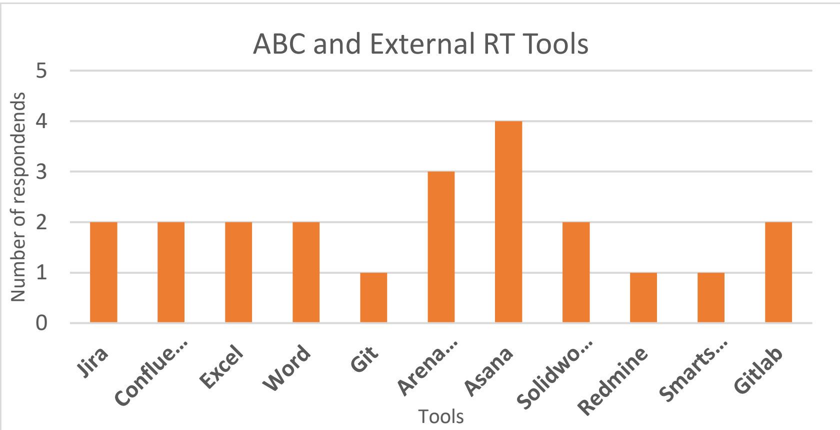


Figure 17: RT Tools at ABC and External companies

5.1.5 BWM Weights Analysis

This chapter will address one of the research gaps mentioned in the theoretical framework and mainly analyze and describe the weights for the most tension and least tension RTI challenges in agile environments based on the practitioners' perspective. As mentioned in *chapter 2.2.3*, no analysis was found on the topic in the studied literature. The calculation of weights (w_{c1}, w_{c2}, w_{c3}, …, w_{c15}) and the ξ_L value was done based on the 5 step process described in *chapter 4.1.3*. The sum of weights calculated for each interview satisfies the conditions: $\Sigma\, wCj = 1$ and $wCj \geq 0$, for all j.

The approach to analyzing the BWM results will be based on the stakeholders' roles as defined in the literature study, *chapter 2.1.7*. The reason to take such an approach is based on the goal to understand how different stakeholders rate the RTI challenges based on their roles. The *Figure 18,* provides the first view on which challenges the practitioners think to create the most tension when integrating RT in Agile or flexible environments.

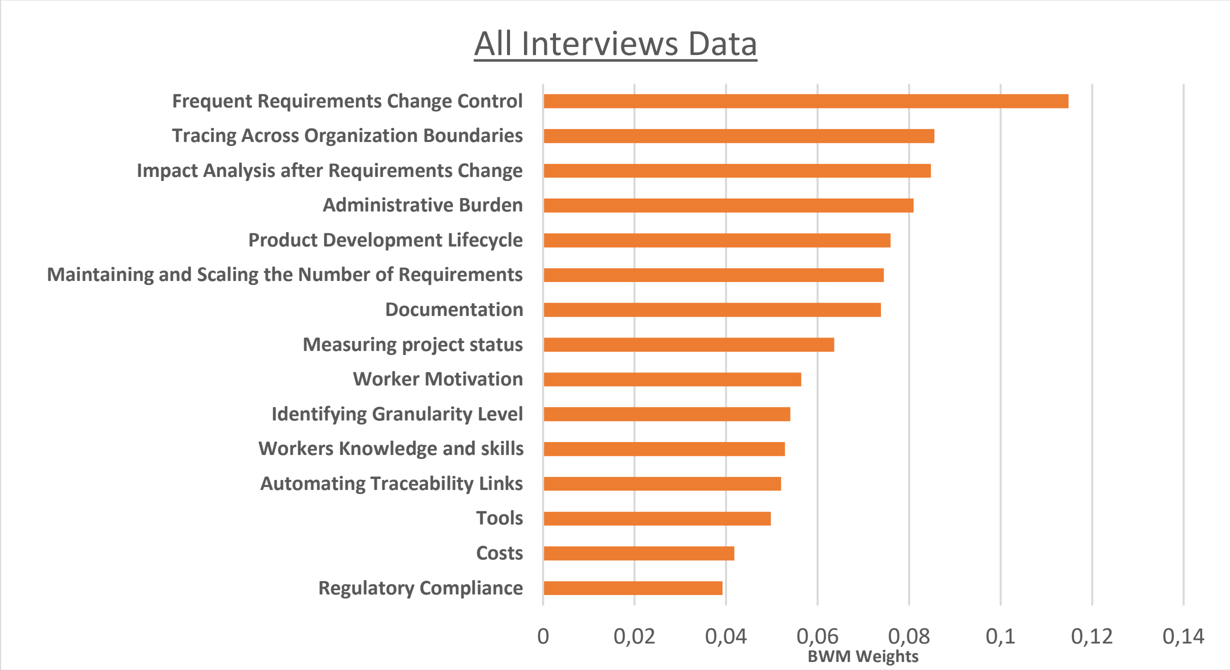


Figure 18: RTI challenges weights based on all interview samples

The practitioners have rated the "*Frequent Requirement Change Control*" as the most tension challenge. The challenge is described in the literature as the ability to keep pace with constant and frequent requirement changes of the projects. Such a result is expected since the challenge was mentioned several times during open questions during the interview process. Several studies have tried to address a group of challenges, including the frequent requirements change witnessed in Agile environments. The tools such as *TraceMan (Extended)*; AMME; Echo, and other mentioned in *Table 7*: RT Solutions for Agile Environments describe that automated task status updates and the TDD approach should reduce the tension of the challenge. However, as mentioned in the theoretical framework, such solutions might address the symptoms but not the actual problem. The actual problem, for instance, can come from an external tension point unrelated to RTI, such as lack of communication between different roles within the organization or lack of knowledge workers to maintain and regulate the requirements flow. During an interview process, an interviewee mentioned that sometimes an employee should take the shoes of 2 or 3 roles (e.g., project manager, developer, and analyst). Therefore, it can be argued that to address such challenges, a broader perspective should be taken into consideration as the first step and then address the problem through different automation tools. This argument can be further supported with the low weight achieved for the challenges of *tools*, *automating traceability links*, and *costs*.

As mentioned at the beginning of the chapter, the current sample differs from the one planned during the research approach design. The final sample contains four stakeholders with managerial roles and 6 with a developer role. Therefore, *Figure 18* does not accurately express the weights given by different roles to the RTI challenges. To address this issue, the data was divided into the management group and the developer group, as shown in *Figure 19* and *Figure 20*. The first very noticeable difference between the two groups is represented by the weight of the *Documentation* challenge. For the managers at ABC, the documentation creates the most challenge when it comes to integrating RT into their projects, and a weight of 0.125 was calculated. However, the developers assigned to documentation a weight of 0.04. The 3x difference between groups on a particular challenge showcases how important it is to understand such divergences and develop a proper decision-making process to prioritize which challenges to be addressed first when developing a business process for RTI.

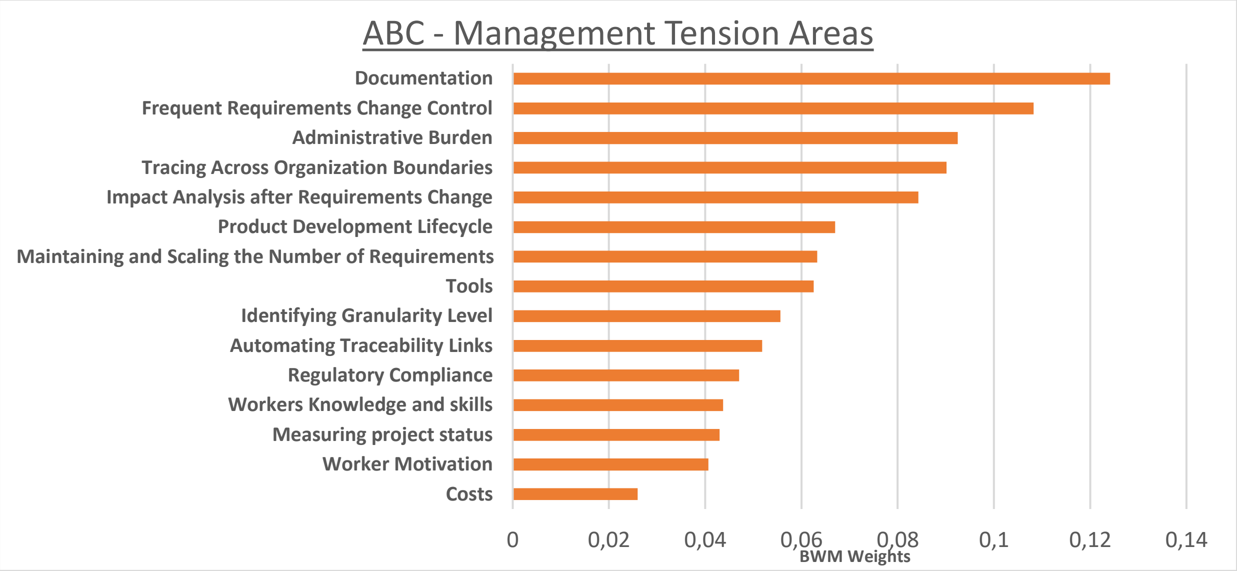


Figure 19: Management perspective on RTI challenges

Nevertheless, the *Frequent requirements change control* is rated by both as a challenging process that creates a lot of tension in their workflow.

An anomaly in the results can be noticed in *Figure 19*, related to the *Measuring project status*. Such a low weight is unexpected because the challenges mentioned by practitioners during the open-questions interviews should also create tensions to accurately measuring the project status, yet this is not the case. Having said that, in studies conducted by *G. Antoniol et al., (2006)*, *Curcio et al. (2018)*, it is mentioned that failing to measure the project status accurately can lead to different design failures during the project development. Nevertheless, the studies point that in practice, the status attribute of every requirement is not used and is ignored by most stakeholders. This can be further explained by the fact that R&D processes and ways of working still prevails over more structured and mechanistic processes required for large-scale product manufacturing and operations.

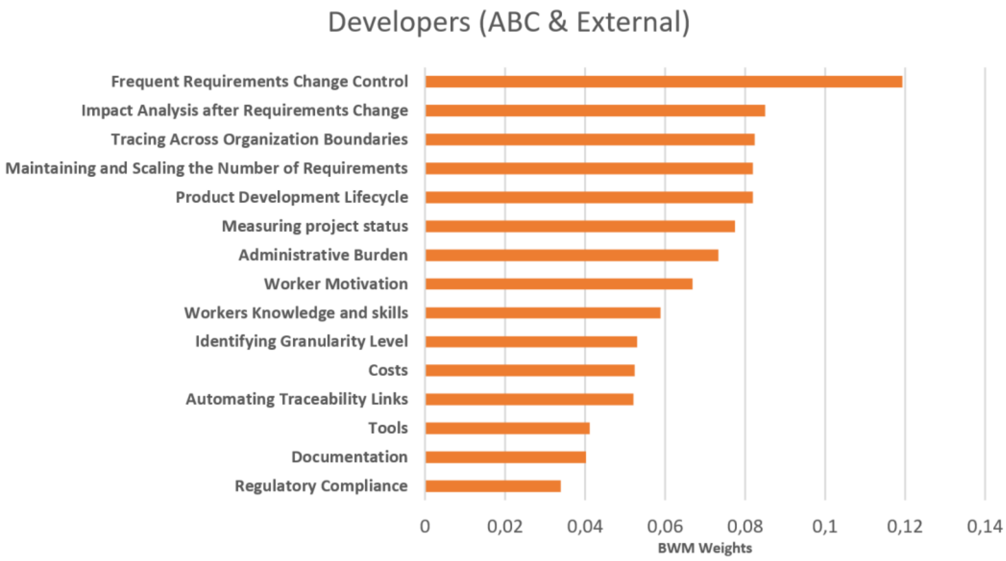


Figure 20: Software Developers perspective on RTI challenges

5.1.6 ABC Challenges – Summary

The ABC challenges discussed above can be summarized in several bullet points as described below:

- Until a certain point, there was no specific methodology used to define the product development methodology. As a result, the management team does not have at their disposal the correct tools to help define a standardized process for a particular project complexity.

- The Agile methodologies are not seen in the company as being people-centric, but rather more process-centric. However, most employees agreed that the method helps to respond to change quickly and promote communication within the team. Nevertheless, as in the case of project complexity, the organization lacks a clear roadmap on how to add Agile to certain projects and the process to be followed.

- The company is still working on defining the correct processes that will make a person responsible for prioritizing requirements and taking the role of the product owner for specific projects.
- The ongoing work to define roadmaps for product development and define the process to assign responsibility for requirements prioritization directly influences the workflow and productivity of the software developers. For developers, the *frequent requirement change* and the lack of standardized processes were mentioned to overwhelm tracking the traceability log. Furthermore, because there is no person responsible for requirements in certain projects, the blame is placed on developers for misunderstanding the requirements. Therefore, from developers' perspective, tighter control on requirement change is required.

- Another challenge is the lack of standardization for writing the documentation among software developers. Even though certain guidelines were created, every developer can define a preferred tool and structure to write the documentation. However, it should be mentioned that on the top level, the requirements specification documents, system design documents, and others follow a more rigorous control and templating. On the other side for certain R&D projects, such control is missing.

- The complexity increases for managers and developers by the fragmented set of tools used across the organization. Version control systems such as git are combined with tools like Jira and Gitlab that both are operating and serving the same goal. The same follows for different aspects of documentation (e.g., Confluence, Word, Google Sheets), which do not have a common place for storage and naming convention. The product backlog and requirement tracing (Arena, Asana, Smartsheets, etc.)

- From the collected data, the practitioners have not defined a set of goals for the traceability systems that will satisfy the business values and mission. Even if the management team defined the goals, these are not properly communicated throughout the organization. As a result, this leaves many employees wondering why some processes are necessary.

- The BWM analysis has shown that there are also several conflicts between stakeholders on prioritizing the RT challenges that should be tackled first.

The challenges will go as input to the business process modeling. The non-executable will be used because of the existing problems. Semi-informal and formal language because it provides a more rigid environment to understand the process flow etc.

5.1.7 RTI Framework

The data analysis performed in the above sub-chapters brings the current research to address the next research gap discovered in the literature study. The gap refers to the lack of a framework that can be used to understand which tension points a manager or any other stakeholder should consider for an effective RT integration in Agile environments. Furthermore, there is no explicit representation of the process practitioners should follow to define possible tensions in their environment clearly. Therefore, with the help of the data analysis performed in the chapter above, and the research conducted for the literature study, an RTI framework is proposed, as shown in *Figure 21* to support the decision-making process of practitioners when it comes to RTI.

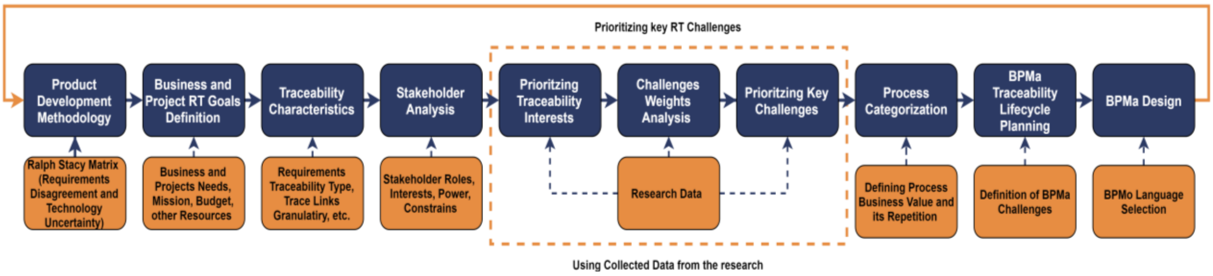


Figure 21: RTI Framework

The process flow shown in the RTI diagram closely resembles the flow used to perform the data analysis in the case study. The variables analyzed in the data analysis chapters were correlated in the theoretical framework. The same variables (tension points) mentioned in the theoretical framework were used to collect key information from face-to-face interviews to understand better the RT integration challenges in an Agile and flexible environment such as the one at ABC. The results from data analysis have shown that by analyzing in a systematic and structured way, the tension points discussed in the theoretical framework can help identify the root cause of several RTI challenges and point to other tension factors that can indirectly influence the process of RT. This point of view is also supported in a study by *Rempel et al. (2013, pg.1),* which mentions that "*…the overall quality and mismatch of analyzed traceability suggests that an upfront-defined traceability strategy is indeed required. Furthermore, we show that the decision for or against traceability relations between artifacts requires a detailed understanding of the project's engineering process and goals*".

Stakeholders Interests and Traceability

In *chapters 5.1.2* and *5.1.5,* the data analysis supported the findings in the literature regarding the stakeholders' roles and their different workflows and interests when it comes to RT. Furthermore, the BWM analysis has pointed to a new tension factor in the project development. Stakeholders with different power in a project had given equal weights to different RT challenges. For example, the developers ranked as high tension the *frequent requirement change control (weight = 0.12),* whilst the managers have as the most tension factor the *documentation (weight = 0.12)*. Therefore, in such scenarios, which RT challenge should be addressed as a priority when the time and budget is limited by the organization? A dichotomous answer to this question cannot be given. However, in such situations, the stakeholder interests can be used as a foundation factor to start the negotiation process. For example, one of the developers' workflows and interests is viewing assigned open requirements, whilst the managers are interested in having a structured product backlog and documentation. Therefore, common ground can be reached by using web-based documentation that can combine both interests.
Another point of view that can be taken is to ask what causes such high tension for *frequent requirements change control* and *documentation*? The analysis of this question can lead to identifying the third hidden factor, which is the causes of both challenges. The root cause analysis of the problem can be identified using the Five Whys technique developed by Sakichi Toyoda, and as described in detail by *Serrat (2017)*.

In the paper by *Cleland-Huang et al. (2014, pg. 59)* is mentioned that "[1] *Traceability research must be driven by the needs of its stakeholders, who ultimately adopt tracing solutions*…[2] *there is little prior work that examines the specific needs of the stakeholder in the traceability process and, as a result, academic researchers are left making assumptions about industry needs*". The developed RTI framework in part I of the study, is founded based on the conception to take in the early stages of RT business process development the stakeholders' needs. The steps such as the "Complexity Analysis", "Goal Definition", not only help to define what tracing techniques are the most important but also aid to correctly categorize the business processes for further RT processes modeling (see *chapter 2.2.1*). Nevertheless, the RTI framework contains the main components to be a valuable guideline for practitioners to define a map of higher and lower surface tension points for RT integration in different types of environments. However, for other types of organizations, new weights calculations should be performed.

Business Process Modeling Phase

The last three stepts of the RTI Framework are composed of *Process Categorization*, *BPMa Lifecycle*, and *BPMa Design*.

The *Process Categorization* step helps to set the correct perspective on the *as-is* and *to-be*-developed process. From the ABC analysis, it was concluded that the organization's *as-is* processes are *Collaborative*. However, there is low *structuring* and *repetition*, which causes tensions points for *frequent requirement change control* and *documentation*, which as a result, increases the *administrative burden* with no added value. Therefore, ABC is currently aiming for a more structured and repetitive process that can be used in multiple cross-functional teams.

The next step in the RTI Framework is *BPMa Traceability Lifycle Planning*. After categorizing the *as-is* and *to-be* processes for RT, the practitioners have to identify the key steps in the product development operations that are inefficient, missing, were designed with no particular goal, or have been the result of some cognitive biases as described by *Razavian et al. (2016)*. In the case of ABC, it was possible to conclude from the interviews and on-site observations that most of the company processes were developed with the initial goal to sustain the basic operations of R&D activities. However, with the company's growth, the established processes cannot support the required standards by the customers, industry, shareholders, and legislators.

The BPMa design is the last and relatively complex step in the RTI Framework. If there is no standardized *business process description language* adopted in the organization, this would be when the practitioners have to make a clear selection. The language selection for BPMo should be based on the process categorization, goals, and the complexity to be described by the language. As described in *chapter 2.2.3*, the language can be formal, informal, or semi-formal. Furthermore, the BPMo can be executable or non-executable. For ABC, the answer for the language that should be used might vary based on the department (e.g., production, R&D). If to look from the R&D perspective (as defined in the research scope), the organization should aim for a non-executable language that can be transformed into executable processes. Having a flexible language will ensure a more tension-free adoption across the company

5.1.8 Answering SRQ II

RQ II: ***How does a framework aiming to enable requirements traceability and reduce tension in the context of Agile environments look like?***

The framework capable of reducing the RTI tensions in Agile environments has seven building blocks aimed to help the practitioners approach the RTI in a structured and strategic way.

From the designed framework, the first step to RTI is to understand the complexity of the project or environment in which the integration should take place. The complexity should be analyzed based on the level of agreement or disagreement for the requirements, and based on technological certainty or uncertainty to be used in the project. This first component in the framework forces practitioners to decide on the correct project development methodology to use (Agile, V-Model, Waterfall, etc) (see *Chapter 5.1.1*). The next step is about understanding the project and the organization's traceability goals. For instance, the business goals can vary from developing a *portable* RT system or a more *cost-effective* RT system. Clearly defining the goal will help understand which traceability methodologies and techniques are the most appropriate for the practitioners' circumstances (see *Chapter 5.1.2*). In this way, it can be avoided that a decision-maker must consider all the fifteen RTI challenges at a single moment in time. As a result, the practitioners should start thinking about trade-offs in the early phases of project planning.

The next step in the framework is to understand the stakeholders' needs and interests in traceability. The literature studies, and the performed data analysis, have clearly shown that the stakeholder roles in traceability are of key importance. If the implemented RT business processes do not support the stakeholder workflow, the practitioner will avoid the process altogether, or will try to circumvent it. Therefore, a framework that can enable requirements traceability in Agile environments should contain a table defining which traceability challenges are of key importance for each stakeholder role. For example, if to consider that two stakeholders, with different roles, are conflicting because of an equal amount of weight given to two different challenges, then the RTI framework provides a negotiation step where the manager should take into consideration the stakeholder's interests and workflow.

The last step of the framework is to help practitioners to design a business process model that will support RT in Agile environments. The framework uses the input from previous steps of the framework to define the process categorization (e.g. *Ad-hoc* or *Collaborative*). As a result, this would help to determine the proper business process modeling language to create the design. The implementation of the business process model should be guided by the BPMa lifecycle and the RT goals defined in the initial steps of the framework.

The processes that introduce RT in Agile environments should be adapted periodically based on the external environments and the changing needs of stakeholders. This is also shown by the feedback loop in the RTI Framework from *Figure 21*.

5.2 Roadmap Design – Part II

Based on the conducted literature study, theoretical framework, BWM data analysis, and the *RTI Framework*, a BPMo model for ABC will be designed In. This chapter describes the aspects of ABC Business Process categorization, ABC traceability lifecycle planning, and the business process modeling using the BPMN language. Moreover, the BPMo design shown in *Figure 22*, represents an extension for the RTI Framework steps. Therefore, both diagrams should be viewed as complementary rather than separate components to solve RTI problem.

It is essential to mention that the following chapter does not attempt to develop an executable model using BPMN. The language will be used as a semi-formal method to communicate the processes that can be implemented at ABC in order to sustain a good BPMa lifecycle in the context of RTI framework and Agile way of working. The model will be based on the literature study, information gathered during the interviews, survey, and BWM data analysis.

5.2.1 RT in Agile - BPMo Design

The RTI Framework described in *Chapter 5.1.7* provides concrete steps to reduce RT tensions in Agile environments. However, the framework should be translated into business processes using a BPMo language. Nonetheless, before designing the BPMo roadmap, it is important first to analyze the ABC environment from the business process perspective. Such an approach will help to understand what processes (e.g, Collaborative, Production, Ad-Hoc) are better suited for ABC, and the challenges encountered in the organization when maintaining the business processes lifecycle.

Furthermore, the process mapping for ABC is also affected by the slight variation of role definition compared to different literature studies. Each organization defines the roles slightly differently in the project. For example, at ABC, a product owner can also fulfill the role of the system architect.

5.2.1.1 Process Categorization and Business Processes Lifecycle

As already mentioned in *Chapter 2.2.1*, according to Leymann & Roller (1999), the processes can be divided into four classifications: "*Collaborative*", "*Production*", "*Ad Hoc,*" and "*Administrative*". Each category describes how structured or ad-hoc a specific business process is and how much business value it brings. From an organizational perspective, any improvement in employee productivity, product quality, or reduction of costs are considered aspects that can bring high value. ABC company has a large base of processes ranging from structured ones in the production environment to the more ad-hoc and less mapped processes in the R&D department. However, the organization has many processes that are constantly being subjected to change due to many uncertainties that come from technology development. Furthermore, business processes considered at some stage rigid and structured can become unsuitable in a short period of time due to new challenges/variables that were not accounted for at the beginning of the process development. The constant change in the business processes resulting from the uncertainties of design innovation leaves many knowledge workers unmotivated to spend time mapping (developing) business processes that might change quickly. Therefore, many projects in the R&D department at ABC give very little attention to process development. Instead, the focus has been shifted to product development. As with any organization that for many years stayed in the R&D phase and heavily relied on external investments, the focus is to break even and release a functional product to the market that meets the customers' specifications.

Nevertheless, shifting a lot of resources in product innovation and development, and leaving behind process mapping and design might not be the fastest way to increase the organization's value. Paying more attention to processes running within the R&D department at ABC could allow managers to take a more top-view approach and observe inefficiencies that, if addressed, can speed up the product development/innovation. However, more literature analysis and interviews should be conducted to arrive at a more definitive conclusion on the topic mentioned above. Still, due to a limited time frame, this was not entirely further researched.

The lifecycle described in *Chapter 2.2.2* represents the idea that the business processes usually cannot be designed to be static for an extended period of time in any organization. Therefore, there should be a constant overview and analysis of the existing business processes to achieve optimum performance within any company. However, even if from a theoretical perspective, following routinely the business process lifecycle steps can increase the business value and contribute to better product quality, in practice, the lifecycle might not be followed rigorously. In a rapidly changing environment such as the one at ABC, process management can require many resources. Usually, changes in the business process can require a lot of time to be implemented. Furthermore, such changes might manifest their benefits in the long term rather than the short term. Time, however, can represent a key resource that might not be available, especially when the market moves quickly and the competition is very high. Nevertheless, the business process lifecycle diagram is shown in *Figure 11, Chapter 2.2.2*, which starts with analyzing the existing "*as-is*" processes before a new re-design. However, the current thesis does not analyze "as-is" processes in great detail due to the lack of resources and time. Still, the observations and interviews from Part I are used to creating a probabilistic overview of the "*as-is*" processes.

In conclusion, it can be argued that process mapping should be approached with diligence in dynamic/ad-hoc environments where resources are limited and time-pressure to market is high. However, in such scenarios, practitioners could rely on the industry best practices that will serve as a good foundation to propel the organization when it has more resources and time to design and standardize the business processes in different departments. For example, for software engineering, such best practices can be version control, standardized documentation, and scheduled peer-review. At ABC, more emphasis had to be placed on the best practices in software, hardware, team management, design innovation, and product development lifecycle at the early stages of the company foundation. Designating specific roles or sub-teams responsible for implementing and researching industry best practices can be an initial first step with high impact. However, as the organization grows more prominent, the complexity to implement changes also rises drastically because of multiple stakeholders' interests. Therefore, even if the management team would act in everyone's best interest to promote industry best practices, a lot of attention should be given to stakeholders' interests/power and to the "negative stakeholders" mentioned in *Chapter 2.1.7*. Furthermore, through interviews and observations performed during non-formal discussions with the ABC management, more transparency should be made on the business goals and business process modeling lifecycle. Overall, the employees reflected that a better overview has to be provided by the top management on the company strategy and how specific projects can help achieve the company goals.

Consequently, this would increase employees' motivation and allow them to contribute better to design innovation and product development life cycle processes.

5.2.1.2 BPMN Language Selection

As mentioned in *Chapter 2.2.3* and briefly in *Chapter 5.1.7*, mapping the business processes requires selecting a specific modeling language. In the literature study chapter (see *Chapter 2.2*) it was mentioned to choose a BPMo language, a designer/manager should pay attention to three key aspects: *Process Categorization, BPMa Challenges* and *BPMo Language Criterias*. As described in the chapter above, the processes at ABC vary per department. But if to closely analyze the R&D branch of the company, the processes that are followed are either missing, are *Ad-Hoc*, or *Collaborative*. When it comes to business process management lifecycle, the practitioners focus mainly on product development and take less into consideration the business process management for several reasons mentioned earlier in the thesis. Therefore, considering the company approach to process management, there are several essential principles to consider. From *Table 10* in *Chapter 2.2.3.1*, it is possible to identify five key criteria that might facilitate the BPMo language integration at ABC. These criteria are ***Expressibility, Low Complexity, Ease of Learning, Tool Support, Universality, and Formality*.** However, it is important to mention that no data was gathered from employees to identify these criteria. These were only based on the observations and the interviews/survey conducted in *Part I* of the thesis. Therefore, the actual requirements of ABC employees for a process modeling language can slightly vary.

After an analysis of available literature, no study was identified that compared the Flowchart, BPMN, and CMMN languages on the above-selected criteria. Nevertheless, the exclusion method can be applied to estimate an appropriate modeling language for ABC from the ones researched in *Chapter 2.2.3*.

It can be argued that **Flowchart**, even though it satisfies most of the above properties, the language lacks expressibility when it comes to describing certain event-driven business use cases. When it comes to *Ad-Hoc* or *Collaborative* processes, the event-driven approach to design business processes can be an essential criterion for a well-defined and explicit mapping. The **CMMN** language was designed by OMG to be case-driven and address the business processes that are ad-hoc. But the language lacks in terms of *Universality, Formality* and *Ease of Learning*. Furthermore, certain market players such as Camunda have also dropped their tool support for the language *(Deehan, 2020)*. Therefore, a language that overall could satisfy the majority of criteria is **BPMN**. According to a study by *Pereira & Silva (2016)*, the BPMN has scored the highest on the criteria mentioned in *Table 10* in *Chapter 2.2.3.1.* However, in the *Pereira & Silva (2016)* article, the BPMN was compared to languages such as *EPC, UML-AD, RAD and IDEF*. The grade given by the author from 0 to 5 was based on the support or lack of it for the abstract criteria discussed above. Furthermore, the grading process is not the most accurate because it is based on the author's perception of the language's capabilities and the other scientific articles describing the languages. In addition, the author didn’t gather any data from practitioners to support its claims of the BPMN results. Nevertheless, the scholars and practitioners overall agree that BPMN is a universal language for business process modeling which was largely adopted by practitioners in many industries.

Therefore, because the BPMN satisfies most criteria and represents a largely adopted language by practitioners, it will be selected as a language to map the RTI process at the ABC Agile environment.

5.2.1.3 Role definition

In *Chapter 2.1.7* of the literature study, different stakeholder roles were analyzed, and their unique interest in traceability was described (*see Figure 8*). The division between each stakeholder role was clearly delineated and their potential input or tasks during the project development. However, in practice, a slight deviation from the theoretical definitions of roles was observed when studying the ABC environment. At ABC, a single knowledge worker could take the responsibility of several roles, such as the ones of a *Developer, System Architect,* and sometimes even *Product Owner*. Furthermore, the role definitions are not set in stone for every single project. As a result, mapping the business process for the RTI has an additional challenge because the role definitions do not vary per department but often vary per individual project.

Such a designation of roles at ABC can be due to three factors:

- Firstly, the organization does not have a clear categorization of roles within the project. Furthermore, at the beginning of each project, in many cases, there is no pre-planning done for assigning an employee a specific role. As a result, the lack of a clear definition of roles often creates confusion about who should be responsible for particular requirements or release dates. It is important to mention that the researched departments at ABC were mainly R&D, where employees usually face more flexibility in product development methodology. Nevertheless, because of the lack of a clear process to define the project methodology and the lack of a process to assigning responsibilities through roles, the development of the products is less efficient and often creates a lot of miscommunication between the teams and the customer (internal and external).

- Secondly, the project (to be developed) might be relatively simple and does not require a whole team working on its components. Therefore, a single employee is assigned responsibility for the deliverable and fulfills most of the project roles. Such an approach can prove effective as long as the employee is not responsible in the same manner for several other deliverables/projects. However, at ABC, one employee is involved in multiple projects at the same time. As a result, the quality of requirement traceability, project documentation, and industry best standards are not followed accordingly.

- The last factor is due to the lack of resources. For example, a single knowledge worker might sometimes be assigned multiple roles because there are not enough resources to support the project with a larger team.

Regardless of why a single knowledge worker is fulfilling multiple roles during a project, there are still mechanisms that, if implemented, can improve product development lifecycle and requirements traceability. For example, creating and assigning the role of a *Reviewer*, can help to ensure that certain company standards for traceability, documentation and other industry best standards are followed. There can be two types of reviewers:

- An internal reviewer who knows the details of the project and besides the standards can also advise on system architecture.

- An external reviewer only ensures that the requirements traceability practices are followed, the documentation is updated, and the correct template is followed.

In conclusion, it can be argued that ABC should pay close attention to follow a specific product development methodology (e.g. Scum), use a common tool base, and follow a process to define different roles within a project, and assign particular responsibility to those roles.

5.2.1.4 Process Mapping (BPMN Overview)

This sub-chapter describes the BPMo diagram designed to improve the requirements traceability practices in an dynamic/ad-hoc environment at ABC and increase the efficiency of the product development lifecycle. The selected language for process mapping and design is BPMN, as described in *Chapter 5.2.1.2*.

The BPMN roadmap was developed using the RTI Framework and the data analysis performed in Part I, combined with the study of ABC processes described in Part II, as shown in *Figure 22* below. The BPMN roadmap follows the building blocks of the RTI Framework. The roadmap is created from a single pool called *ABC - R&D Department.* The *Pool* is further divided into several swimlanes such as *Product Owner*, *System Architect* and *Software Development Team*. The swimlanes, as described in the literature study, represent key roles for achieving the business goal. In this case, the business goal is to integrate RT in an Agile environment and increase product development KPIs at ABC. Furthermore, the *Software Development Team* swimlane is divided into the following roles *Scrum Master*, *Software Developer* and *Reviewers*. It is important to mention that even though one employee can have multiple roles at ABC, it was decided not to combine the roles in the BPMN roadmap. It is easier later to combine the roles in another BPMN design when these are clearly divided in the roadmap. Furthermore, having separate roles in a project is a methodology considered a best practice. This is exactly what BPMN roadmap is developed to convey.

The following subparagraphs provide an overview of each group of processes described in BPMN roadmap. These groups represent a translation of the RTI framework into business processes that are believed to help ABC improve its internal efficiency for requirements traceability and product development.

The reader is advised first to analyze *Figure 22: ABC BPMN Roadmap* ***to better understand the paragraphs below.***

Requirements Validation

The process starts with a message event coming from the customer (internal or external) which requests a particular product or component of a system to be developed. This initializes the first task for the product owner, which is the *Collect and Define Project Requirements*. This task represents a sub-process in which the steps of requirements elicitation, specification, and elaboration are completed. Then the process moves to the system analyst or architect, which has the technical knowledge to *Define the Validity of Requirements*. In the next step, the process divides into two paths:

- If the requirements are not validated, the architect informs the product owner, who further forwards the message to the customer. This route resets the process in the initial phase

- On the other hand, if the requirements are validated, the project can start and move to the next phase of *Project Development Methodology*.

Project Development Methodology

In this part of the process, the product owner, with the help of the system architect, should define the appropriate methodology for product development. First, the system architect should *define the project's complexity* using the Ralph Stacy matrix and clearly state the technology certainty/uncertainty and requirements agreement/disagreement (see *Chapter 5.1.1*). After this task, the product owner has the key information about the project complexity and is responsible for selecting the appropriate methodology for the project. The methodology can be Waterfall, V-Model, Agile (e.g. Scrum, XP), or a combination of both. However, in practice, the product owner might base their decision on previous experience, which does not guarantee the best project outcome.

Nevertheless, at ABC the management team in the R&D department will most likely choose the Agile or a mixed approach of development.

Goal Definition

The goal definition phase starts with the decision to *Identify Business Traceability Goals*. If these were already defined within the organization, the product owner should start with *Identifying Project RT Usage Goals* using the *Requirements Database* file.

The *Business Traceability Goals* refers to the goals discussed in *Chapter 2.1.4, Table 21*, and mainly describes the key characteristics the company is looking to have for its traceability systems and other tools. For example, initially at ABC, a tool called Redmine was planned to be introduced for RT. The tool was open-source had a lot of configurability and the requirements for scalability. However, Redmine was never introduced in the company because it lacked support from other departments, did not completely match the workflow of the projects, and finally, did not have a clear team responsible for its implementation. From the above-mentioned Redmine characteristics, it is possible to deduct the *Business Traceability Goals*, which were important for ABC. These can be summarized as *Cost-Effectiveness, Configurability, Scalability, Valued and Purposed.* In the case of Redmine, the tool didn’t fit the Valued and Purposed goals because it didn’t support the workflow of the employees and was seen to increase the administrative burden. Therefore, the product owner needs to understand the business goals before proceeding to the project implementation.

The *Project RT Goals* refer to safety, Regulatory Compliance, Industry Compliance, Customer satisfaction, and others. Defining project RT goals helps the management team answer the question of why the traceability is done or why the planned project must be developed.

The following process task (step) is the *Analysis of existing software development processes & Tools* that are aimed to support the business and project RT goals. The system architect should perform the analysis, which has two possible routes:

- If there are no tools aimed to support the defined goals, the system architect should start a research process to define the best tools available on the market that will align with the established RT goals. After the analysis, the task moves to the product owner, who starts the complex sub-process of implementation. However, it can be argued that the research and implementation should not be part of the product development process. But rather, a separate sub-team should be delegated that will focus entirely on the analysis and provides a set of tools for the company to use.

- Suppose there is already a set of common agreed tools for RT within the company that supports the main business and project RT goals. In that case, the product owner should start to focus on the next phase of the process, *Traceability Characteristics*.

Traceability Characteristics

The traceability characteristics phase starts by identifying the key artifacts/requirements that should be traced in the project. The next task for the product owner identifies the RT types such as forward, backward, horizontal and vertical traceability, as defined in *Chapter 2.1.2* of the literature study. Afterward, the system architect should take responsibility for suggesting tracing methods such as manual, semi-manual, or automatic tracing (see *Chapter 2.1.2).* However, the tracing methods are also often imposed by the capabilities of the selected tools for the RT. The last two tasks represent the responsibility of the product owner to identify the *traceability technique* (RTM, integration documents, etc.) and finally develop a *traceability information model (TIM)*.

In the context of the BPMN roadmap, the TIM refers only to categorizing requirements into groups so that the information about specific requirements is easier retrieved from the database. For example, the project requirements can be categorized based on the functional purpose, like safety, privacy, reliability, etc.

After defining all the technical details of the RT, the product owner moves to another key phase in the roadmap and mainly *Stakeholder Analysis and Team Definition.*

Stakeholder Analysis and Team Definition

In the stakeholder analysis phases, the product owner and system architect have four key actions that should be accomplished:

- Product Owner: Create the team requirements based on the available budget, complexity, and customer requirements.
- System Architect: Create the team with the necessary competencies based on the provided requirements.
- Product Owner: Inform the team about the start of a new project.
- Product Owner: Identify the key interests of each role within the project. As described in *Chapter 2.1.7* of the literature study, the knowledge worker's interests in traceability vary based on the project role. Therefore, an analysis of the role and interest should help the product owner organize the RT workflow within the project correctly.

These actions will ensure that the team designated for the project has the right competencies and create the base for the team structuring necessary for the next steps.

Prioritizing Key RT Challenges

This phase of the BPMN roadmap is about addressing three key factors:

- What is the process to Research and acknowledge existing RT challenges within the company
- Understanding the stakeholder's interests and their usual way of working within the projects.
- How to negotiate with the team members on a standard workflow and define which RT challenges to tackle first if specific conflicts exist.

The phase starts with the task of *Analyzing existing RT Challenges*. This task is aimed to force the product owner to do a pre-check and identify what went good or bad in the previous projects and what can be done better in the new projects.

The next task is to *request a workflow flowchart per each role* involved in the project. If the workflow is known from previous projects, this request can be omitted. As the other team creates a flowchart with their workflow, the product owner uses the already researched and weighted BWM identified challenges from the Part I of the ABC case study and identifies conflicting RT challenges between different roles. For example, the *Documentation* challenge is weighted by managers with a value of ~0.13 and represents the most tension factor. For developers, on the other hand, *Documentation* is only weighted with ~0.04 and is at the bottom of the list. This represents a conflict RT factor that should be addressed by the product owner in the first place.

The last tasks for the product owner are gathering available workflows from the roles, identifying common workflow patterns, and starting the negotiation process with the team members on which RT challenges and issues to address first so that the administrative burden for all the involved parties is not increased.

Even though the phase of *Prioritizing Key RT Challenges* might sound like a burdensome process to follow, its practice can be made simplistic and take only one meeting, given that workers' workflow is known by the product owner beforehand or from experience.

RT and Agile way of working

The last step of the BPMN roadmap represents the description of the Scrum workflow. During this workflow, all the knowledge workers fulfilling a certain role use the requirements database, sprint backlog, and project RT goals to release a working product to the customer (see *Figure 22* below).

In addition to the industry standardized Scrum workflow, the role of the Reviewer is introduced as described in *Chapter 5.2.1.3 Role definition*. The reviewer meeting is scheduled in the BPMN roadmap on a weekly basis, given that the Sprints take more than one week.

The processes described in the last phase of the roadmap represent a contribution to the ABC organization because it gives a clear view of each action taken by each role involved in the project. Furthermore, because ABC is still in transition to a more standardized way of working (e.g. following concrete Scrum steps), the BPMN roadmap provides a clear overview how the roles should be designated in a project. Moreover, the roadmap workflow can be adapted to satisfy the needs of teams in different projects, not only in the R&D department.

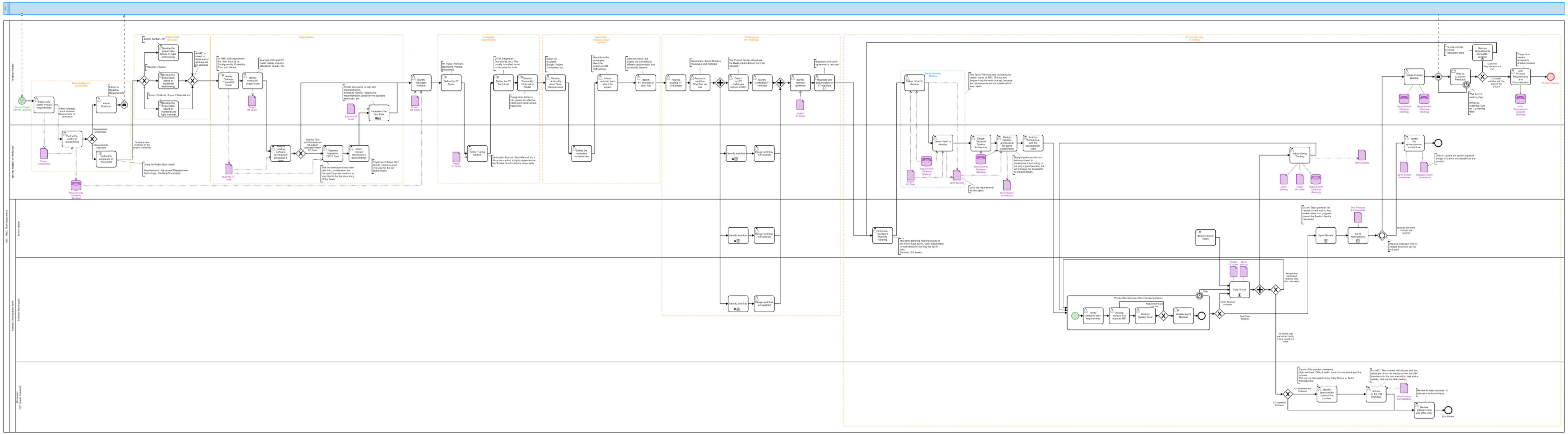


Figure 22: ABC BPMN Roadmap

(Please see the separate attached document for a higher resolution diagram)

5.2.2 Answering SRQ III

RQ III: ***How will a business process model look like in the context of the developed framework?***

The components of the business process model aimed to integrate RT in ABC's Agile environment closely resemble the RTI Framework steps. Each step from the framework, such as defining the *Product Development Methodology*, *Traceability Characteristics*, *Stakeholders Interests*, and others, were translated into specific business process components in the BPMN roadmap. Furthermore, in each step of the roadmap, every stakeholder responsibility has been clearly delineated per role. As a result, this allows the customer and other parties to identify who is responsible for the requirements database, project backlog, quality assurance, meeting planning, system integration, testing, and other parts of the project. It is argued that the project managers should initially spend time in project pre-planning and develop a clear definition of tasks. Even though such an approach might increase at the beginning the administrative burden, in the long-term, the knowledge workers are encouraged to develop well-defined project templates that will increase flexibility while maintaining industry best practices.

By analyzing the *Figure 22* above, the management should make trade-offs between what is desirable and feasible given limited resources and the need to be ambidextrous. The BPMN roadmap approach should still be considered Agile. The decisions for negotiations and trade-offs must be made before any Agile methodology is followed. As a result, because of the strategic planning required at the beginning of the roadmap, the knowledge workers can maintain the best practices for RT and Agile way of working. Moreover, each step of the BPMN roadmap was analyzed and discussed in detail in the literature study chapter. The existing literature has detailed answers on some of the steps discussed in the roadmap. As a result, the practitioners from other organizations can also use the same roadmap, and are encouraged to investigate further what will be the best alternatives for them, given the available information.

Moreover, before designing any business process, a specific language should be selected by the management team. For dynamic environments such as those found at ABC, the focus should be on a BPMo language that fulfills six key criteria: *Expressibility*, *Low Complexity*, *Ease of Learning*, *Tool Support*, *Universality*, *and Formality*. From the studied modeling languages, the BPMN satisfies the majority of the criteria. Moreover, during the interviews conducted at the ABC, practitioners have shown no difficulties in understanding the workflow and task division described by the BPMN roadmap. The BPMN also has the advantage of being translated into executable models for more *Production*-based processes at a later stage.

5.3 Roadmap and Framework Evaluation – SRQ IV

RQ IV: ***Is the developed framework and business process model evaluated as detailed and complete?***

This chapter provides an analysis on the practicality of the BPMN Roadmap and the RTI Framework as viewed by the practitioners at ABC.

The interview results, transcribed in detail in *Table 27* in the Appendix chapter, provides a clear overview of the practicability of the designed framework and roadmap. The interviews were conducted based on the approach described in *Chapter 4.2,* and the results will be generalized and further elaborated in the sub-paragraphs below.

RTI Framework

The practitioners at ABC found the RTI framework useful because it gives a good perspective on the steps to be taken to develop better project planning. According to practitioners, following the framework could generate a checklist of RT challenges and processes to which the managers should pay close attention. One of the interviewees specified that the RTI Framework "*gets you thinking about different aspects that go into project planning, and provides a good overview on the process*".

Improvement Suggestions and Limitations

The framework also had several suggestions for improvement. All the interviewees found the RTI Framework to be too generic and too theoretical. One of the developers mentioned that the framework "*is a perfect picture on how the things should be*. *But in reality, there should be a person (or team) responsible for the quality assessment of the processes. [Furthermore], often because of lack of budget, time and practicality reasons certain steps (from the diagram) might be offset (omitted)".* Another project manager at ABC mentioned that "*I do think the roadmap is useful. But it has to be templated, so it does not cause overhead*". Through the word "*templated*," the project manager referred to a limited set of best practices for each step in the RTI framework that practitioners can use choose from. These best practices will have to be frequently designed based on the organization's needs. As a result, the employee will not have to go through the whole research process each time a new project is started but instead will select certain processes in each step of the RTI Framework from a limited option list.

Overall, the RTI Framework is viewed as complete, as it closely resembled some processes in the past where the organization tried to integrate a new tool. Nevertheless, the practitioners are more interested in the detailed processes and strategic planning described in the BPMN roadmap. Because it provides more details and an accurate overview of the tradeoffs to be made in order to integrate RTI in ABC environment.

BPMN Roadmap

From the interviewees' perspective, the BPMN Roadmap should help generate more accurate RT. The roadmap forces all the roles involved in the product development to follow a process to maintain the project backlog, update the project status, and write documentation while the product is still in development. According to a practitioner, "*very often at ABC to force developers to write documentation is a big challenge*". The task of writing documentation at

ABC is usually left at the end of the project. As a result, the documents never get to be written because a new project knocks on the door. Another interviewee emphasized that BPMN roadmap should "*provide a good foundation to have regulatory compliance (e.g. ISO 9001) and the process described in the roadmap is good for supporting the business/project goals*". Thus, from the interview data is possible to conclude that the roadmap could reduce the RTI Challenges such as *Documentation, Updating Project Status, Product development lifecycle,* and others.

Further in the interview process, practitioners also mentioned several **drawbacks** and possible improvements to the roadmap. Overall, the drawbacks identified in the diagram during the interview process can be generalized to three aspects:

- The BPMN Roadmap takes only the perspective of the external customer. With external customers, the requirements can be locked in the initial phases, and the development process can start from there. However, at ABC, the knowledge workers often deal with the internal customers that provide the software/hardware requirements that could change daily. Therefore, the BPMN Roadmap does not provide a process to introduce new requirements in the database during the *Sprint run* or even *Daily Scrum*.

- The BPMN Roadmap is too generic for a certain task which can require a lot of effort/research to complete. More detailed steps should be identified, like selecting from a list of predefined tools, selecting from templates for requirements categorization, etc.

- At ABC, the requirements should often be traced between different layers such as photonics design, hardware (electronics), and software. Unfortunately, the BPMN Roadmap is missing the linkage between different requirements databases.

The main challenges to introduce BPMN Roadmap at ABC were identified to be the people and the administrative burden. One of the interviewees mentioned that the first thing employees would ask is, "*Will this work in practice?*", "*Is it complex*?". Furthermore, the practitioners raised the concern that some employees would prefer to stick to the old workflows, and the main argument could be "*that their current workflow worked for many years, why should they change?*. Moreover, the BPMN Roadmap is only applicable for projects which are of "*sufficient size and budget. Otherwise, you can waste ~10% of the project's total time to go through this roadmap*". As a result, such processes implementation would not attain the necessary "*critical mass for the things to move forward*".

In conclusion, the overall BPMN Roadmap sentiment from the practitioners at ABC was positive. The process mapping, however, as with any complex deliverable, requires several iterations before producing a satisfying output. Due to the limited time and resources available for the thesis, it was not possible to iterate and implement the feedback suggested for the BPMN Roadmap. However, the improvements proposed by the practitioners represent findings that can be used as a foundation for future research.

6 Research Limitations

This chapter describes the limitations associated with the process of the data collection, data analysis, the developed framework, and the business process modeling roadmap.

6.1 Limitations – RTI Challenges and Framework

There are several threats that should be discussed in the context of the developed framework. The first threats are associated with ***data sampling***, ***data collection,*** and ***analysis***:

- The initially planned sample (as described in *chapter 4.1.2*) was composed of thirteen practitioners, from which two were intended to be external managers and two internal ABC customers. However, due to a lack of responses from practitioners, only ten interviews were conducted. As a result, there was no sample of external managers as initially planned, and only one ABC customer was interviewed. Therefore, the BWM weights assigned to the RTI challenges might not represent the entire researched population. As a result, this can also threaten the validity of the RTI framework.

- The BWM method used to identify the RTI weights was composed of fifteen challenges (criteria). A value far from the recommended maximum of nine criteria in the studies. Therefore, to have more representative results for such a large number of criteria, a larger sample than ten interviewees had to be selected. Another measure could have been to perform a longitudinal study to see if the interviewees' perspectives do not vary over. However, because of the limited amount of time allocated for this research, such a test could not be conducted. Therefore, the stability of the results can represent a threat to the study validity.

- The fifteen challenges identified in the literature study were used in the face-to-face survey to collect data and calculate the BWM weights. The interviewees had to compare the challenge that creates the most tension and the least tension compared to other challenges. However, because of the large number of criteria and the complex concepts that come with RT and Agile methodologies, many practitioners were not very confident about how to compare the challenges between them. This was also observed in situations when they had never experienced a particular challenge in the past. Moreover, the interviewees were often jumping in different time frames when comparing the challenges. For example, did they have to focus on what tensions they face at the moment, which ones they used to have in the past, or which ones will occur in the future? Clear instruction on the time frame was not given, and therefore the results are not representative of a specific time period. However, it was mentioned that it should be focused on the ABC environment.

- The current study cannot fully guarantee the reliability of the measurements. As defined in the data collection *Chapter 4.1.3*, the reliability is measured through two key properties: *Stability* and *Internal Consistency*. To measure stability, a test-retest procedure had to take place. However, because of the lack of time available from the practitioners and for the thesis development, a re-test of the data collected for BWM was not possible. Nevertheless, the BWM, as described by *(Rezaei, 2015),* makes the comparisons "*in a structured way, which makes the judgment easier and more understandable, and more importantly leads to more consistent comparisons, hence more reliable weights/rankings*". The value of $\xi_{L\ (Ksi)}$ is considered as a consistency indicator of the comparisons. Therefore, the condition $\xi_L < 0.1$ was analyzed for every survey. And because in all the cases, it was a guided survey, the interviewees tried to keep consistency in their comparisons. However, the optimum value for ξ_L for 15 criteria was not calculated due to limited time.

- Another limitation can be the non-probability sampling was used to collect data. The population has been divided into strata (project managers, developers, customers, etc.). However, a disadvantage of non-probability sampling is that it is difficult to know how well the population was represented. The ABC population is relatively small. Therefore the sample can be considered representative. Nevertheless, the several interviews conducted externally for generalizability do not guarantee that it accurately represents the challenges of all SMEs with similar environments as the one ABC, R & R&D department. As a result, the fifteen challenges identified during the literature study and the results of the BWM can have issues of generalizability.

Generalizability and Transferability

The first problem of generalizability is that the data sampled from the external companies is composed only of developers. Managers were not available when interviews were conducted. As a result, due to the limited time available for the current thesis, the decision was made to continue only with the developers' data.

Figure 23 and *Figure 24* provide an overview of the RTI challenges at ABC and at the external companies. It can be observed that the weights for the *Documentation* and *Regulatory Compliance* challenges remain low in both cases. This can be explained by the similarity in the developers' workflow and because of their RT interests during the project development. As mentioned in the *chapter 2.1.7*, the developers focus mainly on viewing open requirements, engineering, and providing implementation status updates.

In addition, the challenges of *Frequent requirement change control* and *Impact Analysis after requirement change* have relatively high scores in the BWM results. Such correlation can be explained by the flexible, Agile, and sometimes uncertain environment where the developers execute their work. Since the data collection was performed at organizations of similar size as ABC, and the interviews were conducted with the employees working in the R&D departments, some results show a close resemblance. Moreover, the practitioners have also generally agreed that *Frequent requirement change control* can lead to a lack of power on the product backlog. As a result, there is high uncertainty the constant changes will have on the overall product. Furthermore, if the requirements are not monitored, which can be observed by a low score for the *Documentation* weight, then this can lead to tensions between different stakeholders' roles when the certain "agreed" requirements do not meet the initial set target goals.

Therefore, it can be concluded that there is a correlation between ABC and other organizations. Nevertheless, the framework cannot be generalized to a very large population, given the limitations discussed in this chapter. Nevertheless, the idea of framework transferability to other similar organizations will be further analyzed.

Figure 23: External Developers BWM weights

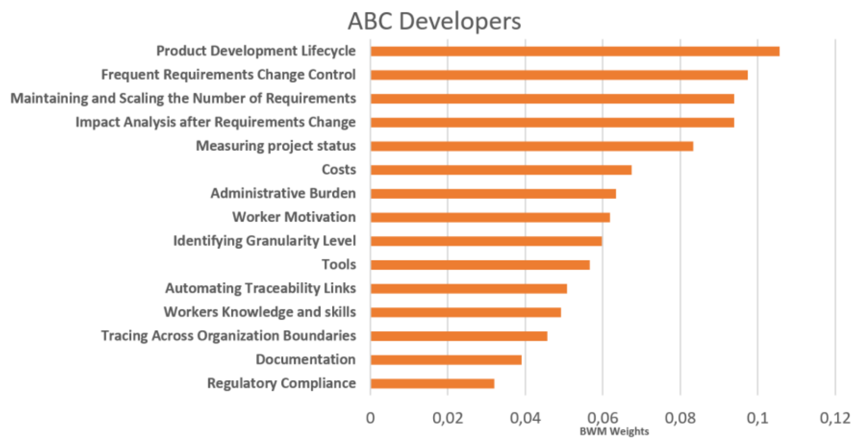


Figure 24: ABC Developers BWM weights

Transferability

The RTI framework provides clear steps for the management to strategize RT planning based on the organization's resources, and make the correct trade-offs given the organization's need to stay agile. Due to the similarities in the workflows and the close resemblance in the RT weights given by knowledge workers from different interviewed organizations, it can be concluded that the developed framework might not be generalizable for all organizations. However, the RTI framework steps are designed to guide the management team in the right direction, whilst at the same time, it gives full flexibility and responsibility to make the right tradeoffs based on the organization's needs. Because of such characteristics, it can be argued that the framework can be transferred and used by many organizations which face similar challenges as ABC (see *Chapter 5.1.6*)

Construct Validity

When identifying the RT and Agile way of working challenges from the literature, no concrete definitions were mentioned in the scholars' studies. Overall, there is a slight variation on how the researchers interpreted challenges such as *frequent requirements change control*, *impact analysis*, *measuring project status*, *worker motivation*, and others. As a result, the definition given in the current research for a particular challenge, does not necessarily reflect word-for-word interpretation and constructs described in the studies of other scholars. Therefore, future studies need to pay close attention to the common aspects of the definitions before analyzing and reaching a certain conclusion based on the data presented in this research and other studies.

6.2 Limitations - BPMN Roadmap

Collecting data for the BPMN Roadmap analysis is also associated with several threats that could have influenced the data interpretation and lead to wrong conclusions. These threats can be summarized as follows:

- As specified in data collection *Chapter 4.2.2*, due to time constraints, only three interviews were conducted to identify practitioners' opinions on the BPMN roadmap. Even though three distinctive roles were selected (manager, architect, developer), the limited number of interviews does not represent the entire population of practitioners at ABC.
- After designing BPMN Roadmap, the interviews also offered the possibility to present to the practitioners the RTI Framework. However, due to the limited availability of the employees and the desire to get feedback for the framework and roadmap, a big part of the interview process was rushed when it came to describing certain components of the BPMN Roadmap. As a result, there is a high risk that practitioners did not fully understand certain parts of the framework or roadmap, leading to more skewed feedback.
- Another limitation associated with the interview process is that none of the interviewees were familiar with the BPMN language. Thus, this could have influenced the practitioners' interpretation of the workflow, even if they were guided through each step of the roadmap.

Generalizability and Transferability

This research had no objective to generalize the results of the BPMN roadmap. Nevertheless, from the second round of interviews, performed for evaluation purposes, the feedback received from the practitioners was that the roadmap, in some instances, is too generic to be directly implemented at ABC. The practitioners were looking for a solution where the tradeoffs of RT were already made for them for specific projects in the organization. However, the roadmap was intended to serve as a flexible guideline that managers can adapt to their needs, rather than set-in-stone steps to follow to implement accurate and flexible RT in the Agile environment.

Transferability

From the transferability perspective, the BPMN roadmap could serve other similar organizations to ABC in terms of establishing better processes and strategic planning for RTI. *Figure 22,* which describes the BPMN roadmap, clearly shows the tradeoffs that a knowledge worker should make in order to: select the correct project methodology, how and when to choose the proper project tools, when should a manager start the negotiations with other stakeholders regarding RT, and how at a later stage, the Agile practices can be included. These steps are not unique to ABC, rather they should be viewed as an extension of the RTI framework. As a result, stakeholders outside ABC could use the exact roadmap for their goals. Nevertheless, each company will have to adapt the processes based on their resources.

7 Conclusion

The process of integrating RT in Agile environments can represent a long and complex journey where multiple tension points have to be considered in different moments in time. The existing literature studies usually focus on a group of RT challenges for which solutions are provided in the form of automated tools, algorithms, and trace models. However, rarely an organization seeks a traceability tool or model that can solve a particular traceability issue. Instead, the businesses look for systems that support a wider range of activities and are capable of integrating with the stakeholders' workflow and needs.

7.1 Answering MRQ

MRQ: ***How can BPMo enable requirements traceability in Agile environments of small and medium-sized enterprises?***

From the collected data during the literature and case study, it is possible to conclude that BPMo could improve RT in SMEs similar to ABC environment, with the condition that certain steps are followed. During the research, one of the first findings is that tackling only the identified RTI challenges within an organization is less likely to yield visible improvements. As argued in the theoretical framework, the RTI challenges are only a symptom of a series of interconnected problems such as: missing business and project traceability goals, neglection of stakeholders' interests in traceability, lack of processes for project planning, etc. The practitioners at ABC have ranked as a top priority the challenges such as: frequent requirement change control, documentation, impact analysis on requirement change, tracing across organizational boundaries, product development lifecycle, etc. However, the employees seemed to frequently refer to a certain cause that is creating these challenges. Some of the mentioned reasons were that the ABC is missing requirements capture processes; there is no responsibility assigned for prioritization of the requirements; more has to be done to coordinate between different departments/teams and other aspects. These findings strengthen the idea described in the theoretical framework that RTI challenges are interconnected with other concepts within the business. As a result, the interconnection of the challenges can create a chain reaction that makes the organizations' processes or product development more complex and less stable. Furthermore, the complexity rises with time. However, sometimes due to constraints such as money, time, human capital, and tensions created by other people, the practitioners cannot fully identify the root causes of the problem, or they deliberately neglect the interconnection of the problem with other people challenges.

Nevertheless, only modeling the business processes for RT integration might not lead towards significant results either. Scholars, and ABC practitioners, have argued that an upfront-defined traceability strategy is indeed required. The RTI Framework was created to help guide practitioners to develop more elaborate traceability strategies. During a case study interview, a practitioner identified that the steps shown in the RTI Framework closely resemble the steps followed by the ABC company when trying to integrate a new RT tool. However, the tool was never integrated due to a lack of support. The practitioner mentioned that RTI Framework process was not fully followed because no BPMo was developed, the RTI challenges were not fully identified, and the existing ABC processes were not entirely analyzed.

Nonetheless, the interviewees described the RTI framework as a source to provide a good perspective on project management and traceability implementation. The findings also strengthen the idea that different roles during the project development are interested in different traceability aspects. An example is the BWM results shown in the research. The managers graded *Documentation* as one of their top challenges, while developers identified documentation as a challenge that causes the least tension in their work. However, there are more similarities in the BWM results than the divergence between the perspective of the roles on the traceability problems.

In conclusion, it can be pointed that BPMo might not be enough in some instances to provide better traceability within SMEs with dynamic environments. However, translating the RTI Framework steps into specific processes that are individual for each organization can lead to better integration of RT in Agile workflow. The ABC practitioners have identified that the designed BPMN roadmap can improve ABC RTI challenges such as frequent requirements change control, requirements management, and others. However, the management team might not have the human capital or time to develop unique strategies per project. As a result, even though the roadmap facilitated the communication of the workflow strategy between different stakeholders and contained relevant solutions for certain RTI challenges, the practitioners were looking for processes templates that can be directly applied based on different tasks and phases defined in the BPMN roadmap. Furthermore, the practitioners identified it as useful to have a common language across the organization to describe business processes. Nevertheless, questions were raised on how to gain the critical mass for the workflow implementation? Who will be responsible for ensuring the quality of the process? How to convince other employees to follow the new roadmaps? Answering these questions was never the scope of the current research; however, it can be a good starting for future research. More specifically, it is interesting to define several BPMN templates per each phase of the RTI Framework and analyze which ones can be generalized for the SMEs.

7.2 Academic Reflection

Scientific Implications

The current study provides a new perspective on RT integration in Agile environments by adopting a managerial perspective rather than a technical perspective taken in the literature. The newly adopted perspective focuses on the business processes, stakeholders' needs, and tradeoffs made based on the company resources and business goals. Such an approach allows taking a broader view on the problem of RT, and design business processes that provide the possibility to support some of the technical solutions already described in the literature.

Furthermore, the RT and Agile problem have been acknowledged for a long time, but scholars were still missing essential data and direction on how to map and direct the research in a way that meaningful insights into stakeholders' needs and strategic planning are captured. The current study aims to serve as a point of reference on how the managerial perspective can be framed so that a clear objective is maintained and the relevant research gaps are filled. Furthermore, the thesis brings meaningful data in terms of stakeholders' interests/needs in requirements traceability by using the BWM method. The approach shall provide future researchers with a good foundation for investigating the processes that will facilitate the negotiation procedure and the tradeoffs to be taken within a dynamic organization that aims to adopt RT. Moreover, a clear visual representation of the framework and roadmaps, which are flexible, can be transferred by scholars to various businesses in order to investigate what practitioners value the most and how the interests of stakeholders vary based on industry, organization size, business structure, responsibilities, etc.

It cannot be claimed that this work provides a dichotomous answer to the problem of RTI in Agile. Nevertheless, it can be argued that the thesis can represent a good starting point for what can be the direction of future research when a managerial/business perspective is adopted to the problem of RT and the Agile way of working. The study provides a good foundation for topics such as: how business resources can influence RT, what is the level of stakeholders interests based on their role, how to make RT tradeoffs based on existing organization constraints, how business process modeling can help improve RT in Agile work environments.

Societal Relevance

The small and medium-sized technology companies that work in an Agile environment will be the main beneficiaries of the *RTI challenges analysis*, *RTI Framework*, and *BPMo roadmap*. All the three main groups of the involved stakeholders: managers (product managers, product owners, quality assurance managers), software developers, and customers, will benefit from the delivered Business Process Model, which was design to minimize the RT challenges which create the most tension in the organization. Furthermore, the feedback provided by practitioners could also serve as a contribution for the reader to know the main drawbacks, necessary processes improvements, and potential future research.

Moreover, the societal relevance of this research is underlined by the participation of four technological companies in the data collection, which further served as the main raw material for the results. During the interviews, a high interest in the topic and a curiosity for the study results were observed among the respondents. This represents one more argument in favor of the high societal relevance of this research.

7.3 Managerial Implications

This paper provides a design-based analysis considering that an RTI framework and a BPMo roadmap have been developed to serve as a guideline for organizations trying to implement the traditional RT techniques with the Agile methodologies into their work environment. Both complementary designs provide clear steps for RT strategizing and show the tradeoffs a practitioner should make in the process. Management requires to make trade-offs between what is desirable and feasible given limited resources and the need to stay agile (flexible). As a result, a framework for making these trade-offs is developed. In addition, a model capturing the main elements needed for traceability is designed, which can be modified/customized by other companies to realize RT given the limitations (or no limitations) brought by resources and business goals/strategy.

ABC Contribution

This study has significant importance, especially for ABC company, since the case study was conducted at the organization. First of all, a considerable contribution is the identification of the current challenges faced by the knowledge workers: managers, internal customers, and software developers, when it comes to RTI in an Agile environment.

As concluded during the second round of the interviews, the designed BPMo Roadmap has the potential of minimizing the main tensions identified per each group of employees, and namely:

- For managers: documentation, regulatory compliance, and measuring the project status;
- For software developers: product development lifecycle, synced tools, clearly defined responsibilities.

Lastly, this research can serve as a source of measuring reliability for future studies conducted at ABC on the topics of Agile and RT.

7.4 Future Research

This research has provided an Agile & RTI Framework, which implies the concept of defining weights for RTI challenges. The created Framework was further used to develop the BPMo of RTI in Agile methodologies. However, there is always scope for improvement and space for further research as suggested below:

- As observed during the interviews, there could be a situation when two stakeholders with different powers/interests should negotiate about the equally given weight to two different RT challenges. Therefore, we recommend future research to investigate what will be the appropriate decision-making process to negotiate between different RTI challenges.

- Taking into account the limited scope of the current research, another possibility of further investigation could be a deep analysis of the weights for the challenges in the context of other organizations activating in various industries and performing a comparison analysis between those industries.

- Another area of research to be explored is the BPMo language to use for creating process models to sustain RT in Agile environments.

- Furthermore, the practitioners classified the RTI Framework and BPMN roadmap as generic in some instances. Therefore, the knowledge workers suggested creating specific process templates to choose from in each step/phase of the framework and roadmap.

7.5 Reflection on Management of Technology

The process of researching and writing my master thesis provided me with a valuable opportunity to not just learn about how to navigate better the uncertainties when it comes to integrate RT practices with Agile methodologies but also to develop a unique contribution to this topic, and namely Agile & RTI Framework, the concept of defining weights for RTI challenges, as well as to develop the BPMo of RTI in Agile methodologies.

My research began with the broad desire to learn more about what methods and tools tech companies use for the adaptability of BPMa in an Agile environment. My interest in this topic arose during the course “Digital Business Process Management” [MOT1531], where many relevant subjects for my research were discussed: Business process improvement strategies and approaches, Agility and adaptability of BPMo systems, and the users, business process and agile enterprises. After this course, I was already determined with the direction of the topic I would like to further investigate in my master thesis.

There were many other courses during MOT program from which I gained valuable knowledge relevant to my research. These courses are the following:

From the “Complex System Engineering” discipline, a good understanding about the concept of design thinking in socio-technical systems helped me a lot in finding a suitable design approach for specific RT challenges in Agile Methodologies.

From the *research design* perspective, there were two courses that provided useful insights on how to design and execute research, as well as how to interpret and critically assess the outcomes of research: “Research Methods” [MOT2312] and “Preparation for Master Thesis” [MOT2004]. During these two courses, I gained sufficient knowledge for developing and executing all the research steps from the initial research problem until drawing a conclusion.

Considering all the above mentioned, this research is very relevant to the MOT curricula through the following subjects: Product Development Methodologies (Agile, Waterfall, V-Model), Business Process Management, Business Process Modeling, designing solutions for socio-technical systems, and defining process management strategies to tackle certain challenges.

Bibliography

Aagesen, G., & Krogstie, J. (2015). BPMN 2.0 for modeling business processes. In *Handbook on Business Process Management 1* (pp. 219–250). Springer.

Alaa, G., & Samir, Z. (2014). A multi-faceted roadmap of requirements traceability types adoption in Scrum: an empirical study. *2014 9th International Conference on Informatics and Systems*, SW-1.

Antonino, P. O., Keuler, T., Germann, N., & Cronauer, B. (2014). A non-invasive approach to trace architecture design, requirements specification and agile artifacts. *2014 23rd Australian Software Engineering Conference*, 220–229.

Antoniol, G, Berenbach, B., Eyged, A., Ferguson, S., Maletic, J., Zisman, A., Holbrook, E. A., Sundaram, S., Zou, C., & Gotel, O. (2006). Center of Excellence of Traceability Technical Report. *Center of Excellence for Traceability*.

Antoniol, Giuliano, Cleland-Huang, J., Hayes, J. H., & Vierhauser, M. (2017). Grand Challenges of Traceability: The Next Ten Years. *ArXiv Preprint ArXiv:1710.03129*.

Batool, A., Motla, Y., Hamid, B., Asghar, S., Mukhtar, M., & Ahmed, M. (2013). Comparative study of traditional requirement engineering and Agile requirement engineering. In *International Conference on Advanced Communication Technology, ICACT*.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., & Jeffries, R. (2001). Manifesto for agile software development. agilemanifesto. org. *Utah, Feb*.

Bjarnason, E., Wnuk, K., & Regnell, B. (2011). A case study on benefits and side-effects of agile practices in large-scale requirements engineering. *Proceedings of the 1st Workshop on Agile Requirements Engineering*, 1–5.

Borg, M., Alégroth, E., & Runeson, P. (2017). Software Engineers' Information Seeking Behavior in Change Impact Analysis-An Interview Study. *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*, 12–22.

Carniel, C. A., & Pegoraro, R. A. (2017). Metamodel for Requirements Traceability and Impact Analysis on Agile Methods. *Brazilian Workshop on Agile Methods*, 105–117.

Cleland-Huang, J. (2012). Traceability in agile projects. In *Software and Systems Traceability* (pp. 265–275). Springer.

Cleland-Huang, J., Gotel, O. C. Z., Huffman Hayes, J., Mäder, P., & Zisman, A. (2014). Software traceability: trends and future directions. In *Future of Software Engineering Proceedings* (pp. 55–69).

Cohen, D., Lindvall, M., & Costa, P. (2003). *DACS State-of-the-Art/Practice Report. Agile Software Development*. Washington DC: Fraunhofer Center Maryland. Disponible en http://citeseerx ….

Curcio, K., Navarro, T., Malucelli, A., & Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, *139*, 32–50.

Deehan, N. (2020). *How CMMN never lived up to its potential*. https://camunda.com/blog/2020/08/how-cmmn-never-lived-up-to-its-potential/

Doyle, J. K. (2005). Face-to-face surveys. *Encyclopedia of Statistics in Behavioral Science*.

Elamin, R., & Osman, R. (2017). Towards requirements reuse by implementing traceability in agile development. *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, *2*, 431–436.

Espinoza, A., & Garbajosa, J. (2011). A study to support agile methods more effectively through traceability. *Innovations in Systems and Software Engineering*, *7*(1), 53–69.

Furtado, F., & Zisman, A. (2016). Trace++: A Traceability Approach to Support Transitioning to Agile Software Engineering. *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 66–75. https://doi.org/10.1109/RE.2016.47

Gayer, S., Herrmann, A., Keuler, T., Riebisch, M., & Antonino, P. O. (2016). Lightweight traceability for the agile architect. *Computer*, *49*(5), 64–71.

Ghanam, Y., & Maurer, F. (2009). Extreme product line engineering: Managing variability and traceability via executable specifications. *2009 Agile Conference*, 41–48.

Ghazarian, A. (2008). Traceability patterns: an approach to requirement-component traceability in agile software development. *Proceedings of the 8th Conference on Applied Computer Scince*, 236–241.

Gilbreth, F. B., & Gilbreth, L. M. (1921). *Process charts*. author.

Gomes, A., & Pettersson, A. (2007). *Market-driven requirements engineering process model–MDREPM*.

Gotel, O. C. Z., & Finkelstein, C. W. (1994). An analysis of the requirements traceability problem. *Proceedings of IEEE International Conference on Requirements Engineering*, 94–101.

Gotel, O., Cleland-Huang, J., Hayes, J. H., Zisman, A., Egyed, A., Grünbacher, P., & Antoniol, G. (2012). The quest for ubiquity: A roadmap for software and systems traceability research. *2012 20th IEEE International Requirements Engineering Conference (RE)*, 71–80.

Griffiths, J., James, R., & Kempson, J. (2000). Focusing customer demand through manufacturing supply chains by the use of customer focused cells: An appraisal. *International Journal of Production Economics*, *65*(1), 111–120.

Hayes, J. H., Antoniol, G., Cui, L., & Yu, T. (2017). Too Little for Big Data? *Grand Challenges of Traceability: The Next Ten Years*, 25.

Hoang Duc, V. (2015). *Traceability in Agile software projects*.

Holder, K., Zech, A., Ramsaier, M., Stetter, R., Niedermeier, H.-P., Rudolph, S., & Till, M. (2017). Model-based requirements management in gear systems design based on graph-based design languages. *Applied Sciences*, *7*(11), 1112.

ISO. (2015). *Management system guidance 8.5 Product and service provision*. https://www.iso9001help.co.uk/8.5.2 Identification & Traceability.html

Jackson, J. (1991). A keyphrase based traceability scheme. *IEE Colloquium on Tools and Techniques for Maintaining Traceability During Design*, 1–2.

Jakobsson, M. (2009). *Implementing traceability in agile software development*. Department of Computer Science, Lund University.

Jander, K. (2016). *Agile Business Process Management: Concepts and Tools for Long-running Autonomous Business Processes*. Fachbereich Informatik der Universität Hamburg.

Jarzębowicz, A., & Weichbroth, P. (2021). A Systematic Literature Review on Implementing Non-functional Requirements in Agile Software Development: Issues and Facilitating Practices. *International Conference on Lean and Agile Software Development*, 91–110.

Jørgensen, M. (2018). Do agile methods work for large software projects? *International Conference on Agile Software Development*, 179–190.

Karagiannis, D. (2015). Agile modeling method engineering. *Proceedings of the 19th Panhellenic Conference on Informatics*, 5–10.

Kirova, V., Kirby, N., Kothari, D., & Childress, G. (2008). Effective requirements traceability: Models, tools, and practices. *Bell Labs Technical Journal*, *12*(4), 143–157.

Kroll, J. A. (2021). Outlining Traceability: A Principle for Operationalizing Accountability in Computing Systems. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 758–771.

Laux, I. (2019). *Coexisting Plan-driven and Agile Methods: How Tensions Emerge and Are Resolved*.

Lee, C., Guadagno, L., & Jia, X. (2003). An agile approach to capturing requirements and traceability. *Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE 2003)*, *20*.

Lee, J. C., & McCrickard, D. S. (2007). Towards extreme (ly) usable software: Exploring tensions between usability and agile software development. *Agile 2007 (AGILE 2007)*, 59–71. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4293576

Lefering, M. (1993). An incremental integration tool between requirements engineering and programming in the large. *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*, 82–89. https://doi.org/10.1109/ISRE.1993.324832

Leymann, F., & Roller, D. (1999). *Production workflow: concepts and techniques*. Prentice Hall PTR.

Liang, F., Brunelli, M., & Rezaei, J. (2020). Consistency issues in the best worst method: Measurements and thresholds. *Omega*, *96*, 102175.

Lin, F., & Chen, H. (2019). *Comparative Study of Requirements Traceability in Facing Requirements Change: Systematic Literature Study and Survey*.

Lormans, M. (2009). *Managing requirements evolution: Using reconstructed traceability and requirements views*.

Lu, R., & Sadiq, S. (2007). A survey of comparative business process modeling approaches. *International Conference on Business Information Systems*, 82–94.

Mader, P., Gotel, O., & Philippow, I. (2009). Motivation matters in the traceability trenches. *2009 17th IEEE International Requirements Engineering Conference*, 143–148.

Maletic, J. I., Munson, E. V, Marcus, A., & Nguyen, T. N. (2003). Using a hypertext model for traceability link conformance analysis. *Proc. of the Int. Workshop on Traceability in Emerging Forms of Software Engineering*, 47–54.

Malviya, S., Vierhauser, M., & Cleland-Huang, J. (2017). Traceability Queries and Strategies for the Requirements Engineering Domain. *Grand Challenges of Traceability: The Next Ten Years*, 16.

Marin, M. A. (2016). *Introduction to the Case Management Model and Notation (CMMN). 1–23*.

Martin, R. C., & Melnik, G. (2008). Tests and requirements, requirements and tests: A möbius strip. *IEEE Software*, *25*(1), 54–59.

Niu, N., Savolainen, J., Wang, W., & Alenazi, M. (2017). Best of Both Worlds: Synthesizing the Human and Method Sides of Requirements Tracing. *Grand Challenges of Traceability: The Next Ten Years*, 13.

Pereira, J. L., & Silva, D. (2016). Business process modeling languages: A comparative framework. In *New Advances in Information Systems and Technologies* (pp. 619–628). Springer.

Pitchford, M. (2019). *9 - Embedded Software Quality, Integration, and Testing Techniques* (R. Oshana & M. B. T.-S. E. for E. S. (Second E. Kraeling (eds.); pp. 269–338). Newnes. https://doi.org/https://doi.org/10.1016/B978-0-12-809448-8.00009-6

Qasaimeh, M., & Abran, A. (2013). An audit model for ISO 9001 traceability requirements in agile-XP environments. *Journal of Software*, *8*(7), 1556–1567.

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, *20*(5), 449–480.

Ratanotayanon, S., Sim, S. E., & Gallardo-Valencia, R. (2009). Supporting program comprehension in agile with links to user stories. *2009 Agile Conference*, 26–32.

Razavian, M., Turetken, O., & Vanderfeesten, I. (2016). When cognitive biases lead to business process management issues. *International Conference on Business Process Management*, 147–156.

Recker, J. (2010). Opportunities and constraints: the current struggle with BPMN. *Business Process Management Journal*.

Regan, G., Mc Caffery, F., Mc Daid, K., & Flood, D. (2013). Medical device standards' requirements for traceability during the software development lifecycle and implementation of a traceability assessment model. *Computer Standards & Interfaces*, *36*(1), 3–9.

Reis, E. (2011). The lean startup. *New York: Crown Business*, *27*.

Rempel, P., Mçder, P., & Kuschke, T. (2013). An empirical study on project-specific traceability strategies. *2013 21st IEEE International Requirements Engineering Conference (RE)*, 195–204.

Rezaei, J. (2015). Best-worst multi-criteria decision-making method. *Omega*, *53*, 49–57.

Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences*, *4*(2), 155–169.

Schwarz, H., Ebert, J., & Winter, A. (2010). Graph-based traceability: a comprehensive approach. *Software & Systems Modeling*, *9*(4), 473–492.

Sekaran, U., & Bougie, R. (2016). *Research methods for business: A skill building approach*. John Wiley & Sons.

Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040–1051.

Serrat, O. (2017). The five whys technique. In *Knowledge solutions* (pp. 307–310). Springer.

Sheppard, V. (2020). *Research Methods for the Social Sciences: An Introduction*.

Sincero, S. M. (2012). *Personal Interview Survey*.

Spanoudakis, G., & Zisman, A. (2005). Software Traceability: A Roadmap. *Handbook of Software Engineering and Knowledge Engineering*, *3*. https://doi.org/10.1142/9789812775245_0014

Stacey, R. D. (2007). *Strategic management and organisational dynamics: The challenge of complexity to ways of thinking about organisations*. Pearson education.

Taherdoost, H. (2016). Sampling methods in research methodology; how to choose a sampling technique for research. *How to Choose a Sampling Technique for Research (April 10, 2016)*.

Taromirad, M., & Paige, R. F. (2012). Agile requirements traceability using domain-specific modelling languages. *Proceedings of the 2012 Extreme Modeling Workshop*, 45–50.

Thelwall, M. (2017). Microsoft Academic: A multidisciplinary comparison of citation counts with Scopus and Mendeley for 29 journals. *Journal of Informetrics*, *11*(4), 1201–1212.

Uludag, Ö., Kleehaus, M., Caprano, C., & Matthes, F. (2018). Identifying and structuring challenges in large-scale agile development based on a structured literature review. *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, 191–197.

Urbieta, M., Antonelli, L., Rossi, G., & do Prado Leite, J. C. S. (2020). The impact of using a domain language for an agile requirements management. *Information and Software Technology*, *127*, 106375. https://doi.org/https://doi.org/10.1016/j.infsof.2020.106375

Weske, M. (2007). *Business process management architectures*. Springer.

Wiemuth, M., Junger, D., Leitritz, M. A., Neumann, J., Neumuth, T., & Burgert, O. (2017). Application fields for the new object management group (OMG) standards case management model and notation (CMMN) and decision management notation (DMN) in the perioperative field. *International Journal of Computer Assisted Radiology and Surgery*, *12*(8), 1439–1449.

Zhao, Y., Zaman, T. S., Yu, T., & Hayes, J. H. (2017). Using deep learning to improve the accuracy of requirements to code traceability. *Grand Challenges of Traceability: The Next Ten Years*, 22.

Appendix

Name	Description
Appendix A	**(Part I & II)** Selection criteria for literature background and theoretical framework
Appendix B	**(Part II)** the description of the language components; such as Flow Chart and CMMN
Appendix C	**(Part I & II)** Ethical guidelines on private data in the Humam Research
Appendix D	**(Part I)** Open-questions interview results used to measure the tension points such as “stakeholders needs”, “RT techniques”, “Business goals”, etc.
Appendix E	**(Part I)** BWM Weights for the 15 challenges identified from the literature study.
Appendix F	**(Part I)** The structure of the face-to-face survey
Appendix G	**(Part II)** Appendix E contains the semi-structured interview questions and transcripts related to BPMN roadmap design.

Table 17:Appendix - Table of contents

Appendix A - Selection Criteria

Almost two decades have passed since the introduction of Agile methodologies; companies are still struggling to fully manage and find the correct models to best adopt RE in Agile context. This observation led to my interest in further analyzing the problem and better understanding the complexity the organizations are facing. Moreover, the nature of the research, the large number of articles released on the topic from 1990 till 2020, and the impairment to provide a dichotomous response, can represent a symptom of a wicked problem *(Rittel & Webber, 1973)*. The nature of wicked problems is such that there can be room to find a research gap and narrow down the scope of the topic to a particular problem of interest.

Initial and refined keywords

The first step in my literature research was to define the keywords. The initial keywords used to perform the search were "Agile", "Requirements Tracing", Tension", "Management", "Traditional Methods" and “Business Process Modeling”. Any combination of the words mentioned above if introduced in the "Google Scholar" search engine, were resulting between 10 000 and 25 000 articles. Several articles (eight in total) from the initial results were used to better understand the different branches of research published on the topic of Agile, documentation, RT, problems in managing, and case studies. This also helped to gather more in-depth knowledge about the scientific terminology, which gravitates around the topic. After analyzing several papers on Agile RT and software development by Furtado & Zisman (2016), Lee & McCrickard (2007), and Curcio et al. (2018), a set of new refined keywords were found such as "Requirements Analysis", "Extreme Programming", "Requirements interaction management". A more detailed description of how the words were refined is further elaborated in *Table 18.*

Initial Keywords	Refined Keywords	Excluded (*based on scope*)
Agile	*Scrum, *Extreme Programming (XP), *Continuous software engineering, *Lean Software Development *Dynamic system development method *Feature-driven development (FDD), *Scaled Agile Framework (SAFe), *Rapid Application Development (RAD)	Even though the overall scope is more generic than a specific Agile methodology, all the refined keywords will be used to find the relevant articles for the scope.
Requirements Engineering	*Plan-Driven Approach, *Tayloristic methods *Process-oriented software *Traditional engineering methods *Requirements Management	*Traditional engineering methods *Tayloristic methods *Plan-Driven Approach *Process-oriented software
Tracking and Tracing	*Horizontal Traceability, *Lightweight Traceability *Documentation *Requirements interaction management	-
Business process Modeling	*Business process management *BPMN Approaches *Business process modeling languages *Agile business process modeling	
Tension	*Challenges/Problems/Issues	-

Table 18: Search words refinement process

After refining the keywords, the next step was to narrow down the scope of the research. Web search engines such as "*Google Scholar*", "*Microsoft Academic*", "*Scopus*", "*Wiley online library*," and "*TU Delft Research Repository*" were used in combination with the words defined in *Table 18*. The number of returned results from the search engines was used to narrow the scope, but it did not represent the selection criteria for scientific articles.

Source Name	Relevance
Google Scholar	The search engine was used to perform the initial research on the topic to get a better understanding of the generic concepts, it helped to refine the initial keywords and narrow down the scope.
Microsoft Academic	According to (Thelwall, 2017), Microsoft Academic can be considered a good source for research. For the year 2017, certain searches on the engine have provided ~5% more results than Scopus. The engine is also known to index spurious papers that come from arXiv and SSRN. Therefore, the search engine will be used with care, and papers will be studied on reliability (see *Table 20*).
Scopus / Wiley online library	The search engines were used to narrow down the scope with the use of refined keywords and find new sources.
TU Delft Research Repository	Used to narrow the scope and find new sources from the references.

Table 19: Search tools and their relevance

Research Strategy

Searching for papers using phrases such as "Continuous software engineering and Agile", "Scrum and Requirements Traceability challenges", "Lightweight Traceability in Agile context" led to many research articles and study cases that described certain models and techniques to achieve better traceability in Agile projects *(Ghazarian, 2008) (Qasaimeh & Abran, 2013)*. The words Agile/Scrum/XP and other methodologies were used interchangeably with various combinations from the list of defined keywords. This helped to find new articles, titles, and combinations that could be used to search further for new papers, for example, "Requirements interdependencies and Scrum". Because my search queries contained words such as "software", "requirements", "Agile", "tracing," many indexed articles were not related to the aspect of RT. To filter between the desired articles and papers which were outside the scope, a quick read through the title and abstract was done. Another indicator used for quick scanning through the articles was the author's keywords. From the initially selected articles, an analysis of references was performed to obtain further studies on the topic of traceability. A great help in this direction was a paper by *(Curcio et al., 2018)*, where research was performed on the overall RE concept to find research gaps and obstacles in implementing RE in Agile.

During my search for papers, I did not find any studies that tried to describe a generic process/model that has to be followed to implement RT in Agile. This led to new queries such as "BPMN and Requirements Tracing in SCRUM".

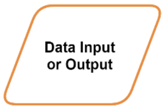
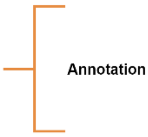
Selection Criteria

Criteria	Inclusion	Exclusion
References	Papers > 10 references	Papers < 10 references
Research Type	Books/Chapters, Journals, Scientific Journals, Grey Literature.	Personal blogs, News websites
Research Field / Topic	* Papers reflecting on Agile and Documentation. * Papers with the focus on Requirements Tracing (Requirements Engineering) and Agile. * Papers focusing on technology behind RT. * Papers describing the tension between RT and Agile methodologies * Study cases on RT and Agile. * Papers related to RT * Papers describing the tension to combine Plan Driven approaches with Agile methodologies.	* Papers related to advantages and disadvantages of Agile methodologies. * Papers related to differences between Agile and Plan Driven approaches. * Papers describing different Agile methodologies or Plan Driven methodologies.
Year	-	No restrictions
Credibility	The article had a clear objective, findings, and a justified conclusion. (no dichotomous answers)	The article does not provide an adequate description, findings, and conclusion.

Table 20: Research articles selection

Appendix B – BPMo Languages

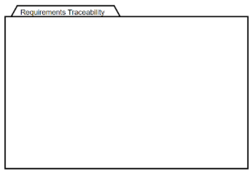
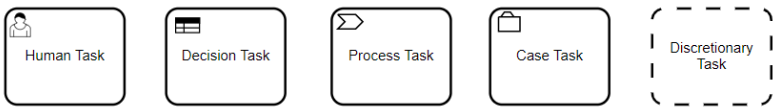
Flowchart Components

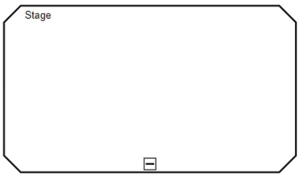
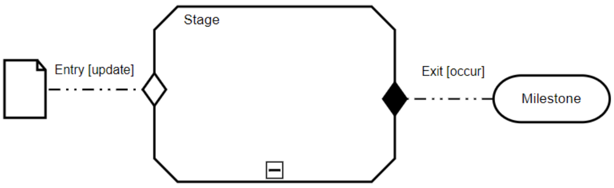
Component	Description
Flowline	The flowline shows the order of the workflow. It is used to connect the components of the diagram in a comprehensive way, so a clear order of the operation is shown. 
Terminals	The terminal is used to indicate the beginning or end of the process, task, or algorithm the designer tries to convey. The flowchart diagram should always start with a terminal. 
Process	The process component is where the user or the system/algorithm/program performs a set of operations that lead to a change of a specific value, state, location, or make a call to perform a particular action. 
Decision	The decision symbol illustrates a conditional operation that directs the process flow in one of two possible directions. The most common operation is usually dichotomous. 
Input/Output	The component indicates the action of data input or output in the process. The process of data input/output can be either manual or automated. 
Annotation	As in the case of many other modeling languages, annotations are used to present additional information about the process. The designer can view these as informative comments attached to the workflow. 
Predefined Process	A predefined process, also known as subroutines, is a component that allows the designer to use a predefined (existing) process and reference it anywhere in the workflow. 

Document	The component is used to denote that a particular process requires the use of a document that contains data from a database, from previous input, or hard drive. 

Table 21: Flowchart Components

CMMN Components

Component	Description
Case Plan & Roles	The case plan represents the component that provides a complete description of the case that is being handled. However, the CMMN modeling usually does not clearly describe how the data gets into the *case*. Furthermore, not all work tasks are clearly defined since management often relies on the knowledge workers' experience to handle a particular task. 
Tasks	The tasks are components that can only be part of a case plan. These represent the actual work that is accomplished during the case. CMMN distinguishes between several types of tasks: – **Human task:** the human task can be either *blocking* or *non-blocking*. The task refers to a specific role whose responsibility is to process and participate in solving the case. – **Decision task:** is used to call a decision requirements model from the case plan. The decision requirements table (model) helps to guide the user through the decision process. The task is considered to always be a blocking task. – **Process task:** A process task is executed using an imperative workflow similar to the *call activity* from the BPMN language. – **Case task:** A task which links to another CMMN diagram – **Discretionary task (CMMN 1.1):** The discretionary task keeps the case worker deciding whether to perform the particular task or skip it. 
Event Listeners	The event listeners in CMMN are similar in functionality to the events in BPMN language. However, the CMMN is mainly limited to three types of events: *generic event listener*, *timer event,* and *human event listener*. Such paucity of events is due to the declarative semantics of the language. 

<table>
<tr><td>Stages</td><td>A stage is used to divide the case into subdivisions. The stage allows the designer to group tasks, sequence flows, and sub-stages to form complex flows. The stages can also be discretionary. An example of the stages can be the division of customer complaints into two categories: product complaints and service complaints.
</td></tr>
<tr><td>Milestones</td><td>The milestones represent accomplishments (or sub-goals) within the process. The component is used to indicate that a particular stage within the case has been reached or completed. The process does not continue until the main task, events, or stages have not been completed. Therefore, the designer can use the milestones to convey the progress of a particular case better.
</td></tr>
<tr><td>Case File</td><td>The case file component is used to represent a document or data file that is relevant to a particular task, stage, or case. The case files are usually interlinked with other items via connectors. This implies that the other element for its execution uses the case file. For example, a case file can contain patient information, user account details, etc.
</td></tr>
<tr><td>Sentries (Criteria)</td><td>Sentries is a component used in CMMN to indicate entry and exit criteria to any task, milestone, stage, or case. Sentries have a diamond-shaped form and can also stand as stand-alone elements with no attachments to other tasks.
– Entry Criteria: The entry criterion set the condition for the process flow to complete all the entry components before continuing with the workflow, for example, completing a survey before moving to the next stage.
– Exit Criteria: The exit criterion informs the user when the workflow can continue from a particular task or stage. For example, a user can continue the workflow from a specific Stage only when a certain milestone was reached.
</td></tr>
</table>

Decorators	Decorators are components that provide more detail about a stage or task to be completed. Decorators are of four types: – **Auto Complete ■:** The decorator is used when the modeler wants the stage or case to complete as soon as all the items that composes stage are finished. – **Manual Activation ▷:** The manual decorator is used when the task or stage should be initialized manually by the user. If the decorator is not placed in the item, then the task will start automatically as the entry criteria is satisfied. – **Repetition #:** A repetition decorator is used when a milestone, stage, task, or case can be repeated multiple times. However, only the items that have at least one entry criteria can be assigned the repetition decorator. – **Required !:** The decorator is used when the process designer would like a stage, task, or milestone to be executed before the scope: case or stage, is allowed to be finalized.

Table 22: CMMN Main Components

TUDelft

Appendix C – Ethics Review for Human Research

Delft University of Technology
ETHICS REVIEW CHECKLIST FOR HUMAN RESEARCH
(Version 12.03.2021)

Table 2: Risk Assessment Checklist

Note: *if you answer "yes" to any of the questions in this checklist, please ensure that you summarise and confirm how these will be dealt with in Section IV (Risk Management and Informed Consent) below. Where appropriate please include the relevant advice/approval (eg: from the Privacy Team, Data Steward or HSE representative) as an additional attachment to this application.*

Potential Risk	**Yes**	**No**
1. Does the study involve participants who are particularly vulnerable or unable to give informed consent? (e.g., children, people with learning difficulties, patients, people receiving counseling, people living in care or nursing homes, people recruited through self-help groups).		X
2. Are the participants, outside the context of the research, in a dependent or subordinate position to the investigator (such as own children or own students)?[1]		X
3. Will it be necessary for participants to take part in the study without their knowledge and consent at the time? (e.g., covert observation of people in non-public places).		X
4. Will the study involve actively deceiving the participants? (For example, will participants be deliberately falsely informed, will information be withheld from them or will they be misled in such a way that they are likely to object or show unease when debriefed about the study).		X
5. Will the study involve discussion or collection of personal sensitive data (e.g., financial data, location data, data relating to children or other vulnerable groups)? Definitions of sensitive personal data, and special cases are provided on the TUD Privacy Team website.		X
6. Will drugs, placebos, or other substances (e.g., drinks, foods, food or drink constituents, dietary supplements) be administered to the study participants? *If yes see here to determine whether medical ethical approval is required*		X
7. Will blood or tissue samples be obtained from participants? *If yes see here to determine whether medical ethical approval is required*		X
8. Is pain or more than mild discomfort likely to result from the study?		X
9. Does the study risk causing psychological stress or anxiety or other harm or negative consequences beyond that normally encountered by the participants in their life outside research?		X
10. Will you be offering any financial, or other, inducement (such as reasonable expenses and compensation for time) to participants?		X

[1] **Important note concerning questions 1 and 2.** Some intended studies involve research subjects who are particularly vulnerable or unable to give informed consent. This includes research involving participants who are in a dependent or unequal relationship with the researcher or research supervisor (e.g., the researcher's or research supervisor's students or staff). If your study involves such participants, it is essential that you safeguard against possible adverse consequences of this situation (e.g., allowing a student's failure to complete their participation to your satisfaction to affect your evaluation of their coursework). This can be achieved by ensuring that participants remain anonymous to the individuals concerned (e.g., you do not seek names of students taking part in your study). Please ensure that you include such risks – and how you will mitigate against them in your risk section.

TU Delft

Potential Risk	Yes	No
Important: if you answered 'yes' to any of the questions mentioned above, you **MAY** be asked to submit a full Research Ethics Application.		
11. Will the experiment collect and store any personally identifiable information (PII), including name, email address, videos, pictures, or other identifiable data of human subjects?[2]		X
12. Will the experiment involve the use of devices that are not 'CE' certified? *Only, if 'yes': continue with the following questions:*		X
➢ Was the device built in-house?		
➢ Was it inspected by a safety expert at TU Delft? *(Please provide a signed device report)*		
➢ If it was not built in house and not CE-certified, was it inspected by some other, qualified authority in safety and approved? *(Please provide records of the inspection).*		
13. Has this research been approved by a research ethics committee other than this one? *If yes, please provide a copy of the approval and summarise any key points in your Risk Management section below.*		X
14. Is this research dependent on a Data Transfer Agreement with a collaborating partner or third party supplier? *If yes please provide as a copy of the signed DTA and summarise any key points in your Risk Management section below.*		X

[2] Note: You have to ensure that collected data is safeguarded physically and will not be accessible to anyone outside the study. Furthermore, the data has to be de-identified if possible and has to be destroyed after a scientifically appropriate period of time. Also ask explicitly for consent if anonymised data will be published as open data.

Appendix D – Interview and Survey Results (Part I)

Interview results from the face-to-face survey, including the open-question to identify tensions points at ABC

Management Perspectives

Description	Methods used for Traceability	Existing Challenges
Company: ABC Role: Project Manager, Product Owner Department: Test and Measurement / R&D Experience: 5-7 years Most Tension: Documentation Least Tension: Regulatory Compliance	Documentation (Products datasheet -> Module Specs -> SmartSheets) Each requirement is discussed with an architect and customer to assess feasibility. The process followed is a more top-down approach and is similar to V-Model. For smaller projects is used Agile methods with tools such as (Scrum Boards, Asana, Confluence)	Rapidly Growing company *The company is in the phase of R&D for the product to be released for the market* Backlog increases rapidly Need for ISO compliance
Company: ABC Role: Team Lead, Chief of staff Department: Production, IT, Manufacturing Experience: 10-15 Most Tension: Frequent Requirement Change Control Least Tension: Worker Motivation	Asana (tool) *PDM (tool)* Arena (tool)	There is no product owner to keep track of requirements There is no person to translate the customer requirements *One person having multiple roles within the project (developer & product owner)* *The requirements are not clear, and change quickly.* There is no process and responsibility assigned to do prioritization of the requirements (which can be done together, which ones can be combined) *Approvals for certain requirements change take time and is not efficient*

Company: ABC Role: Quality Assurance Department: Production, R&D Manufacturing, Purchasing, Sales Experience: 20 years Most Tension: Impact Analysis after the requirement change Least Tension: Costs	Asana, Gitlab (for process tracking), Jira, Confluence. Arena (Document business processes, QMS process, quality manual, describes the organization, how the processes are controlled, placed to drive the direction, objectives of the organizations) *PDM is not used anymore.* Arena TEC Design, electronics, PIC documentation, .gds, - Altium files linkage, all the documentation. A product requirement document. SFP is stored in the team project/design project, the stage where the document is moved to the arena and controlled, parametric functionality Actual testing is outside of the Arena software.	Deployment for the software, affects other parts (hardware, manufacturing). The technical approval and the deployment is not covered. Where we make of change, the change should take into consideration different types of aspects. *For example, if we change the documentation (requirement change), we need the customer to approve, then we have to change the software—no visibility on this process.*
Company: ABC (customer side) Role: Project Manager Department: Production, R&D, Test and Measurement, IT, *Manufacturing*, Purchasing, Sales Experience: 20 years Most Tension: Tools Least Tension: Measuring Project Status	Documents such word, excel, used to store requirements and processes design. Overall, there is no tool to capture the requirements and processes that are being developed properly.	Challenge is a consistently deployed /used process that should be followed and implemented within the organization. There is none. *Hard to define a traceability process because the product is still in the R&D phase.* *Also, there can be a communication barrier because of cultural differences.* *I do not believe we have a requirements capture process at all!*

Table 23: Management Interview Questions Results Summary

Developers Perspectives

Description	Methods used for Traceability	Existing Challenges
Company: ABC **Role**: Software Engineer, Project Manager **Department**: Production; Test and Measurement **Experience**: 5-7 years **Most Tension**: Measuring Project Status **Least Tension**: Regulatory Compliance	Asana (tool) Arena (tool) We don't have a specific process to keep traceability links between different components of a system.	Scenario: *A feature request comes in. Then, the tasks get to a specific developer that manages the requirement and develops it).* *The problem with this is that there is no time for documentation. Then more people join the project, and the customer changes their mind about a specific requirement. Then the burden of the requirement goes to the developer. But actually, It's about the decision made in the beginning.* Other: Lack of a full system that integrates everything. The non-functional requirements are not monitored tracked
Company: ABC **Role**: Architect, Data Analysist, Embedded Software Engineer. **Department**: Research and Development **Experience**: 7-10 years **Most Tension**: Costs **Least Tension**: Measuring Project Status	Issue lifecycle traceability in Asana, Spec doc -> Eng doc -> Qualification and Test documents on Dropbox/office/PDM (V-Model). Specification / Test traceability in RedMine (deprecated). BOM (build of material) in Arena, Mechanical design traceability to assembly and instructions in Solidworks PDM (part number linked to Arena).	*Constant feedback increases the requirement change rate, which bloats the requirement traceability log and makes it unusable. This decreases the traceability appeal to the team members, and the process ends up being avoided by the engineering staff.*

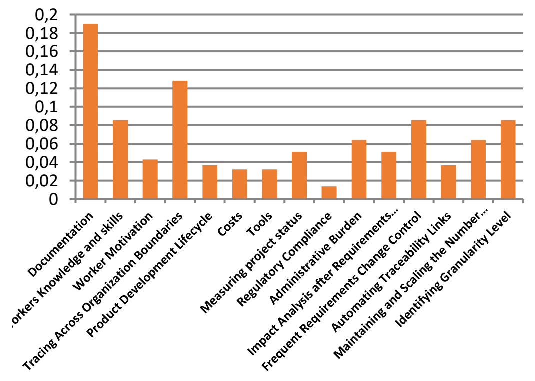
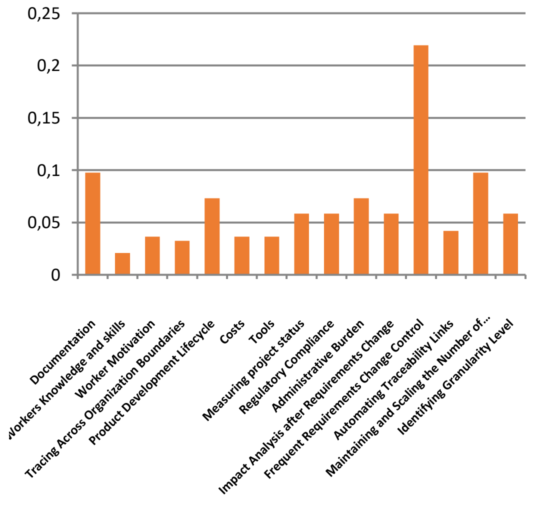
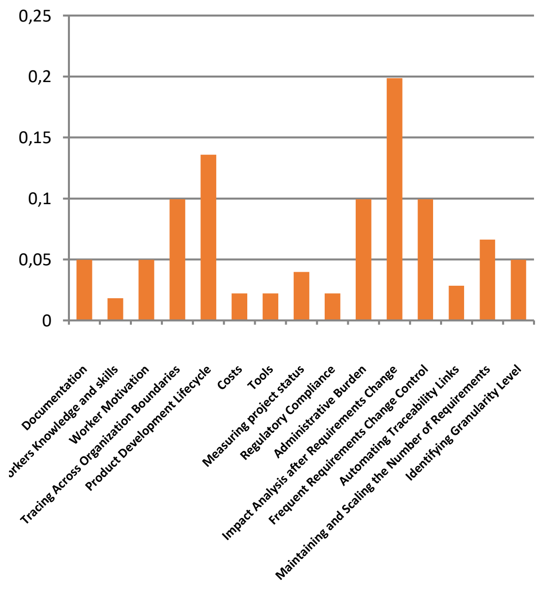
Company: ABC **Role**: Software Engineer, Hardware Engineer **Department**: R&D **Experience**: 20+ years **Most Tension**: Frequent Requirements Change Control **Least Tension**: Costs	Asana for tasks tracking Qualification, Test, and Requirements documents (e.g., word, excel) on Dropbox. Some projects use Confluence	For small projects (2-4 people) it is easier to maintain, update and keep the quality of requirements ourselves (as developers). This increases communication and understanding of the architecture and overall requirements of the system. However, for larger projects, this becomes burdensome, and now it becomes more difficult to maintain. At the moment, tighter control on requirements at the start of the project will be beneficial. However, few processes have been implemented to support such a change. Therefore, there is a challenge to develop a product of high quality in a flexible environment where requirements constantly change. *More flexibility results in lower quality control*
Company: External **Role**: Software Engineer **Department**: R&D **Experience**: 0-5 years **Most Tension**: Tracing Across Organization Boundaries **Least Tension**: Impact Analysis after the requirements change	Confluence for the documentation, at to define the design and architecture Confluence - User stories and Customer requirements Jira for code development Kanban board for software developer. Tracker - (Atlassian) Requirements	*Even if all the people followed the same course for Agile (green belt), every person insists on its own way to implement the Agile process and use the Jira and Confluence tools.* Difficulty to apply Agile in early research projects, Requirements are not necessarily come as a necessity.

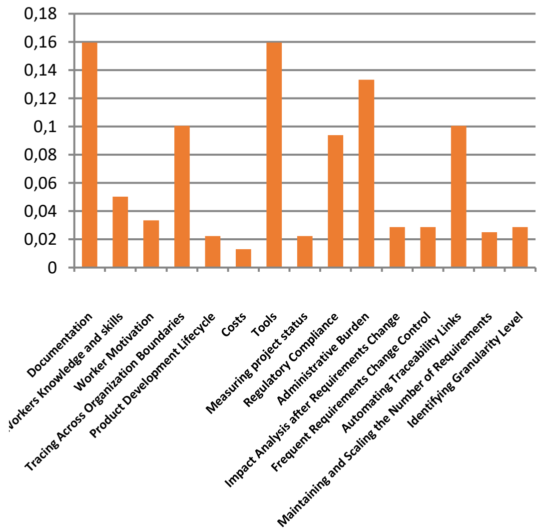
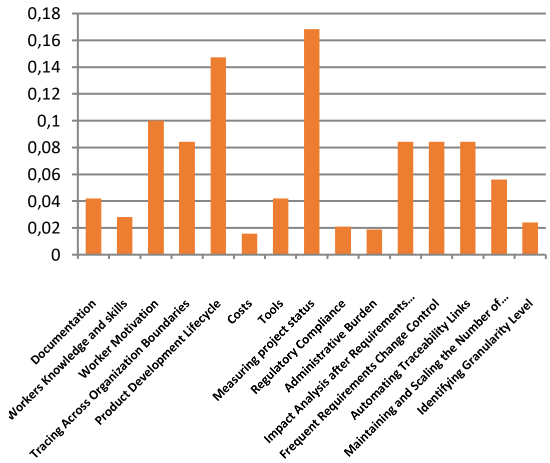
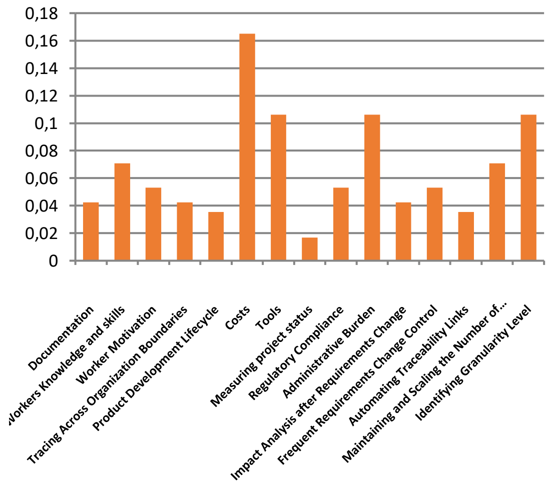
Company: External **Role**: Architect, Software Engineer, IoT Consultant **Department**: R&D, Production, Test and Measurement **Experience**: 0-5 years **Most Tension**: Frequent Requirements Change Control **Least Tension**: Tools	Jira, TTD, bi-weekly review with stakeholders Confluence for documentation. Good naming schemes + Conventions for naming - feedback on the naming conventions 100% test coverage. 10% of project time dedicated to QA	*Apprehension of overhead* *<u>Example:</u>* *Everything should go between 4 eye principle. Is it valuable enough to go through this traceability process? Something that the developers face when doing the traceability at this level?* Operational cost *A rapid succession of scope changes can lead to muddied traceability*
Company: External **Role**: Software Engineer **Department**: R&D **Experience**: 0-5 years **Most Tension**: Frequent Requirements Change Control **Least Tension**: Tools	JIRA (tickets management, requirements), Confluence (Design, best practices, architecture), GIT (Implementation and code), CI Jenkins (Verification, deployment, test, validation, really depends on the project), S&PC (Security and performance center responsible for maintenance, security updates and incidents)	*No standard chapters for documentation; any responsible architect can document the project in its own way.* No agreement on the official language to be used in the documentation. (Many documents in both English and Dutch) *The desires of the client might conflict with the Agile processes established within the company*

Table 24: Developers Interview Questions Results Summary

Appendix E – BWM (ξ (KSI) values and weights)

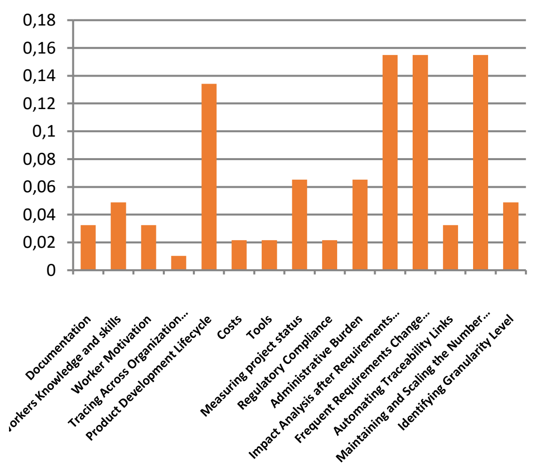
The interviews in table below are organized in the same order as in the Appendix E. Therefore, the KSI values are fist presented for the managers, developers, and external developers

Interview Details	Data	Value		Weights
Company: ABC Role: Project Manager, Product Owner Department: Test and Measurement / R&D	**KSI**	**0,066740**		
	Most Tension	Documents	0,189952 329	
	Least Tension	Regulator Compliance	0,013690 258	
Company: ABC Role: Team Lead, Chief of staff Department: Production, IT, Manufacture. Experience: 10-15	**KSI**	**0,073162234**		
	Most Tension	Frequent Requirement Change Control	0,219486 703	
	Least Tension	Worker Motivation	0,036581 117	
Company: ABC Role: Quality Assurance Department: Production, R&D Manufactur. Experience: 20 years	**KSI**	**0,062825103**		
	Most Tension	Impact Analysis	0,198608 39	
	Least Tension	Costs	0,022067 599	

Company: ABC (customer side) Role: Project Manager Department: Production, R&D, Test and Measurement, Manufactur., Experience: 20 years	**KSI**	**0,041511332**		
	Most Tension	Tools	0,159490905	
	Least Tension	Measuring Project Status	0,022333582	
Company: ABC Role: Software Engineer, Project Manager Department: Production; Test and Measurement Experience: 5-7 years	**KSI**	**0,068375355**		
	Most Tension	Measuring project status	0,168308566	
	Least Tension	Regulatory Compliance	0,0210385707129738	
Company: ABC Role: Architect, Data Analyst, Embedded Software Engineer. Department: Research and Development Experience: 7-10 years	**KSI**	**0,04753915**		
	Most Tension	Costs	0,164988814	
	Least Tension	Measuring Project Status	0,016778523	
Company: ABC Role: Software	**KSI**	**0,061197917**		
	Most Tension	Frequent Requirements Change Control	0,154947917	

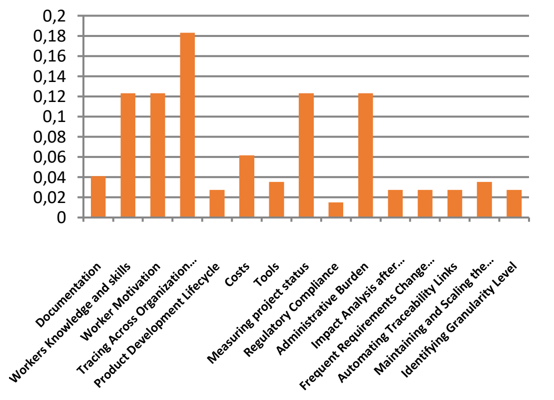
<table>
<tr><td rowspan="2">Engineer, Hardware Engineer

Department: R&D

Experience: 20+ years</td><td></td><td></td><td></td><td rowspan="2">
</td></tr>
<tr><td>Least Tension</td><td>Costs</td><td>0,021701389</td></tr>
<tr><td rowspan="3">Company: External

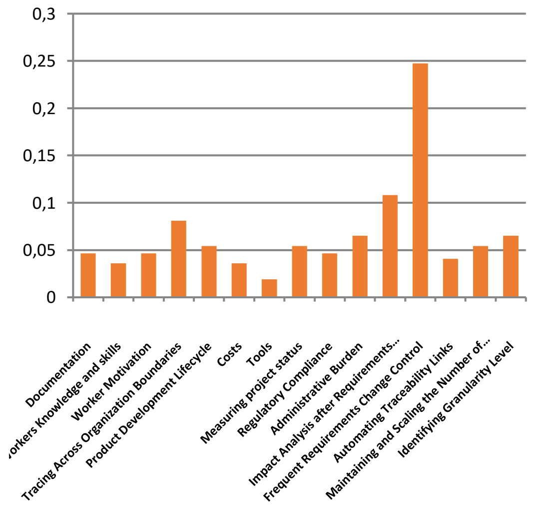
Role: Software Engineer

Department: Research and Development

Experience: 0-5 years</td><td>KSI</td><td colspan="2">0,062890625</td><td rowspan="3">
</td></tr>
<tr><td>Most Tension</td><td>Tracing Across Organization Boundaries</td><td>0,168308566</td></tr>
<tr><td>Least Tension</td><td>Regulatory Compliance</td><td>0,0210385707129738</td></tr>
<tr><td rowspan="3">Company: External

Role: Architect, Software Engineer, IoT Consultant

Department: Production, Test, and Measurement

Experience: 0-5 years</td><td>KSI</td><td colspan="2">0,077322909</td><td rowspan="3">
</td></tr>
<tr><td>Most Tension</td><td>Frequent Requirements Change Control</td><td>0,247433309</td></tr>
<tr><td>Least Tension</td><td>Tools</td><td>0,018901156</td></tr>
<tr><td rowspan="3">Company: External
Role: Software Engineer

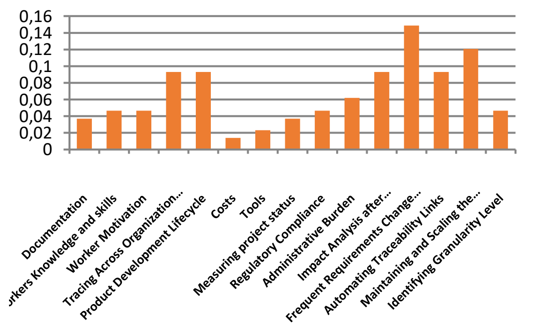
Department: Research and Development
Experience: 0-5 years</td><td>KSI</td><td colspan="2">0,037151703</td><td rowspan="3">
</td></tr>
<tr><td>Most Tension</td><td>Frequent Requirements Change Control</td><td>0,148606811</td></tr>
<tr><td>Least Tension</td><td>Tools</td><td>0,023219814</td></tr>
</table>

Table 25: BWM KSI Values and Weights

Appendix F – Interview and Survey (Part I)

Included the face-to-face survey used to collect the relevant information for the data analysis chapter. Which, in the end, helped to define the RTI Framework.

Requirements Tracing in Agile Environments

You are invited to participate in an survey for a master thesis research entitled "Requirements Tracing in Agile Environments" conducted at the faculty of Management of Technology at the TU Delft .

The purpose of this survey is to get a better understanding about the tensions faced by organizations when combining requirements traceability practices with Agile way of working. The results of this survey will be used to highlight which factors in practice create the most tension within the companies when integrating the two methodologies. The data from your participation will be used to identify and compare the literature findings with the existing practical challenges.

The final goal of the thesis is to develop a roadmap using business process modeling notation (BPMN) which will help practitioners to identify and tackle the most critical factors when combining the requirements traceability and the Agile methodologies. The roadmap should also help organization to better plan and allocate available resources so that tensions points are reduced during the process of product development.

Completing this survey takes about 30 to 45 minutes. Your participation in this research is entirely voluntary and you can withdraw at any time. I believe that there are no known privacy risks associated with this study. Your answers in this study will remain confidential to my best ability.

Thank you for your participation and support.

Mihai
M.Iliuhin@student.tudelft.nl

* Required

In what industry the company you are currently working is operating? (Please choose all that apply) *

- [ ] Electronics
- [ ] Software
- [ ] Telecommunication
- [ ] Energy
- [ ] Transport
- [ ] Construction
- [ ] Manufacturing
- [ ] Aerospace
- [ ] Pharmaceutical
- [ ] Health Care
- [ ] Hospitality
- [ ] Entertainment
- [ ] Other:

What is the size of the company?

Choose

Which of the following roles best describes your current position? (Please choose all that apply) *

- [ ] President/VP/CEO/COO/CIO/CTO
- [ ] Business Development Manager
- [ ] Project Manager
- [ ] Product Manager
- [ ] Product Line Manager
- [ ] Process Manager
- [ ] Operations Manager
- [ ] Product Owner
- [ ] Architect
- [ ] IT Staff
- [ ] Data Analyst
- [ ] Software Engineer
- [ ] Photonics Engineer
- [ ] Hardware Engineer
- [ ] Test Engineer
- [ ] Quality Assurance
- [ ] Other: ________

In what department(s) within the organization most of your work is taking place? (Please choose all that apply) *

- [ ] Production
- [ ] Research and Development (R&D)
- [ ] Test and Measurement
- [ ] Information Technology (IT)
- [ ] Manufacturing
- [ ] Purchasing
- [ ] Marketing
- [ ] Sales
- [ ] Finance
- [ ] Human Resources (HR)
- [ ] Other: ________

Years of relevant experience *

Choose ▾

To what extent do you agree/disagree with the following statements (1 refers to "strongly disagree" and 9 refers to "strongly agree"): *

	1	2	3	4	5	6	7	8	9
I have good knowledge about Agile methodologies (e.g. Scrum)	○	○	○	○	○	○	○	○	○
I have good knowledge about Requirements Engineering (Elicitation, Specification, Validation, Management, etc.)	○	○	○	○	○	○	○	○	○
I have good knowledge about the Requirements Traceability methodologies and processes	○	○	○	○	○	○	○	○	○

To what extent do you agree/disagree with the following statements (1 refers to "strongly disagree" and 9 refers to "strongly agree"): *

	1	2	3	4	5	6	7	8	9
I have good knowledge about Agile methodologies (e.g. Scrum)	○	○	○	○	○	○	○	○	○
I have good knowledge about Requirements Engineering (Elicitation, Specification, Validation, Management, etc.)	○	○	○	○	○	○	○	○	○
I have good knowledge about the Requirements Traceability methodologies and processes	○	○	○	○	○	○	○	○	○

What are the current tools and methodologies used by the organization for requirements traceability?

Your answer

Who in your opinion should be responsible to maintain, update, and keep the quality of the requirements and traceability links within a project? *

Your answer

Are there any challenges/problems the organization faces when it comes to integrate or maintain the Requirements Traceability processes alongside Agile way of working? *

Your answer

What solutions have been implemented within the organization to balance between Requirements Traceability and Agile way of working? *

Your answer

Are there any active projects currently within the company for improving the Requirements Traceability techniques alongside Agile way of working? *

Your answer

TU Delft

Which of the following factors is creating the "least tension" within your organization when it comes to integrate requirements traceability processes alongside Agile methodologies? *

Choose

How much more tension **the other factors** are creating when **compared to the "least tension factor"** selected above on the scale of 1-9. (1 refers to equal amount of tension and 9 refers to absolutely more tension) *

Please Note: You will find the selected "least tension factor" in the below list. For that factor a value of 1 should be set, because they are of equal tension.

	1	2	3	4	5	6	7	8	9
Documentation									
Workers Knowledge and skills									
Workers Motivation									
Tracing Across Organization Boundaries									
Product Development Lifecycle									
Costs									
Tools									
Measuring project status									
Regulatory Compliance									
Administrative Burden									
Impact Analysis after Requirement Change									
Frequent Requirement Change Control									
Automating the traceability links									
Maintaining and Scaling the Number of Requirements									
Identifying the Granularity Level									

Which of the following factors is creating the "most tension" within your organization when it comes to integrating requirements traceability processes alongside Agile methodologies? *

Choose

How much more tension **is** creating **the "most tension factor" selected above** when compared to other factors below on the scale of 1-9. (1 refers same level of tension and 9 refers to absolutely more tension.) *

Please Note: You will find the selected "most tension factor" in the below list. For that factor a value of 1 should be set, because they are of equal tension.

	1	2	3	4	5	6	7	8	9
Documentation									
Workers Knowledge and skills									
Workers Motivation									
Tracing Across Organization Boundaries									
Product Development Lifecycle									
Costs									
Tools									
Measuring project status									
Regulatory Compliance									
Administrative Burden									
Impact Analysis after Requirement Change									
Frequent Requirement Change Control									
Automating the traceability links									
Maintaining and Scaling the Number of Requirements									
Identifying the Granularity Level									

Appendix G – Interview and Results (Part II)

Interview Question	Intent
Q1: Will, in your opinion, the RTI Framework provide any assistance to the practitioners to develop better traceability strategies for their organization or specific projects?	RTI Framework Practicality
Q2: How, in your opinion, the BPMN Roadmap (if implemented), will affect the Requirements Traceability Challenges weighted by the developers and managers?	BPMN Diagram Efficiency and Accuracy
Q3: What are possible improvements that can be made to the BPMN Roadmap?	BPMN Diagram Improvements
Q4: What, in your opinion, will be the main implementation challenges when it comes to BPMN Roadmap diagram?	BPMN Implementation Challenges
Q5: Taking into consideration the Requirements Traceability Tensions at ABC, what, in your opinion, the company had to do differently to overcome the current challenges?	ABC Processes Retrospective

Table 26: BPMN and RTI Framework Interview Questions

Interview Answers

Interviewee Role ABC	Answers
Role: Software Engineer, Project Manager Department: Production; Test and Measurement Experience: 5-7 years Most Tension: Measuring Project Status Least Tension: Regulatory Compliance	**Q1: RTI Framework** *I definitely see value in having such a roadmap for project development and RT. However, this is a perfect picture of how things should be.* *In reality, there is a lot more to this process. For example, there should be a person (or team) responsible for the quality assessment of the processes, somebody to enforce such processes. Oftentimes, because of lack of budget, time, and practicality, certain steps (from the diagram) might be offseted.* *Furthermore, the diagram does not show how to handle the implementation for different departments.* ***Q2: BPMN Diagram*** *Very often at ABC to force developers to write documentation is a big challenge. However, if we have the process to write the documentation whilst the product is developed weekly, this will definitely improve the product quality and help measure the project status better.* *However, challenges such as "Frequent requirement change control" are not really addressed for the ABC case. The process shown in the roadmap will work great when an external customer is providing the requirements. In this case, you lock the requirements, and every change made is at the customer's expense.*

	At ABC, the customer I work with is usually internal, which has a lot of freedom to change the requirements at any moment in time. Moreover, the customer is treated too friendly and is welcomed at any moment to make a change. Therefore, some challenges are not really addressed by the BPMN. *On the other note, I think challenges such as Administrative Burden, Tools, Product development lifecycle will be improved and will be less of a burden.* ***Q3: BPMN Diagram Improvements*** *As briefly mentioned in the previous question, the roadmap assumes that the requirements are fixed! But in the ABC case, the requirements often times change during the "Sprint".* *Also, the negotiation process with the client is missing in the roadmap. When a new requirement comes, the customer's first question is: "How long will this take?". Oftentimes, this is a difficult question because if the requirement requires re-engineering of a big system component, we don't want to delete what we already have. This usually requires a repurposing of what is already in the system, and the amount of work is very hard to predict. Nevertheless, such a process is missing in the roadmap.* ***Q4: BPMN Implementation Challenges*** *The first thing that comes to mind is that people will start asking questions like: "Is to complex?", "Will this work in practice?". People are used to working in a certain way. So if you introduce something new, they will always say that the new methodology will not work. Also, other employees will say that their current workflow worked for 15+ years; why should they change?* *Another aspect is that no matter what process/workflow you will introduce, there will be cases when people will get creative and will bypass the workflow. As a result, this will make the process you try to implement look less valuable in the long term.* ***Q5: ABC Processes Retrospective*** *I do believe that some designing (processes) of the workflow I've seen the BPMN roadmap had to be implemented in the past.* *Furthermore, the company grows with no base processes to help with growing teams. As a result, the new employees who join the company bring with them their own methodologies and workflows and start to implement them because they see a lot of missing parts. As a result, we now end up with a conglomerate of processes and workflows per project and team. The employees were not properly introduced to the dynamics (workflows) of the*

	company, but deliverables were expected from them. Therefore the new colleagues had to improvise. *Therefore, having an initial strategy or teams to develop certain standard processes had solved many issues that we currently face.*
Role: Project Manager, Product Owner Department: Test and Measurement / R&D Experience: 5-7 years Most Tension: Documentation Least Tension: Regulatory Compliance	**Q1: RTI Framework** *I think structuring the projects is important, since this will generate some checklists. Also, following the process will generate some documentation by default.* *I do think the roadmap is useful. But it has to be templated, so it does not cause overhead. So, for example, if you have limited time, you cannot constantly monitor the progress or consider all the RT challenges.* *Therefore, even though the roadmap provides some perspective to RT and product development, in practice, it might be more useful to have limited options to choose from in each step of the RTI Framework. This will speed up the process a lot.* ***Q2: BPMN Diagram*** *I think for managers, such workflow should help generate traceability and documentation. The roadmap balances between development and traceability; this gives good flexibility and a project approach. Also, following such a process provides fewer moving targets which are good for developers and can reduce the "Frequent requirement change control" tension.* *Also, such processes provide a good foundation to have regulatory compliance (e.g., ISO 9001). Furthermore, the process described in the Roadmap is suitable to support the business goals.* *In theory, the roadmap should improve the currently faced RT challenges; however, there are a lot more unknowns in practice.* ***Q3: BPMN Diagram Improvements*** *I believe the roadmap is too generic to be implemented in practice directly. Also, some tasks in the process require a lot of work to be accomplished. Therefore, having some templating (a limited number of options to choose from) will provide a better roadmap/process.* *Furthermore, the roadmap/processes are developed with an external customer in mind; however, we also deal with internal customers at ABC. There is more flexibility in the workflow with internal customers, and all the procedures are usually not followed. You often do not define concrete traceability steps and requirements with the internal customers at ABC. Therefore, the*

	roadmap map could provide some administrative burden for certain scenarios. ***Q4: BPMN Implementation Challenges*** *As previously mentioned, the BPMN roadmap generates a certain amount of overhead. For the described process to work, the project should be of sufficient size and budget. Otherwise, you can waste ~10% of the total time of the project to go through this roadmap. Sometimes you want quick/ad-hoc developments.* *Nevertheless, a common language for process mapping makes sense to be implemented. However, the coordination to implement it at ABC can be challenging. Different teams work in a certain way; asking people to do things differently causes a lot of overhead.* *For example, when people were asked to transition from Microsoft Planning to Smartsheets, there were many pros and cons raised by different people. I believe this goes for any implementation that we try to do.* *Therefore, gaining a "critical mass" for the things to move forward will be the biggest challenge for the BPMN roadmap.* ***Q5: ABC Processes Retrospective*** *More had to be invested in the coordination between different departments and teams. For example, in R&D, we have a lot of flexibility, and introducing a tool that works for Production might not be suitable for us. Therefore, coordination and negotiations had to take place earlier.* *In Conclusion: The BPMN Roadmap should be implemented slowly, part by part. For example, we lack the quality department, which the BPMN Roadmap advocates, but this comes slowly also with the organization scale-up.*
Role: Architect, Data Analysist, Embedded Software Engineer. Department: Research and Development Experience: 7-10 years Most Tension: Costs Least Tension: Measuring Project Status	**Q1: RTI Framework** *The framework describes very well the process we went through at ABC when we tried to introduce the Redmine RT tool. However, we were missing the last two steps of categorizing the existing business processes and analyzing the existing process challenges.* *For ABC, the BPMa lifecycle is not that important yet. But if the product matures, we have to pay closer attention to it.* ***Q2: BPMN Diagram*** *BPMN Roadmap will improve the aspect of Documentation, however many other challenges such as "Frequent requirements change control", "Measuring project status", "Tracing across organizational boundaries", and others are not properly*

	addressed by the diagram and I think will remain unchanged if we follow the BPMN Roadmap processes. ***Q3: BPMN Diagram Improvements*** *In the case of ABC, the main product is a combination between photonics, hardware, and software. The BPMN roadmap does not provide a clear representation of how these requirements should be traced between the three layers of our product.* *Furthermore, the process to introduce a new requirement during the "Sprint run" or "Daily Scrum" is missing in the BPMN roadmap. The frequent requirements change during Sprint runs is one of the most challenging aspects we have to deal with at ABC.* ***Q4: BPMN Implementation Challenges*** *In the first place, implementing a common business process modeling language is a challenge. For example, the production/manufacturing teams could benefit from something like this because the processes are more structured. But in the case of the R&D department, they should be more flexible in choosing what they want.* *Furthermore, having a common language requires training of certain people to operate the language correctly. As a result, if a change has to happen to the process, an employee has to go to the process gatekeepers to make the changes. This could increase the administrative burden.* *But in the long-term, having a standard modeling language will definitely increase efficiency for some departments and projects.* ***Q5: ABC Processes Retrospective*** *The management had to place more emphasis in the beginning to train people on the concepts of Design Innovation and Product development lifecycle. Many employees are not that familiar with the process and do not see the bigger picture for certain tasks and projects that should be done.* *Also, investment in different licenses, hardware, and tools to facilitate the employees' jobs could have drastically sped up the product development and quality of the product.*

Table 27: BPMN and RTI Framework Interview Results