

Agile Methodology Implementation in Lebanon

Scarlet Gaby Rahy



University of
Salford
MANCHESTER

UNIVERSITY OF SALFORD

SCHOOL OF SCIENCE, ENGINEERING AND ENVIRONMENT

JULY 26, 2021

This document is toward the degree of Doctor of Philosophy

Contents

1	Introduction	1
1.1	Problem Context	1
1.2	Research Motivation	3
1.3	Aims, Objectives, and Research Questions	4
1.4	Research Contributions	6
1.5	List of Publications	8
1.6	Thesis Structure	8
2	Literature Review	10
2.1	Introduction	10
2.2	Software Development Methodologies	11
2.3	Agile Software Development	12
2.3.1	Scrum	14
2.3.2	eXtreme Programming (XP)	14
2.4	Agile Concepts	15

<i>CONTENTS</i>	ii
2.4.1 Roles	16
2.4.2 Artefacts	16
2.4.3 Ceremonies	17
2.5 Agile Software Development Teams	17
2.5.1 Inter-team Communication	18
2.5.2 Trust	19
2.5.3 Inter-team Boundaries	20
2.6 Benefits and Challenges of Agile Implementation	21
2.6.1 Agile Method Perception	22
2.6.2 Customer Involvement	22
2.6.3 People Challenges	23
2.6.4 Inhibitors and Exciters of Agile Implementation	24
2.7 Agile Tailoring	26
2.7.1 Regulated environments and Agile Tailoring	27
2.7.2 Non-functional Requirements and Agile Tailoring	28
2.8 Software Development Industry in Developing Countries	29
2.8.1 Software Development Industry in Lebanon	30
2.9 Summary	34
3 Research Design	36
3.1 Introduction	36

CONTENTS

iii

3.2	Research Model	37
3.2.1	Research Philosophy	38
3.2.2	Research Logic	39
3.2.3	Methodological Choice	40
3.2.4	Research Strategy	40
3.2.5	Time Horizons	41
3.3	Research Purpose and Research Outcome	43
3.4	Research Approach	44
3.5	Research Sites	45
3.5.1	The Netherlands: DevelopCo	45
3.5.2	India: HealthCo	47
3.5.3	Lebanon: LebDevCo1, LebDevCo2, LebDevCo3	49
3.6	Data collection (Phase 1)	54
3.6.1	Recruiting Participants	55
3.6.2	Transcribing Data	56
3.6.3	Observations	56
3.7	Data analysis: Within-Case Analysis based on Grounded Theory (Phase 2)	57
3.7.1	Open Coding	58
3.7.2	Constant Comparison Method	59
3.7.3	Memoing and Sorting	60

3.7.4	Theoretical Coding	60
3.8	Data Analysis: Cross-Case Analysis (Phase 3a)	61
3.8.1	Chosen Approach to Conducting Cross-Case Analysis	64
3.9	Model creation process (Phase 3b)	66
3.10	Agile Tailoring to the Lebanese SME Context (Phase 4)	69
3.11	Summary	70
4	Approaches to Agile Implementation	72
4.1	Introduction	72
4.2	Agile Implementation in DevelopCo	73
4.2.1	DevelopCo Process Flow	73
4.2.2	Geographically Separated Inter-Team Collaboration in relation to Agile Implementation	74
4.2.3	Communication Between Practitioners	76
4.2.4	Team and Customer Collaboration	78
4.2.5	Inter-team Communication Tools	80
4.2.6	Approach to Non-Functional Requirements	83
4.3	Agile Implementation in HealthCo	84
4.3.1	HealthCo Process FLOW	84
4.3.2	Employee Training and Development	85
4.3.3	Communication Between Practitioners	86
4.3.4	Team and Customer Collaboration	87

CONTENTS

v

4.3.5	Inter Team Communication Tools	88
4.3.6	Approach to Non-Functional requirements	89
4.4	Agile Implementation in Chosen Research Site in Lebanon	90
4.4.1	Description of the Baseline in each Research Site	91
4.4.2	Overview of Agile Implementation	94
4.4.3	Influences on Agile Implementation Exerted by Non-team Members	96
4.4.4	Agile Implementation in Relation to Ceremonies and Prac- titioners' Roles	99
4.4.5	Time Factor and its effect on Agile Implementation	105
4.4.6	Customer's Involvement in the Agile Process	107
4.4.7	Agile in Regulated Environment	110
4.4.8	Approach to Non-functional requirements	112
4.4.9	Factors that Require Improvement	112
4.4.10	Inhibitors and Exciters of Agile Implementation	114
4.4.11	Influences on Agile Implementation in Lebanon	116
4.4.12	LebDevCo1 Process Flow	118
4.5	Summary	119
5	Cross Case Analysis Findings, Model Creation, Agile Tailoring to the Chosen Lebanese SME Context	120
5.1	Introduction	120
5.2	Cross-Case Analysis Findings	121

5.2.1	Interaction and Information Flows	123
5.2.2	Inter-Team Communication	125
5.2.3	Approach on the Handling of Non-functional Requirements	126
5.2.4	Customer Involvement	127
5.3	Model Creation for Agile Processes in Lebanon	128
5.3.1	Gap between Actual Implementation and Designed Agile	128
5.3.2	Conceptual Mapping	129
5.3.3	Model for Lebanese SME	131
5.4	Intervention Findings	131
5.4.1	Intervene to Overcome the Lack of Agile Ceremonies . . .	133
5.4.2	Intervene to Reduce the Lack of Agile Roles	136
5.4.3	Intervene to Overcome Task Estimation and Allocation by Managers	138
5.4.4	Intervene to Reduce Employee's Fear	141
5.4.5	Intervene to Enhance Agile Understanding	141
5.4.6	Intervene to Improve Lebanese Customer Involvement . .	142
5.4.7	Actions to Mitigate the Implementation of Agile in the Global South Context	145
5.5	Agile Model Tailored to the Lebanese Context	145
5.6	Post-Implementation Evaluation and Analysis	148
5.6.1	Conceptual Map: Post-Model Implementation	148

5.6.2	Gap between Actual Implementation and Designed Agile: Post-Model Implementation	150
5.6.3	Inhibitors and Exciter of Agile Implementation: Post- Model Implementation	154
5.7	Summary	155
6	Discussion	156
6.1	Introduction	156
6.2	Answering the Research Questions	156
6.3	Contribution to Theory	169
6.4	Contribution to Practice	173
6.5	Limitations	175
7	Conclusion	178
7.1	Summary of Thesis	178
7.2	Reflection	181
7.3	Contribution Summary	182
7.4	Future Work	184
	References	186
	Appendix	204
A	Appendices	205
A.1	Ethical Approval Form	206

CONTENTS

viii

A.2 Interview Guide	207
A.3 Communication Tools used by Practitioners	210
A.4 Artefact List at LebDevCo1	211
A.5 Daily Scrum Meeting Time Records	212

List of Tables

2.1	Role, artefacts and ceremonies	15
2.2	Main supporters to the Lebanese software development industry .	31
3.1	Overview on DevelopCo participants	46
3.2	Overview on HealthCo participants	48
3.3	Overview on LebDevCo1 participants	51
3.4	Overview on LebDevCo2 participants	53
3.5	Overview on LebDevCo3 participants	54
3.6	Elements identified in case-based comparison	63
4.2	Factors identified for improvement by practitioners	113
5.1	Influences in Lebanese organizations cross compared with HealthCo and DevelopCo	122
5.2	Cross case comparison of agile concepts used in DevelopCo, HealthCo, and LebDevCo1	124
5.3	Design reality gap	129
5.4	Three questions in the tailored retrospective	136

5.5	Customer collaboration evaluation criteria	143
5.6	Design reality gap model post-implementation	151
5.7	Time gain upon stand-up implementation	153
6.1	Illustrative example of using agile methods to deal with documen- tation work items	159
6.2	Illustrative example of using agile methods to deal with safety critical work items	160
6.3	External influences that impede agile implementation in the Lebanese context	162
6.4	Internal influences that impede agile implementation in the Lebanese context	164
6.5	Influences that amplify agile implementation in the Lebanese context	165
A.1	Communication Tools used by Practitioners	210
A.2	Daily Scrum Meeting Time Records for Team A; Day: 1	212
A.3	Daily Scrum Meeting Time Records for Team B; Day: 1	212
A.4	Daily Scrum Meeting Time Records for Team A; Day: 40	213
A.5	Daily Scrum Meeting Time Records for Team B; Day: 40	213

List of Figures

1.1	Thesis's phases illustration	5
2.1	Technology Institutions and Capabilities model [23]	25
2.2	Geographical distribution of software development companies across Lebanon [165]	32
3.1	Revised Research Onion [151]	37
3.2	Research classification using the Onion Framework	42
3.3	Grounded Theory components	57
3.4	Iterative analysis	59
3.5	Cross-case illustration	61
3.6	Cross-case analysis informed by Cruzes's approach [56] that is based on Miles and Huberman's approach [112]	65
3.7	Summary of methodological issues in this study	70
4.1	Swim lane diagram portraying the process flow of DevelopCo . . .	74
4.2	Swim lane diagram portraying the process flow for HealthCo . . .	85
4.3	TIC lens adapted to the Lebanese context	115

4.4	Influences on agile implementation in Lebanon	117
4.5	Swim lane diagram portraying the process flow for LebDevCo1 . .	118
5.1	Conceptual mapping of impediments faced in relation to the agile methods used	130
5.2	Model for agile implementation in Lebanon	131
5.3	Process of final model development	132
5.4	Tailored retrospective	135
5.5	Agile estimation techniques	139
5.6	Risk mitigation tagging technique	140
5.7	Customer collaboration spectrum	144
5.8	Phases followed to increase customer collaboration	144
5.9	Agile model tailored to the Lebanese context	147
5.10	Conceptual mapping post-intervention	150
5.11	Difference observed during Daily Scrum Meetings between the first and the last week of the intervention	153
5.12	TIC model post-implementation	155
6.1	Agile model tailored to the Lebanese context	172

Acknowledgement

I give thanks to, O Lord my God, with my whole heart and I will glorify your name forever. Psalm 86:12

I wish to express my deepest gratitude and affection to my parents Gaby and Violette Rahy for supporting and encouraging me every step away; my husband, George Abi Chebel, for being my pillar of strength, supporter, and best friend; and my sisters, Stephanie Rahy and Melany Rahy for being my number one fans.

I am extremely indebted to my supervisor Dr. Julian M. Bass for teaching me, challenging me, encouraging me, drawing out the best in me, and most of all, for always having faith in me through my ups and downs. I also would like to thank my local advisor Dr. Bassem Bassil for his continuous unconditional support.

Finally, my sincere and profound thankfulness to all those who supported me in any aspect during this process, and they are numerous. I am thankful for all research sites and practitioners in the field who contributed to this study. Though I do not mention you all here, I hold you in high esteem.

DECLARATION OF ORIGINALITY

I hereby declare that the research and work in this PhD report are original; from my own study and work. All sources of information used for this study have been referenced or mentioned in the acknowledgment section.

I certify that, to the best of my knowledge, this report does not infringe upon anyone's copyright. All ideas, techniques, and quotations used from other people's work has been fully acknowledged using references. This report has not been previously submitted to obtain any form of academic degree at the Salford University or any other university.

I further declare that I have dedicated full time to study this PhD topic under the guidelines of my supervisor and rules of Salford University.

Scarlet Gaby Rahy

ABSTRACT

Agile methods are being adopted in the software development industry increasingly and even beyond this sector. Outsourcing software development from developing countries in the Global South is increasing in the industry. With Lebanon being the hub for outsourcing software for the MENA region, practitioners require guidance to overcome challenges upon agile implementation taking into consideration the Lebanese context. The literature lacks information on agile software development in the Global South, specially Lebanon. The aim of this thesis is to add new knowledge on agile software development industry in Lebanon through studying the extend of agile methodology implementation, agile tailoring, and impediments faced when implementing agile.

The methodology implemented is a mixed method qualitative research divided into four phases involving 49 practitioners from 5 different software organizations. The first phase describes the data collection strategy that involves interviews and observations. The second phase involves the data analysis in a within-case analysis using an approach informed by Grounded Theory. The third phase is divided into two parts. Part A involves a cross case analysis comparing the findings from all research sites. Part B involves the model creation. The last phase is the intervention study conducted at the chosen Lebanese software development site.

This thesis describes the status of agile software development in the chosen Lebanese sites through creating a baseline model. This model uses the swim lane diagram to describe the process of agile software development in one iteration while highlighting the ceremonies, roles, and activities conducted. This thesis also identifies a collection of 22 influences, out of which 10 are novel influences, that affect the practitioner's ability to implement agile in Lebanon. Of these three are novel amplifiers: level of information sharing, overcoming geographical distances, and the need to succeed. Also seven are novel impediments: extensive micro management by the CEO and upper managers, the reasons behind the lacking of agile ceremonies, task estimation done by managers in agile, employee's fear, lack of respect to time, the effects of the economic crisis on daily routines, and nepotism.

In addition, the thesis corroborates 12 influences on agile implementation in the chosen Lebanese research sites to the existing literature.

The study then applies a cross-case analysis to compare agile implementation in all studied research sites in The Netherlands, Kenya, and India. This analysis reveals two approaches to agile implementation characterized by either a pure implementation of agile methods or mixing agile with traditional methods. The study then introduces illustrative examples demonstrating a step-by-step process to tackle non-functional requirements using agile methods and introducing two new artefacts.

Then, this thesis creates a novel tailored agile implementation model to fit the Lebanese context. The model tackles the identified influences using agile practices that are tailored when needed. The model is finally validated by engaging with practitioners in the Lebanese research site causing positive changes. These include interventions to overcome task estimation and allocation by managers, reduce employee's fear, enhance agile understanding, enhance collaboration through introducing the previously lacking agile roles and ceremonies, actions to improve Lebanese customer involvement, and actions to mitigate the implementation of agile in the Global South context.

Chapter 1

Introduction

1.1 Problem Context

Agile software development is spreading extensively in the software development industry, where companies are shifting from traditional methods to agile [158]. Under agile, software development companies are able to adapt and respond to the rapidly changing customer needs [92]. Agile enhanced the ability to tolerate continuous change and increase product innovation [60]. Agile's successful implementation is coupled with a decrease in project duration and cost. Thus, agile projects are expected to yield results in areas of productivity and effectiveness [13]. Originally, agile methods targeted small software development projects; but now they are increasingly applied to large scale software development projects [59]. Agile was even recently applied to fields other than software engineering [27, 146].

Agile principles revolve around satisfying the customer through early and incremental delivery of valuable software. With this rises the need for communication among the different stakeholders, 'business people and developers' [25]. Thus, roles, artefacts and ceremonies are three interlinked concepts that work to produce a working software.

Self-organizing teams play a vital role in agile software development. They are seen as the spirit of agile [159, 91]. Agile teams should sacrifice a level of autonomy [20] to achieve coordination and collaboration, which are essential for their success [65, 106, 150]. In addition, the inter-dependencies between tasks increase the project complications. As a result, communication between teams and across the different boundaries becomes vital for the success of any project. These include inter-team boundaries, team and customer boundaries, and product owner and developers boundaries. Here rises the role of boundary spanners who act as coordinators [173], mediators, and facilitators of internal communication [77]. Failure to achieve constant communication leads to mistrust between the team members and consequently affects the work flow [164].

The tailoring of agile software development methods came after the struggles faced by large-scale projects. Tailoring agile methods aided in spreading the implementation of agile in the software development industry. With regulated environment, tailoring agile methods improved the implementation process [76]. Also, literature has developed initial ways to integrate non-functional requirements in the agile software development process.

Research on agile in the Global North has dominated the literature. There is a paucity of research literature on agile understanding and implementation in the Global South. It is limited to few articles on software development and reports produced by NGOs, banks, UN organizations, and governmental institutions. With agile concepts being developed initially in the Global North, a question rises on the ability of practitioners in the Global South to perceive these concepts and implement them effectively as they are.

Following the world trend, Lebanon has witnessed an increase in investment in the software industry. With the growing demand for software development, companies providing this service in Lebanon have evolved from offering traditional services in banking, education, and healthcare sectors, to innovation and content generation services [168].

The market in Lebanon offers a competitive advantage for three main reasons.

First, Lebanon is ranked fourth in the quality of maths and science education. This, coupled with high adaptability skills, supplies the market with skilled labour [167]. Second, effective communication skills with knowledge of a minimum of three languages differentiate Lebanese talents [96]. Third, with the current economic crisis and currency devaluation, the market is able to offer competitive rates.

Lebanon has been the leading outsourcing hub for the MENA region [152]. With the rise of agile as a method to enhance software development, companies providing this service are implementing such methods [9]. Little is known on the factors influencing agile software development in Lebanon. Thus, challenges are faced with no guidance, specifically dedicated to the Lebanese context, available to overcome them .

1.2 Research Motivation

Outsourcing software to developing countries is increasing among customers. These countries include, but are not limited to, Ukraine, India, Malaysia, Thailand, and Lebanon. As in the global market, the demand for software development in Lebanon is rising. Up to 75% of software development SME's revenue is generated from export of software to foreign markets [168]. Software development providers in Lebanon aim to enhance the quality and productivity of their development process to be able to compete in the market.

The popularity of agile methods is increasing within the software development industry [28, 5]. Agile methods meet the evolving requirements and enhance productivity and flexibility [66]. Benefits of agile methodology are also visible to the customer through enhanced quality and strong relation between the customer and software development company [158].

The human factor lies in the centre of agile methodologies and represents the core for its success [90, 159]. Thus, the implementation of agile differs from one context to another. These settings include experience, culture, environment, level

of expertise, and gender [159]. With agile, the manager becomes a facilitator rather than a controller and in turn, developers are empowered. Studies show that developers are mostly satisfied by agile methods since they feel more engaged in the process [66]. When developers are engaged in the process, software development productivity increases [13]. Organizations and practitioners in Lebanon are deficiently implementing agile methods, thus losing the previously stated benefits of agile. The study will look into the extent of agile methodology implementation, the impediments faced when implementing agile, and agile tailoring to fit the Lebanese context in the chosen Lebanese research sites.

1.3 Aims, Objectives, and Research Questions

This research aims to improve empowerment perceived by selected practitioners using agile software development methods in Lebanese small and medium-sized enterprises (SMEs).

To achieve this aim, the following research questions are identified:

1. **Research Question 1** How do selected practitioners from software companies describe the different approaches to agile tailoring of roles, ceremonies, and artefacts?
2. **Research Question 2** How do selected practitioners describe the current implementation of agile in Lebanese SMEs, and how does this compare with the practitioners investigated in Research Question 1?
3. **Research Question 3** What are the selected practitioners' perceptions of the effect of tailoring agile methods in the Lebanese SMEs on improved empowerment?

To achieve the aim of the study, objectives are identified in the form of study phases. An illustration is presented in figure 1.1.

Phase 1: Collect data from the selected research sites.

Phase 2: Determine the characteristics of agile software development in selected research sites. This phase is presented in chapters 4 (DevelopCo in The Netherlands and HealthCo in India) and 5 (LebDevCo1, LebDevCo2, and LebDevCo3 in Lebanon).

Phase 3: Phase 3 is divided into two sections: **3a:** Apply a cross-case analysis between the selected research sites. **3b:** Develop an agile software development model tailored for Lebanese practitioners

Phase 4: Evaluate and update the model by investigating its application in the selected research site in Lebanon and critically examine the issues that influence the implementation of the different agile concepts in an intervention.

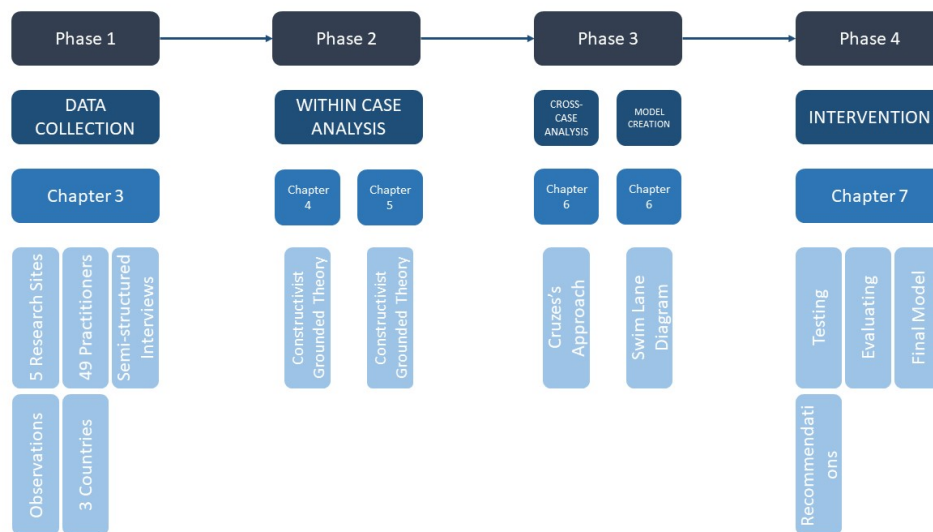


Figure 1.1: Thesis's phases illustration

1.4 Research Contributions

The thesis contributes to knowledge on agile software development implementation in the studied sites in Lebanon. The thesis identifies the influences that were addressed using a tailored agile software development model. Below is a list of this thesis's contribution:

- As a result of conducting a detailed comparison and analysis of two software development companies in India and The Netherlands, I observed two approaches to handling non functional requirements. On one hand, a Dutch software development company uses artefacts (user stories) in the agile methodology to manage non-functional requirements. On the other hand, the Indian healthcare company persists on using conventional plan-based methods for managing non-functional requirements. After careful analysis, I propose new work practices designed to integrate stringent non-functional requirements into an agile setting. This resulted in practical implementations comprising two novel artefacts, Documentation Work Item and Safety Critical Work Item, accompanied by illustrative examples.
- In the second stage of the research, the analysis of the findings produce a baseline model that encapsulates 22 influences to agile implementation in the studied Lebanese sites. The contribution of the thesis is to identify this collection of influences that characterize software development in the studied Lebanese sites that adversely affect the productivity. Of these influences, ten are novel. The five newly identified internal negative influences are extensive micro management by the CEO and upper managers, the reasons behind the lacking of agile ceremonies, task estimation done by managers in agile, employee's fear, and lack of respect to time. The three newly identified positives influences are level of information sharing, overcoming geographical distances, and the need to succeed. Two newly identified external negative influences are the effects of the economic crisis and nepotism on daily routines of software development teams. This research presents a new framing for agile tailoring by discussing it in the Global North and Global South

perspective. Instead of focusing on project characteristics such as large scale agile, regulated environments, and organizational structure, this research looks into agile as a methodology created in the Global North and propagated in the Global South, and analyses how it can be tailored to fit the context it is implemented in. Thus, the development process ceremonies and roles are tailored to fit a specific context through empowering self-organizing teams and reducing hierarchical management models.

- After thoroughly studying the evidence of this research, I created a novel agile software development model tailored to the agile context that addresses the influences identified. Some agile methods were implemented as-is while others had to be tailored to fit the Lebanese context. This model was created using swim lane diagrams as a mean to represent the processes and information flows in agile software development. This idea came as a mix between the researcher's industrial engineering background and the need to visually identify bottlenecks through the display of roles, responsibilities, and sequence of processes. Several versions of this model were created after each iteration before reaching the final designed tailored agile process.
- In this study, the model was tested by applying it to selected teams in a Lebanese software development company. This resulted in a refined tailored agile software development model. Practitioners reported positive change in the processes previously conducted in the chosen research site. These include interventions to overcome task estimation and allocation by managers, reduce employee's fear and enhance agile understanding and collaboration through introducing the previously lacking agile roles and ceremonies, actions to improve Lebanese customer involvement, and actions to mitigate the implementation of agile in the Global South context. The TIC model is used to assess the practice of agile methods in the organizational context at two different instances in the study. It is also used to represent the inhibitors and excitors affecting agile implementation in the baseline state and the final state. The comparison between the two TIC models showed the effect on inhibitors, excitors, virtuous circles, and viscous circles.

1.5 List of Publications

As a consequence of the research conducted in this study, the following papers have been produced:

- Rahy, S., and Bass, J. (2020). Implementation of agile methodology in developing countries: case study in Lebanon. IFIP WG9. 4 European Conference on the Social Implications of Computers in Developing Countries. International Federation for Information Processing (IFIP). (In-press) [135]
- Rahy, S., and Bass, J. (2020). A post-colonial analysis of agile software development methods in ICT4D. IFIP WG9. 4 European Conference on the Social Implications of Computers in Developing Countries. International Federation for Information Processing (IFIP). (In-press) [134]
- Rahy, S., and Bass, J. (2019). Overcoming team boundaries in agile software development. Journal of International Technology and Information Management. (In-press) [133]
- Rahy, S., Bass, J. (2018, October). Information flows at inter-team boundaries in agile information systems development. In European, Mediterranean, and Middle Eastern Conference on Information Systems (pp. 489-502). Springer, Cham. [132]
- Rahy, S., Bass, J. (2020). Managing Non-functional Requirements in Agile Software Development. IET Software (Accepted)

1.6 Thesis Structure

The rest of this thesis is as follows:

Chapter 2 Literature Review presents a detailed description of the related literature. Agile software development methods are described along with the related

concepts, teams, challenges, benefits, and tailoring. Then, agile implementation is described in developing countries, focusing on Lebanon.

Chapter 3 Research Design describes the research model, followed by a detailed description of the research approach and all the qualitative methods used.

Chapter 4 Approaches to Agile Implementation reports agile strategies, cross-boundary interactions, and approach to non-functional requirements at DevelopCo and HealthCo, and provides an extensive and detailed view on agile implementation in the chosen research sites in Lebanon.

Chapter 5 Cross Case Analysis Findings, Model Creation, and Agile Tailoring to the Chosen Lebanese SME Context compares and contrasts the similarities and differences between the information flows of the studied companies and their relation to communication, cross-boundary interactions, challenges and non-functional requirements. The initial model is presented. This chapter also presents the intervention findings in the chosen research site in Lebanon, the final refined model and a post-implementation evaluation and analysis.

Chapter 6 Discussion answers the research questions. In addition, this chapter explains the contribution to theory, contribution to practice, and the limitations.

Chapter 7 Conclusion summarizes the thesis, presents the reflection, restates the contributions of this thesis, and suggests future work.

Chapter 2

Literature Review

2.1 Introduction

This chapter aims at introducing the reader to the previous literature conducted on agile software development. Any substantial and useful research demands a thorough literature review that demonstrates a grasp of the field being researched [37]. It presents the theoretical information needed to direct the research. The literature was collected from scholastic and academic material including, but not limited to, books, articles, journals, reports, and conference proceeding. The analysed information builds-up the theoretical framework and identifies research gaps.

This chapter serves the above mentioned purposes by first presenting an overview on software development methodologies. Then, agile software development models (Scrum and eXtreme Programming) are introduced while highlighting the key concepts of roles, artefacts, and ceremonies. The agile teams are then examined by analysing the inter-team communication, investigating trust needed to achieve success, and exploring inter-team boundaries. Further, this chapter introduces the benefits and challenges of agile methods while focusing on the challenges related to

perception of agile method, involvement of customers, and difficulties encountered by people. Also, agile tailoring is examined in regulated environments and when dealing with non-functional requirements. Finally, this chapter studies agile in developing countries while focusing on Lebanon's case.

Overall, this chapter contributes to the thesis by justifying the necessity of examining agile software development methods in Lebanon where research is lacking. Also, this chapter assesses the potential ways agile implementation may be evaluated in terms of concepts, stakeholders, and challenges encountered. It also gives grounds for conducting new research under different theoretical perspectives.

2.2 Software Development Methodologies

Software engineering trends tend to change significantly and rapidly due to their evolving nature [35]. Literature organizes the software development methodologies into three main groups: the traditional life-cycle, the iterative or incremental, and the component based development [149, 144]. There are several software development life cycle models created to organize, structure, and process the various activities of software development. They follow a linear approach to development. These include the Waterfall, Unified, V model, and W model [149]. Iterative-incremental models base their development on continuous customer involvement in the development process [144]. Agile methods follow this approach [162]. As for the component based development, it is concerned with a reuse-based approach. With commercial programs such as DCOM and COBRA, this method has increased its popularity [149].

With the current evolving business environment, the software development industry is in constant need to adapt its policies and methodologies to the new environment [35]. Contrary to the traditional and plan-driven methodologies, iterative and incremental methodologies provide adaptability and flexibility needed to face the projects that exhibit high variability in tasks, technology, and capabilities [118].

Agile methods supply the software development process with such characteristics [26, 67, 3, 110, 48].

2.3 Agile Software Development

A central list of agile principles was introduced to the software development industry in the Agile Manifesto in 2001. Agile software development is growing deeply within the same organization and horizontally across different organizations [1]. It has increasingly become an essential component to overcome competition; and with globalization, organizations expanded their development teams to different geographical locations [88]. Thus rises the need to study the challenges and practices of information sharing in geographically distributed agile software development teams [69] and the way agile implementation differs across diverse geographical locations.

Methods of agile software development have emerged as a response for the weakness of traditional methods. Agile methods are based on the values and philosophies developed in the Agile Manifesto [26]. Quoting the agile manifesto, agile methods favour:

*“Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan”*

The principles behind the agile manifesto promote continuous customer involvement and feedback, delivery of a working software, self-organizing teams, effective communication within and across teams, and fast response to change requirements. The goal of agile application lies in adaptability, flexibility, and responsiveness [66].

The software development process is characterized by high speed and evolving business requirements. Thus, rises the need to create a method that responds to change rapidly. With agile methods, people as well as organizations, aim to be responsive. Agile advocates the replacement of documents with white boards, plan-based methods with incremental progress of work, and hierarchical organizational structure with fast decision making [49].

The main new idea that agile introduces is the recognition of people as main drivers in the software development process [89]. Individuals in agile are given a level of autonomy to form self-organizing teams. The different personalities and skills individuals hold affect the team building process. These agile project teams focus on increasing both individual competencies and collaboration levels among individuals [49]. With agile methods originating in the Global North, the expected behaviour of individuals is related to the background and cultural context in the Global North.

Agile methods represent a 'northern' idea being propagated and adopted in the Global South. Thus rises the need for agile tailoring. The research around agile tailoring is predicated on the assumption that methods must be adapted to the local context [40]. Conventionally, agile method tailoring tends to focus on factors such as business domain, technology stack, and project characteristics (such as project scope, project size, and disposition of teams). Research tends to disregard the adaptation to the cultural context in which agile is being implemented. Section 2.7 presents further details on agile tailoring.

Agile uses experimentation and introspection as means to adapt to the constantly changing world [66]. This is possible through software development methods that include Dynamic System Development Methods, Lean Software Development, Scrum, and Extreme Programming [131, 3, 67]. Lean software development is almost as old as the concept of lean itself. Lean software development is looked at as a set of principles that may be applied to software development leading to quality and process improvement [130]. The below sections introduce scrum and XP, the most widely adopted agile methods [129].

2.3.1 Scrum

Scrum is one of the most adopted agile methods [129] named by the ‘method of use’ by 52% of the respondents [1]. It was developed by Jeff Sutherland and Ken Schwaber. Scrum is characterized by having few concepts and rules yet sets out roles and responsibilities clearly [122]. Scrum is based on high levels of cooperation between team members while focusing on delivering the highest priority business value [153]. Developers under Scrum build several continuous intermediate products that build up the final end product [122]. Planning in Scrum is done in three occasions: daily Scrum planning, sprint planning, and product release planning. A sprint is one iteration designed to deliver work usually lasting 2 to 4 weeks. Scrum holds certain principles:

- Teams are self-organized and consist of a scrum master, product owner and around five members from the development team.
- Self-organizing teams work under the Pull principle.
- All work load is set to be completed in a pre-defined time frame (sprint).
- At the end of each sprint, team members deliver a code or a potential deliverable code.

2.3.2 eXtreme Programming (XP)

eXtreme Programming is an agile method mostly developed by Kent Beck [24]. XP inherits the characteristics of agile while focusing on automated testing, constant integration, and small releases. XP encourages continuous rapid customer feedback resulting in an effective and flexible approach to software development. The development process focuses on the functional requirements that add business value. XP follows the bellow set of values [24]:

- Communication is advocated between programmers, customers, and managers. Pair programming is an XP practice that encourages communication channels.
- Simplicity of design is a key requirement in XP. Time and effort to finish ‘the simplest thing that could possibly work’ is much less than the time and effort needed to finish a complex design.
- Feedback is essential to create momentum in XP. Feedback is exchanged between customers and programmers.
- Courage is a requirement to maintain a clear development process and to change or fix designs while maintaining development speed.
- Respect is essential to maintain communication and collaboration between programmers, managers, and customers.

2.4 Agile Concepts

Conventionally, agile methods are comprised of roles, artefacts, and ceremonies. Table 2.1 shows a list of the roles, artefacts, and ceremonies. This section discusses them in further details.

Roles	Ceremonies	Artefacts
Self-Organizing Team Scrum Master Product Owner	Daily Scrum Sprint Planning Demo Retrospective	Feature Sprint Release Product Development Programmed Governance

Table 2.1: Role, artefacts and ceremonies

2.4.1 Roles

Roles include scrum master, product owner, and teams. The **scrum masters** act as a liaison between the team and the product owner. They are responsible to identify, acknowledge, and remove any impediments that hinder the software development process. The activities conducted by a scrum master are identified as [17]: Process Anchor, Stand-up Facilitator, Impediment Remover, Sprint Planner, Scrum of Scrums Facilitator, and Integration Anchor. **Self-organizing teams** are the hallmarks of agile methodology [90]. Teams are cross-functional and responsible for identifying and attaining the sprint goals within the project boundary [143]. **Product owners** perform activities to manage scale, distance, and governance. They are responsible for defining stories and prioritizing the backlog.

2.4.2 Artefacts

Artefacts have been identified by researchers in different ways. An all-inclusive list of artefacts was identified by investigating all the development cycle resulting in identifying 25 artefacts that fall under five categories: programme governance, product, release, sprint and feature [19]. Programme governance artefacts provide a layer of governance and oversight to mitigate risk and enhance inter-team coordination. These include risk assessment and contracts. Product artefacts are produced at the product level such as the product backlog and user acceptance tests. Process artefacts are applied at the process release level such as the regression tests. Sprint artefacts are concerned with the development process in each sprint. They include the sprint backlog, burn down chart, and user story estimates. Finally, the feature artefacts describe specific aspects of certain features such as the user stories and testing criteria.

2.4.3 Ceremonies

Ceremonies provide communication opportunities between the different roles in agile. **Daily scrum meetings** or daily stand-ups allow team members to discuss their work on a daily basis. The scrum master organizes this meeting. Team members answer three questions: What did I do yesterday? What will I do today? What impediments are in my way? **Sprint planning meetings** are led by the product owner and dedicated to plan the coming iteration. The team will set the goals for the next sprint taking into consideration any updates presented by the scrum master. A **Demo** is a session done at the end of the sprint during which the team demonstrates the features developed. The product owner will either approve or reject the team's work. **Retrospectives** are meetings held at the end of the sprint during which team members identify three positive and negative incidences that occurred throughout the sprint. Team members develop action plans to overcome the obstacles.

2.5 Agile Software Development Teams

Agile software development focuses on people interaction and social communication. The Agile Manifesto focuses on the importance of agile teams and promotes a people-focused view of agile software development [25]. Self-organizing teams are seen as the spirit of Agile, enabling the social and human aspect of software engineering. The Agile Manifesto encourages effective communication through motivating and supporting such kind of teams [25]. Communication is essential for the implementation of agile practices, for instance, the daily stand-up meetings with 90% usage, followed by sprint planning with 88% usage, and thirdly by retrospectives with 85% usage [1].

Communication, feedback, coordination and collaboration are key characteristics for a self-organizing agile team [65]. The Agile Manifesto grants all team members the opportunity to participate in the decision making process [26] through striving

towards consensus-based decisions with opinions from all team members being valued [90]. This creates obstacles since some team members show reluctance in decision making, and may consider it as a burden rather than a privilege and rely on the scrum master for taking decisions [65]. Such people-related challenges were tackled by Nerur et al. A team that follows a collaborative decision making approach requires effort and patience and is faced with a lot of challenges. The authors suggest that the traditional role of the project manager should evolve to reach a collaborator and facilitator role instead of a planner and controller role [118]. In addition, smaller teams are better suited for such a decision making process [118], since large teams are faced with additional challenges during this process.

One of the four main themes of concern in agile implementation is inter-team coordination or dependency [58]. Inter-dependencies between the different tasks of software development may be difficult to organize or prioritize [121, 11]. The number of teams and developers involved along with the workload and dependencies make team coordination and networking behaviour essential to achieve project success [104, 173]. With the increase in dependent tasks, communication on mutual obligations increase. Here, team members are faced with different perceptions that might create a possible decline in trust [114]. In addition, communication barriers rise with mutual mistrust among the different roles [79]. Such dependencies and criticisms can be facilitated by the presence of a boundary spanner [173].

2.5.1 Inter-team Communication

The global software development research tracks three main areas in distributed collaboration: coordination, temporality, and communication. Our study focuses on communication and the importance of developing a collaborative nature between geographically separated teams and among members of the same team. Extensive research on agile software development principles and practices is known [4, 9, 18, 95, 128], but less is done on inter-team communication. Inter-team knowledge sharing in agile software development is still in the rise [59, 173], and

is identified as an important topic in research. Practices that are applied specifically for knowledge sharing in agile software development are still under study. Previous research has investigated inter-team communication tools but our research addresses the information sharing challenges at team boundaries and investigates how to overcome them.

2.5.2 Trust

Agile teams cannot work in isolation, thus inter-team communication is a necessity to achieve the success of agile software development [59, 90, 150, 173]. Self-organizing agile teams should sacrifice some level of autonomy [20]. The delivery of customer requirements needs to be the product of inter-team coordination [20]. Failure to achieve effective communication in agile development process leads to failed patterns in delivery. A key challenge for communication among teams is building trust. Establishing trust is important all through the development process but is highly essential in the early stages of the process for example during kick-off meetings [164]. Such gaps in communication may lead to misleading or low quality information. This may cause the formation of false impression, usually negative, formed on other team members or across different teams. This results in low productivity. Thus, inspecting communication means and ways to enhance them is a necessity especially in software development to increase productivity [6].

It is a necessity that the different stakeholders, in an agile implementing environment, trust and respect each other [111]. Trust cannot be created; rather it is a process that develops and grows over time through repeated interaction and shared experience [164]. Agile methods, such as pair programming and collective code ownership, require the presence of trust. Consequently, communication practices in agile enhance and work on building this trust. These practices include retrospectives, daily stand-ups, and sprint planning. Open communication with the team, frequent knowledge sharing across teams, and honest feedback in retrospectives are all factors that have a direct impact on trust across teams [111]. Awareness on team members' collaboration leads to making accurate attributions and trustworthiness

of each member [164].

2.5.3 Inter-team Boundaries

Inter-team information sharing has been identified as a key factor in agile software implementation by several researchers. Agile software development requires the transfer of information throughout several stages, which may lead to miscommunication or extensive communication paths [44]. The right combination of communication mechanisms improves agile implementation while a mismatch may be an impediment [129]. Communication becomes harder in large-scale development projects because of employee turnover, formation of new teams, and addition of new developers and technical experts especially in regulated environments [61].

To compensate this, agile development uses cross-functional teams and encourages open and direct communication. Dingsoyr et al. indicated that there are a number of arenas to achieve knowledge sharing and coordination in teams, and these change over time [62, 59]. Santos and Goldman developed a theoretical model that enhances the effectiveness of inter-team knowledge sharing through organizational conditions, such as team integration and agile methods adaptation, as well as stimuli that include motivation toward a common goal [150]. Smite et al. emphasized the importance of establishing a networking culture to enhance cross team interaction as a driver for better performance. Networking behaviour is affected negatively by factors including complexity and unfamiliarity of the tasks, frequent process changes, and cultural differences [173].

In a software development project, task uncertainty, task interdependence, and size of work unit produce difficulty in managing, identifying, and prioritizing tasks [121]. The work load and the number of developers and development teams make team coordination a key process indicator to the success of the project [104, 173]. There are several factors that negatively affect the team coordination such as complexity and unfamiliarity of the tasks, frequent process changes, and cultural differences [173]. In agile software development, teams work toward

consensus-based decisions; with opinions from all team members being valued [90]. Simultaneously, teams may require a collaborator [117] or boundary spanner [173] to coordinate inter-team dependencies.

The nature of agile software development produces boundaries between different actors such as, inter-team, team and customer, and product owner and developers [132]. Communication becomes a key challenge with the need to coordinate between different stakeholders [15]. This leads to the rise of boundary spanners as communication facilitators [33, 173]. Boundary spanners act as coordinators who provide a source of information, a target for feedback [173], a mediator between different teams [171], and a socio-material assemblage [64]. Boundary spanners don't only facilitate communication inside the organization but also help form organizational identities [77].

2.6 Benefits and Challenges of Agile Implementation

With the increasing complexity of software development projects, agile methods became progressively common. Agile methods rely on people and their creativity to address this complex nature, rather than relying on formalized processes [117]. Implementing agile methods is associated with increased productivity, employee satisfaction, high quality functional software, defect reduction, improved knowledge and learning, and innovative solutions [157]. The benefits were also visible to the customer through increased product quality and improved relationship with customer [158]. The literature presents a large number of case studies that highlight the benefits of agile implementation. These are associated with three critical success factors: correct delivery strategy, proper practice of agile software engineering techniques, and high-caliber team [46].

With numerous benefits and opportunities emerging from agile methodology rises a set of challenges. After thoroughly examining the related literature, agile challenges include management and organizational issues, people-related issues, process-

related issues, and technological issues [155, 117, 158]. This study focuses on the various challenges and groups them into three categories: agile method perception, customer involvement, and people challenges.

2.6.1 Agile Method Perception

One of the main challenges faced upon agile implementation is the understanding of agile concepts and the ability to implement them. Agile methods require a shift from the ‘command-and-control management to leadership-and-collaboration’ [117]. Practitioners tend to focus on practice and neglect the need to learn and understand the principles and values promoted in agile. The traditional methods and roles are no longer effective in the agile methodology. Thus, rises the challenge to find suitably trained graduates and create agile-specific training for employees [52]. Further, challenges rise in the ability to sustain agile and embed it thoroughly into the process [83].

Research on agile understanding and implementation in the Global North has dominated the research field. There is a paucity of literature on agile understanding and implementation in the Global South. With agile concepts being developed initially in the Global North, a question rises of practitioners’ ability in the Global South to perceive these concepts and implement them effectively. Section 2.8 presents the literature on agile software development in developing countries and looks specifically into Lebanon’s case.

2.6.2 Customer Involvement

Agile methods enhance customer involvement in the development process through encouraging regular customer feedback [90]. The customer is a person or a group of people who request a software and decide on its compatibility [31]. Agile practices, such as providing feedback and prioritization, advocate a high level of collaboration between the development team and the customer. The

identification and prioritization of customer requirements is a complicated process [18]. Customer feedback is considered by some team members as personal criticism and an offence. Thus, team members focus on self-defence rather than work adjustment and adaptation [90].

Inter-team collaboration is a base for agile software development [106, 150]. All team members are required to have a dynamic behaviour adaptable to the customer's requirements since customers are major players in the agile journey [106]. Selecting communication methods between teams and customers is crucial for the success of agile especially when the customers are partially available [101].

Inadequate customer involvement hinders the agile software development process. It causes problems in gathering data, attaining feedback, prioritizing requirement, and team productivity [94]. Hoda et al. revealed a spectrum of the customer involvement and presented a list of strategies named Agile Undercover to secure a level of customer involvement. These include changing customers' mindsets, providing alternatives in the contract signing process, using buffer zones, changing the priority of user stories, assessing risk of customer involvement, assigning story owners, assigning a customer proxy from the development team, demoing to customer, and applying agile internally within teams only.

2.6.3 People Challenges

Management and team members are faced with a set of challenges when implementing agile. Managers are challenged to commit extensively to agile methods when implementing them which causes perturbation to the managers, team members, and organization as a whole [36]. Also, managers struggle to guide engineering and development personnel to integrate agile practices into their processes.

Self-organizing teams are closely involved in agile practices, such as estimation, planning, and development, which leads to the rise of a set of challenges [91]. Team members lack an effective level of communication which leads to lack

of knowledge sharing. This hinders the development process and reduces trust between teams and among team members. In addition, lack of skilled personnel may cause improper task estimation and allocation. Also, team members are challenged with the ability to achieve a cross-functional team especially with the dilemma of specialization and cross-functionality.

2.6.4 Inhibitors and Exciters of Agile Implementation

The strengths and weaknesses of agile software development are highlighted in this study using the Technology Institutions and Capabilities model (TIC). The TIC model was created by [23], shown in Figure 2.1. This lens identifies a relationship between institutional theory, the capability approach, and ICT. In the context of the model, ICT is the technology used to ‘deliver human centred development’, the capability approach looks into the freedom given to individuals to expand their abilities using ICT, and institutional theory focuses on structure, technology, and institutional driven change. The bidirectional arrows between the model’s elements represent the influences and interrelationships. The positive ones are labelled as exciters and the negatives ones are labelled as inhibitors.

The relation between each of the two elements represents one dimension in the model [23]. Dimension A links the institutions theory with the capabilities approach and examines the formal rules and informal norms. Dimension B connects the capabilities approach and ICT and studies the effect of access to ICT opportunities and practical skills, or their absence. Dimension C links the institutions theory to ICT and analysis the effects on transparency, networking, and communication.

This model presents a framework that combines institutional theory and the capability approach in relation to ICT and analyses the influences along the three different elements. This lens uses a case study approach, guided by [148], to explore the practitioner’s perception on the strengths and weaknesses of agile software development in the chosen Lebanese software development sites.

Inhibitors and exciters are then identified across each of the three dimensions. The visual representation of the positive reinforcement and impediments will lead to the identification of virtuous and vicious circles that enhance or diminish the role of agile methodology in the software development industry. Virtuous circles are identified across one dimension where one inhibitor, or exciter, will lead to another inhibitor, or exciter, thus forming a virtuous circle. Vicious circles are created when one inhibitor in dimension A leads to an inhibitor in dimension B which leads to an inhibitor in dimension C. The starting point may be any one of those dimensions.

In this study, the capabilities approach is the people's ability to fully understand and correctly implement agile. ICT is a software development method used with the agile ceremonies implemented and artefacts created. Institutional theory includes the roles of the employees and the structure of the organization. This analysis will explore how each of the model's elements when strengthened can decrease the inhibitors and increase the exciters in each case.

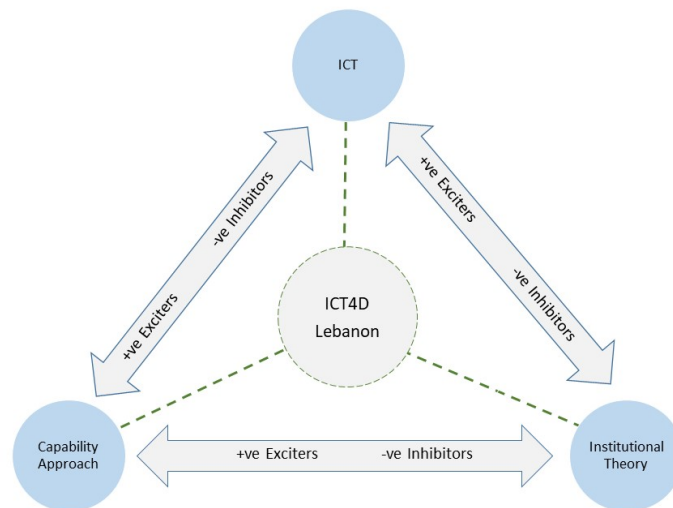


Figure 2.1: Technology Institutions and Capabilities model [23]

2.7 Agile Tailoring

In certain contexts, the adoption of mainstream agile methods is complex and requires a lot of effort [66]. After failure in applying agile methods in a number of projects, agile methods tailoring became a more common process [40]. Agile software development methods, such as scrum and Extreme Programming (XP), were introduced under the assumption that they could be used in developing any type of project [108]. However, literature has shown several failures in developing software when using agile methods [40]. As a result, the idea of agile tailoring started developing and agile methods were being adapted to fit a development context such as type of product developed or organization needs.

In the context of software process, software tailoring method is defined as “the adaptation of the method to the aspects, objectives, environment and reality of the organization adopting it” [40]. With the uniqueness of each software development process, tailoring agile methods is seen as practical process [51]. Thus, the implementation of a certain software development methods depends on technical, organizational, or project nature. Consequently, it is difficult to implement a software development method ‘as-is’ with no tailoring. Software organizations could further tailor agile methods to fit their company strategy and product type [115]. Thus, practitioners in the field and researchers have become particularly interested in studying agile methods tailoring.

Agile tailoring can be observed when organizations mix plan based methods with agile methods [?]. Plan based methods are characterised by a linear approach starting with gathering requirements, designing the solution, and then building the software. On the other hand, agile methodology is characterised by its dynamic aspects with short delivery cycles coping with changing requirements. For instance, one of the ways to tailor agile methods is to create both agile and plan-based artefacts to enhance the technological strategies and quality assurance methods of the organization [19]. Also, agile roles and responsibilities are tailored and new roles might emerge; for example, the role of the product owner may be tailored [18].

Certain researchers oppose agile tailoring. For instance, Ken Schwaber, the co-founder of scrum states: “Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.” Other researchers support the idea of agile tailoring in large-scale context [19, 17, 20, 59], regulated environments [76, 139], geographically distributed organizations [16, 22, 20], and through non-functional requirements [10, 39, 138]. Others support the mixing of agile methods with other traditional methods [163]. In this section, I concentrate on agile tailoring in relation to regulated environment and dealing with non-functional requirements. I do not consider agile tailoring in large-scale contexts, since the majority of Lebanese software development companies are SMEs (refer to Section 2.8.1).

2.7.1 Regulated environments and Agile Tailoring

Regulated environments are characterized by businesses where quality assurance, safety, security, or traceability are considered as key concepts [76]. One of the main characteristic in a regulated environment is the compliance of businesses with certain regulations, guidance, and official standards [76]. Initially, the implementation of agile software development in a regulated environment did not comply with the initial principles of the agile manifesto. Several recent studies, however, suggest that the tailoring of agile methods to meet the needs of a regulated environment may improve the implementation process [76, 41, 139].

The tailoring of agile methods came after the failure of several large-scale agile software development projects [75]. Thus, it became a possible solution to tailor agile methods before implementing them in large-scale software development projects [107]. When agile methods are tailored, they take into consideration the business objectives, environmental aspect, area of development and culture [40].

2.7.2 Non-functional Requirements and Agile Tailoring

Previous research has studied non-functional requirements in agile software development [29, 10, 39, 73, 138] and investigated different approaches to analyse non-functional requirements in the agile context. Addressing non-functional requirements in software development remains a crucial factor for the success of any development project.

Software development requirements are divided into two parts: functional requirements and non-functional requirements. The latter represents the behaviour of the system such as software performance in relation to usability, maintainability, security, and availability [10]. Non-functional requirements analysis and incorporation in the software developed is critical and essential for its success [10]. High risks are associated with neglecting non-functional requirements including scalability and security [136]. Customers are not fully aware of all non-functional requirements in the initial stages of the project, instead they focus on fully developing functional requirements leading to challenges in terms of software scalability, decrease in software quality, and increase in maintenance cost [29].

Simultaneously, the agile methodology does not provide a clear approach for handling non-functional requirements [12]. Agile allows space for minimal documentation which in turn causes traceability issues of non-functional requirements provided by the customer or regulatory agencies throughout the development. This leads to an increase in the amount of rework [10]. Researchers proposed non-functional requirement modelling frameworks, NORMAP, with a simulation tool, NORMATIC, for a semi-automated process [73, 72]. Also, NERV is another framework for elicitation, management, and validation of non functional requirements in agile. AFFLINE is a framework for integrating non-functional requirements in agile with a reference architecture for support [38]. All these frameworks and models base their studies on the automatic capturing or conceptual reinforcement of non-functional requirements from project documents [137], which is not always possible and their implementation has been limited.

There are several approaches used to analyse non-functional requirements [138, 98]. Research has proposed enhanced versions of these approaches [10]. Yet literature still lacks the identification and handling of non-functional requirements in agile software development context [72]. Agile teams identify factors that affect the non-functional requirements testing and acknowledge the need for further research in this field [39]. The gap in knowledge is even profounder on non-functional requirements in the context of cooperating software development teams. This topic is challenging due its cross-functional aspects and lack of clarity in this aspect in most part of projects. I aim to reduce the gap in the literature on the relation between agile tailoring, on one hand, and non-functional requirements on the other.

2.8 Software Development Industry in Developing Countries

Developing countries have been implementing agile methods in their software development process. Practitioners in countries are aware of the benefits of agile and realize the importance of agile implementation [142]. Nevertheless, the knowledge gaps between business owners, managers, practitioners and customers on agile software development hinder its implementation. Gaps between the developers and the customers also hinder the development process [32]. This is due to the customers IT skill level, communication pattern between customers and developers, and customer's lack of knowledge in the software development process [7].

Studies have investigated agile methods adoption in developing countries such as Kenya [20], Ethiopia [142] and Egypt [115]. In Ethiopia and Egypt, practitioners, with little experience in agile methods, chose to implement agile methods after light literature review on the subject and word-of-mouth knowledge transfer. The study in Egypt revealed positive effects on the software development process and customer satisfaction, while revealing challenges such as, high pressure on developers, inadequate use of effort estimation, and lack of sprint planning.

2.8.1 Software Development Industry in Lebanon

Lebanon is a country in Western Asia. It is strategically located at the intersection of Europe, Asia and Africa. Lebanon offers one of the most liberal economic climates in the Middle East. Lebanon's economy is service-oriented mainly driven by tourism and banking services. Following the world trend, Lebanon has recently witnessed an investment increase in high tech companies. According to the Global Competitiveness report in 2017-2018, Lebanon is ranked first in the technology readiness ranking and second in the innovation ranking in the Middle East and North Africa (MENA) region (excluding the Gulf countries). This is due to Lebanon's competitive advantages that include strong human capital base, cost competitive labour force, liberal market, and access to growing regional markets. The Lebanese information and communication technology sector is expected to reach 543.3 million by the end of year 2020 [96].

The software development industry in Lebanon emerged in the early 1970s with a few small to medium sized companies. Since hardware was expensive, the purchase was limited to the public sector, health institutions, and the banking sector. With the creation of personal computers in the mid-1980s, the sector witnessed a remarkable increase. Since the 1990's, the sector witnessed an enormous boost with the creation of the internet. Shortly after the civil war that devastated Lebanon, the government issued a series of laws to support small to medium organizations. The law on the protection of software copyright was not issued until 1999. Lebanon still lacks the necessary legal framework which heavily affects the industry. The delay in issuing these laws is due to political instability and economic priorities [45].

In year 2000, the Lebanese Central Bank launched a program called 'Kafalat' which supports small to medium sized innovations. This supported the young innovators who are interested in the software development industry. [168] lists the main associations, governmental incentives, and NGOs that provide support to the Lebanese software development sector. These are summarized in Table 2.2.

Organization Type	Number of Organizations	Example
Professional Associations	6	Lebanon SoftShore Cluster (2007)
Government Initiatives	3	SME Support Unit (2005)
Financing	9	FARO Fund (2011)
Business Incubators and Accelerators	5	Berytech (2001)
University Initiatives	2	LAU Institute of Family and Entrepreneurial Business (2004)
Programs and Competitions	2	The Ideaz Price (2012)
Other Entrepreneurial Support	12	Cloud 5 (2012)

Table 2.2: Main supporters to the Lebanese software development industry

In 2016, the ICT sector was estimated to be 436.2 million US dollars. The most recent report reveals that more than 200 companies are involved in the IT sector in Lebanon [96]. The industry is expanding from retail activities to innovation and content generation. Software development industry in Lebanon is mostly comprised of small and medium sized companies. According to a report published by the UNDP, 62% of the companies have less than 25 employees [165]. And only 8% have more than 100 employees. The geographical distribution of these companies shows that slightly less than half of the software development companies are located in Mount Lebanon. Figure 2.2 shows the geographical distribution of the companies across Lebanon.

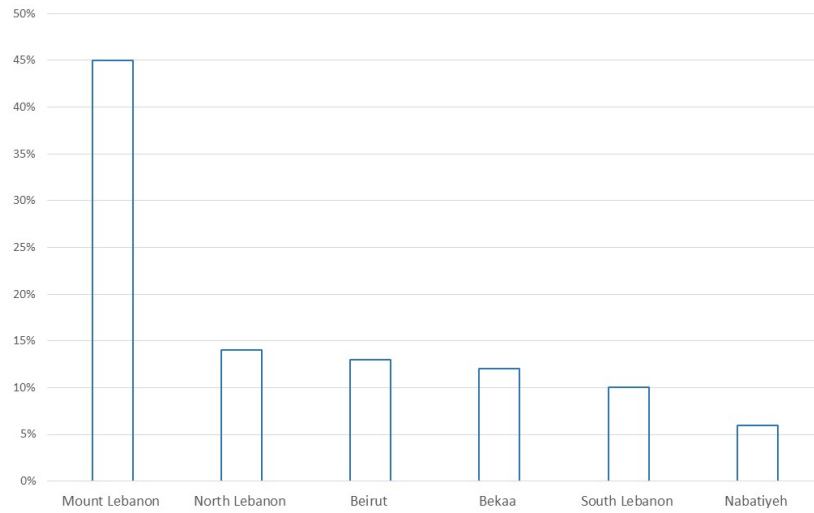


Figure 2.2: Geographical distribution of software development companies across Lebanon [165]

The demand for software development is rising. The primary demand is driven by traditional sectors such as banking, education, healthcare, and insurance [168]. Public institutions have also increased the demand for the development of e-government in the last decade [8]. Also, the foreign demand for the Lebanese software development is rising [85]. The most recent statistic in this regard, year 2013, shows that 25% of Lebanese companies are export focused with 75% of their revenues generated from exports and subcontracting of software development to foreign markets [168].

The Lebanese software development companies deliver high quality services in the areas of banking, education, health, communication, and trade. Lebanese companies were highly contributory in setting up software services in Syria, Saudi Arabia, and Qatar [74]. Most Lebanese companies are specialized in one of the following areas [14]:

- Specialized applications development
- Web solutions development
- E-services supply to the banking, healthcare, education, and retail sectors
- Multimedia products development
- Mobile applications development
- Game development for multiple platforms

Industries in Lebanon are threatened by the unstable political situation, turmoil in the region, and rising regional competition [168]. The current political situation is unstable due to internal and external unrest. Lebanon was devastated by a 15 year long civil war. The country is currently facing tensions between the government and the citizens. Lebanon is located in the Middle East that has been facing numerous conflicts from Syria, Iraq, Israel, and Yemen. In addition, Lebanon faces competition from neighbouring countries such as Jordan, Egypt, and UAE. Unfortunately, the economic crisis deepened after the August 4th explosion of 2,700 tons of ammonium nitrate in Beirut. Thus, rises the need to fortify the software development industry in Lebanon with tailored programs that increase the adaptability and flexibility of the companies.

The software development industry in Lebanon faces many challenges. Lebanese software development companies complain about not finding the skilled labour needed. Professionals in website development, programing, and software engineering are hard to find [165]. A mismatch between the academic sector and the labour market is detected. Zaraket et al. applied a skill-based representation over four analysed domains: university, companies, talents, and project [172]. This was evaluated in Lebanon on the academic and software industry. A list of software skills was created and prioritized according to the market's needs. Insufficient training hinders the improvement of the sector [168]. In addition, the legal framework needs updates and adjustments to cope with the continuously changing sector. The country's poor ICT infrastructure and lack of government plan for the IT industry

are barriers to growth. Also, the small size of the domestic market hinders the growth of software development companies [168].

Tarhini et al. present an innovative agile methodology for managing in-house software development [161]. This methodology aims at reducing time and cost in IT development projects for SMEs. The methodological design focuses on the roles defined in the Agile Manifesto, the agile process, and the artefacts initiating and resulting from the process. The study is validated by applying this methodology in a Lebanese university. A student registrar system was developed using the traditional software development methodologies and the agile methodologies. The result was a reduction of the development cost by 30%. The study also looks into the selection of the database technology. This selection depends on the complexity and interrelationship between the data used. The customization of existing methodologies enable SMEs to deliver the required software while decreasing the costs inquired.

There is a scarcity of research on the software development industry in Lebanon and its characteristics. The research is limited to reports from NGOs and private banks in addition to few articles that discuss the various subjects that fall under the software development industry. The scarcity lies in the benefits of implementing agile, its challenges, practices, and information flows in the Lebanese context. Thus, there is a large scope for conducting new studies. Studies show the benefits of implementing agile techniques. Therefore, the research will explore how these methods may be applied in the Lebanese context. This study aims at providing an understanding of the agile implementation in the Lebanese software development context.

2.9 Summary

To summarize, this chapter presents the literature available on agile software development. The presented work motivated this research and sets ground to study

agile implementation and create a tailored model in the Lebanese context. The scarcity of research in all aspects of agile implementation in Lebanon drives this study. The upcoming chapter describes the methodologies used to conduct this study.

Chapter 3

Research Design

3.1 Introduction

Researchers in the field of software engineering aim at expanding the research findings through appropriate research methods. This chapter presents the research methodology adopted for this study. It outlines the research strategy, data collection methods and data analysis procedure. It presents a detailed explanation of the applied research methodology used to achieve the aims and objective of this study.

This chapter starts by presenting the Research Onion [151] and this study's position in accordance to each element. This is followed by discussing the research purpose and outcome. The chapter then details the research approach which follows a mixed method qualitative methodology. Then, the five research sites of this study are presented followed by the data collection process which explains the processes of recruiting participants for interviews, transcribing data, and conducting observations. In addition, the data analysis procedure is presented through explaining the within-case analysis procedure based on the Constructivist Grounded Theory approach [42] and the cross-case analysis based on Cruzes's approach [56]. Finally, approach to create the model is presented along with an explanation of the

intervention in the chosen site.

3.2 Research Model

Research may be classified using the research model or Research Onion [151]. Figure 3.1 shows the research model stages which include the philosophy, logic, methodological choice, strategy, time horizons and techniques and procedures to complete the research studies. The below Sections 3.2.1 to 3.2.5 give an overview of the possible choices and reveal the choice for this study. Data collection and data analysis techniques are explained in Sections 3.4 to 3.7.

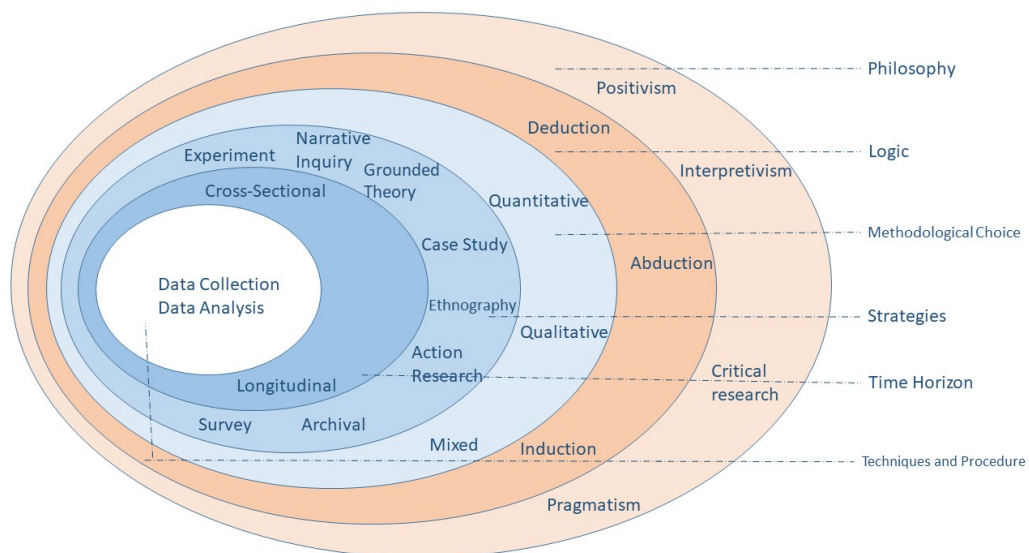


Figure 3.1: Revised Research Onion [151]

3.2.1 Research Philosophy

Research philosophy fortifies the knowledge on how the research is conducted. Research can be carried out using one of the four different philosophical perspectives, positivism, interpretivism (or constructivist), critical research (or participatory), and pragmatism [127, 68].

Positivism searches for evidence from the real world and deduces logical inferences [123]. Positivists search for ‘real observations’ based on measurements of variables, testing hypothesis, and concluding inferences from sample population [147]. Positivists argue that this type of study is independent from the researcher [116]. Positivists attempt to test a specific theory to understand the phenomena under study [116]. Post-positivism emerged after the notion of absolute truth was challenged. Thus, post-positivists take into account that evidence gathered is imperfect and flawed [127]. They attempt to test theories to refute them or increase the confidence in them. Post-positivism or positivism is most associated with controlled experiment, case study, and survey [68].

Interpretivism, also known as constructivism, relates scientific knowledge to human context. Evidence is gathered through qualitative data such as language, documents, artefacts, interviews, and focus groups [100]. Interpretists concentrate on analysing phenomena through meanings given to them by people rather than analysing theories. This social reality, defined by people, is subjective and holds multiple realities. Interpretists aim to understand these meanings and use them as building blocks in theorizing [81].

Critical research is intertwined with politics and political agendas [127]. Critical researchers recognize how cultural, political, and social domination may affect people’s ability to change their economic and social status [116]. Action research and case studies are methods used to conduct critical research. In the software development context, critical research challenges existing software development practices such as open source, agile methodology, and process improvement [68].

Pragmatism emphasizes the understanding of the research problem and uses all approaches and methods available. Pragmatists are concerned with practical problems and what is important ‘today’ [68]. Since pragmatism is not confined to

specific philosophical views, pragmatists believe that the researcher is free to use whatever research methods that serve their purpose [127]. They emphasize practical knowledge over abstract knowledge. Pragmatism may combine both, positivism and interpretivism.

This research follows a pragmatic research philosophy. The researcher's aim is to understand the implementation of the currently widely used agile methodology in the software development industry. The aim is to understand the research phenomenon based on the practitioners' perceptions on agile software development. In addition, the chosen research philosophy is influenced by the interpretive philosophy since grounded theory is used to generate theory rather than verifying existing ones.

3.2.2 Research Logic

Research logic refers to the direction by which the research is conducted [109]. It is identified by either inductive, deductive, or abductive approach.

Inductive Research or bottom-up approach moves from specific to general. The researcher aims to identify patterns, concepts, or theories from the collected data. Inductive research may be applied in both qualitative and quantitative methods.

Abductive Research begins with a 'surprising fact'. The research starts with a conclusion and then premises are considered [151]. Then, the premise that is most sufficient is chosen.

Deductive Research or top-down approach moves from general to specific. The researchers collect data as means to accept or refute a hypothesis. Deductive research is applied in quantitative method.

This research uses a bottom-up inductive approach. This research analyses Lebanese agile implementing software development companies and cross compares them with other studied companies to develop an agile model fit for the Lebanese context.

3.2.3 Methodological Choice

Three methodological choices or research designs are classified in the literature as qualitative, quantitative, and mixed methods [55].

Qualitative method is used to discover, explore, and understand the underlying motivation, reason, and opinion on a certain phenomena [55]. Qualitative methods delve into the participants' opinion, whether individuals or social groups, through collecting descriptive and conceptual data. Analysis is usually done inductively building from specific to general themes [126].

Quantitative method uses statistical and mathematical techniques to test theories by evaluating the relation among variables [55]. Quantitative data involves the use of classes, measurements, and numbers to perform hypothesis testing, correlation analysis, and predictive models [169]. Such methods assume a fixed research design and are associated with a deductive approach.

Mixed method integrated both methods, qualitative and quantitative. The core belief assumes that the combination of both methods produces higher understanding of the problem and fortified confidence in the research strategy[55].

This research uses a qualitative methodological choice. A qualitative methodology is conducted to allow the researcher to dive into the complexity of human behaviours in the agile software development industry. The study's aim is to gain an in-depth understanding of agile implementation in the Lebanese context from practitioners in the field and improve the productivity of software development. Data collection involves open-ended interviews and team observations at the chosen site.

3.2.4 Research Strategy

There are several research strategies considered for studies. Experiments, surveys, and archival research are most related to quantitative research. Whereas action research, case study research, ethnography, grounded theory, and narrative inquiry are associated with qualitative research [151].

Action Research uses a participate and elaborate approach to solve real life problems [103]. A successful action research has an effective outcome for the participating organization beyond the time limit of the research.

Case Study Research studies a certain phenomena or topic in the ‘real life’ [170]. A case study is defined by choosing the case and understanding its boundaries [78]. Case study research aims at generating insights, descriptions, and theories from in-depth study of real life contexts [70].

Ethnography is the earliest form of qualitative research dating back to the 1700s. It is the study of people and cultures. Researchers have expanded the use of ethnography to reach multiple disciplines such as software engineering [156].

Grounded Theory is used to generate theories by in-depth analysis of the participants’ social behaviour [80]. Researchers use the inductive approach to build theories which challenges the traditional theory testing approach. Grounded theory follows a set of techniques and procedure [80, 53].

Narrative Inquiry is a descriptive study concerned with the experience of participants. Such researchers value the participants’ narration of events and preserve the chronological order of their ‘story’ [43]. Beginning 1960s, narrative research has penetrated many disciplines and is now a methodology used in cross-disciplinary research [47].

This research uses a case study research analysis guided by a grounded theory approach. This research employs first a within-case analysis and then a cross-case analysis. The research approach is explained in section 3.4. The analysis is guided by a Grounded theory approach since it allows the researcher to address the complexity of human behaviour in software development context [154].

3.2.5 Time Horizons

The research’s time horizon depends on the availability of research sites and the resources available. The research may employ a cross sectional study or a longitudinal study [170].

Cross-sectional Study involves the study of a particular phenomena at a specific

time. Such studies are constrained by time.

Longitudinal Study studies change over a specific time period. It involves a research design that observes variables for a short or long period of time.

This research applies both time horizon techniques. First, five research sites are studied in a cross-sectional method, where open-ended interviews were conducted to analyse the agile implementation strategies. Then, a longitudinal approach is taken to study agile implementation strategies in a chosen Lebanese software development company.

Figure 3.2 summarizes the research classification of this study.

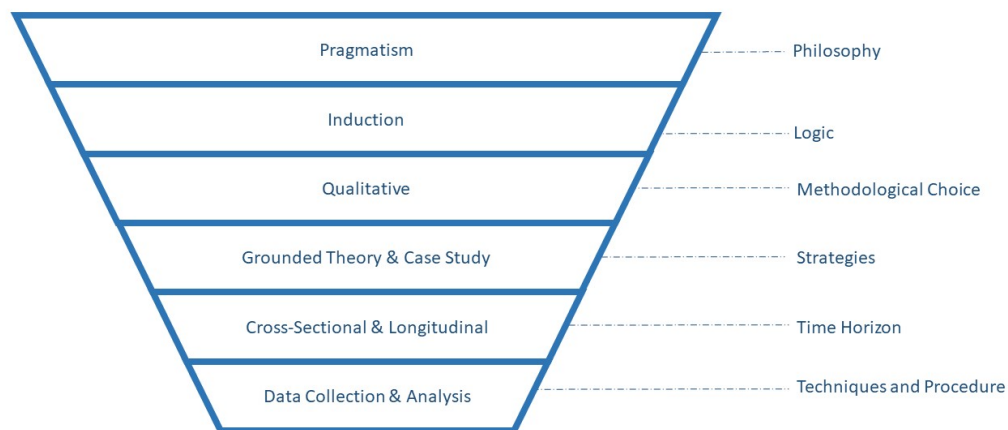


Figure 3.2: Research classification using the Onion Framework

3.3 Research Purpose and Research Outcome

The research purpose aims at defining the reason behind conducting the research [145]. Research purpose looks into the study context classified into four approaches: exploratory, descriptive, explanatory, and predictive [145].

Exploratory research aims at identifying an initial understanding of a hypothetical or theoretical idea. It involves extensive data collection leading to the understanding of the subject area. The researcher may create initial models, identify variable, and lay ground to future studies.

Descriptive research explores and explains phenomena as-is. Researchers aim at providing additional information about the topic through investigating and identifying particular characteristics.

Explanatory research is a higher order research, building on the exploratory and descriptive research, aiming to explain relationships between variables. New data is collected to deduct an in-depth analysis of the subject.

Predictive research looks into hypothesized and generalized relationships to develop a certain prediction.

The research outcome characterizes the findings into two categories, applied and basic research [145].

Applied research generates findings that aim at solving a specific real world problem. Improvement or change of certain practices is an integral part of such research.

Basic research, also referred to as pure research, aims at expanding knowledge and producing theories. Basic research produces new principles and ideas that form the basis for innovation and progress in studied fields.

This research is guided by an explanatory purpose of developing an agile software development model specifically for the Lebanese context. It follows an applied research strategy aiming to solve real world problems, specifically in improving software development productivity in the Lebanese context.

3.4 Research Approach

Mixed method qualitative research is adopted for this study. The research approach is aligned with the research objectives following the four phases identified in section 1.3.

Phase 1 revolves around data collection detailed in section 3.6. Semi-structured open ended interviews were the primary source of data. Observations were also conducted in the research site chosen for the intervention.

Phase 2 employs a Constructivist Grounded Theory approach to conduct a within-case analysis. The data analysis of the interviews for within-case analysis was done consecutively for each participating research site. The data set from the DevelopCo interviews were analysed first, then HealthCo, then LebDevCo1, LebDevCo2, and LebDevCo3. Details on the research sites are presented in Section 3.5.

Phase 3 is divided into two sections. **Phase 3a** employs a cross-case analysis where research sites are compared and contrasted. The reference point is the research sites in Lebanon. So the research sites in DevelopCo and HealthCo are compared and contrasted to the research sites in Lebanon. The cross-case analysis conducted is informed by Cruzes's approach [56] that is based on Miles and Huberman's approach. **Phase 3b** develops the agile model for Lebanon. A gap analysis is conducted between the actual agile methodology implemented and the designed agile. Conceptual mapping is developed to relate the influences that affect agile implementation identified in phase 2 and the agile methods used to overcome them. Then, an agile model is developed that uses input from the literature review, findings, and cross case analysis. This is the proposed designed model.

Phase 4 describes the intervention conducted at the selected research site detailed in section 3.10. This phase evaluates and updates the model by investigating its application at the selected research site in Lebanon and critically examines the issues that influence the adoption of the different agile concepts. An intervention was conducted at LebDevCo1 with two teams aiming at increasing productivity

through improving agile implementation. The process was incremental. The proposed model underwent several enhancements during the intervention before reaching the final designed version in the post-intervention phase.

3.5 Research Sites

This section describes the research sites which spread across 4 different countries: The Netherlands, Kenya, India, and Lebanon. All research sites are described in the below sections. For each site, I provide a description on the line of work, location and number of practitioners interviewed. The researches conducted at DevelopCo and HealthCo are used as preliminary studies to gain experience in qualitative data collection and analysis using grounded theory, in addition to the primary goal which is to advance the knowledge on agile implementation.

3.5.1 The Netherlands: DevelopCo

DevelopCo is an agile software development company that provides services for large software projects. It is co-located between The Netherlands and Kenya. The company uses agile to provide business solutions and create custom software. It develops business solutions for agile companies, non-agile companies, and governmental departments using agile software development techniques and conducts their business administration using enterprise agile. It has a main office in The Hague, The Netherlands and a partner office in Nairobi, Kenya. The company was founded in 2005 and has a diverse set of clients spread across Europe and Africa. It was chosen according to the snowball sampling technique; academic contact eased the connection. In the second phase, the professional contact provided access and organized a schedule to interview the participants.

A total of nine participants were interviewed from this company having different roles and responsibilities. Table 3.1 provides an overview on the participants' role,

responsibility, location, and years of experience in the company. Interviewees have been in the company for a range of 6 months to 3 years except for the director, who has been in the company since 2005. The data collected was obtained from semi-structured open-ended questions. The length of interviews ranged from 45 minutes to 1 hour and 15 minute interviews discussing the participants' experience.

Participant	Role and Responsibility	Location	Years in company
Director1-DevelopCo	Director of the Company	Kenya	4 years
Director2-DevelopCo	Director of the DevelopCo and Product Owner	Netherlands and Kenya	7 years
TechnicalLead-DevelopCo	Technical Lead and Product Owner	Netherlands and Kenya	4 years
PR-DevelopCo	Public Relations Manager	Netherlands	6 months
Sales-DevelopCo	Sales Coordinator	Netherlands	3 years
Designer-DevelopCo	Designer and Public Relations Team Member	Netherlands	2 years
Developer1-DevelopCo	Scrum Master and Developer	Kenya	2 years
Developer2-DevelopCo	Front End Developer	Netherlands	3 years
HR-DevelopCo	Human Resources Manager	Netherlands	6 months

Table 3.1: Overview on DevelopCo participants

3.5.2 India: HealthCo

HealthCo is a multinational company that provides IT services in the healthcare industry. The study conducted took place in Bangalore, India. HealthCo aims to improve patient healthcare by providing state-of-the-art medical technology. The products include medical imaging, laboratory diagnosis, healthcare IT, and electronics. The company has been innovating in the healthcare sector for more than 120 years. The employees who were interviewed for the data collection have been in the company for a range of 6 years to 18 years with an average of 10.5 years. It was chosen according to the snowball sampling technique; academic contact eased the connection.

A total of nine participants were interviewed from this company having different roles and responsibilities by Dr. Julian Bass who gave me permission to use the collected data. Table 3.2 provides an overview on the participants' role, responsibility, location, and years of experience in the company. The data collected was obtained from semi-structured open-ended questions. The length of interviews ranged from 45 minutes to 1 hour and 10 minute interviews discussing the participants' experience.

This data was collected by Dr. Julian Bass as part of a different study he was conducting. Dr. Julian gave me permission to use the collected data. Thus I listen to the recorded interviews and thoroughly read the transcribed data. The relevant information to this study were then coded using the same procedure for the other studies.

Participant	Role and Responsibility	Location	Years in company
Developer-HealthCo	Developer and model integrator	Bangalore	8 years
SeniorManager-HealthCo	Project manager for 5 sites and Agile coach for the Scrum team	Bangalore	16 years
SeniorDeveloper-HealthCo	Developer	Bangalore	6 years
ProgramManager-HealthCo	Sub-segment head for multiple projects	Bangalore	18 years
ProjectManager-HealthCo	Employee's management and resource's allocation	Bangalore	10 years
SeniorSoftEng-HealthCo	Developer and code reviewer	Bangalore	6 years
Architect1-HealthCo	Involvement in design, development and testing phase	Bangalore	11 years
Architect2-HealthCo	Involvement in design, development and testing phase	Bangalore	2 years
TeamManager-HealthCo	Team manager and analyst	Bangalore	11 years

Table 3.2: Overview on HealthCo participants

3.5.3 Lebanon: LebDevCo1, LebDevCo2, LebDevCo3

The thesis aims at providing new knowledge in the understanding of agile implementation in the software development industry in Lebanon and improve the software development productivity. The agile software development industry is spreading in Lebanon yet literature lacks studies in this area. Thus, we chose practitioners involved in agile software development form three different research sites labelled LebDevCo1, LebDevCo2, and LebDevCo3 to protect their anonymity.

In addition, an interview was done with the Head of Software Development Department in the Ministry of Economy and Trade. This interview aided in gaining an overview of the software development industry in Lebanon. The head of department also provided us with reports and statistics.

Lebanon: LebDevCo1

LebDevCo1 is a Lebanese based software development company providing solutions in areas of banking, analytics, technology, academy, insurance, retail, healthcare, and multimedia. LebDevCo1 has its main headquarters in Mount Lebanon, and two other branches in North Lebanon, and Bekaa Valley. LebDevCo1 has two major business lines. In the first, LebDevCo1 provides merchandising management and support software for global retailers and large distributors. This line is mainly provided for customers in the European market, France, Spain, Germany, the North American market, United States and Canada, and in the North African market, Tunisia. For example, one of the clients is a US based client worth 260 Billion USD. LebDevCo1 provides this client with an entire system to manage their numerous lines of products, stores, pricing strategy, promotions, and analytics. In the second line of business, LebDevCo1 provides web based solutions and services for banks and financial institutions. This line of business is available for the local market, i.e. the Lebanese market.

A total of seventeen interviews were conducted in LebDevCo1. Table 3.3 provides an overview on the participants' role, responsibility, and location. Employees interviewed have been in the company for a range of 5 years to 15 years. The data collected was obtained from semi-structured open-ended questions. The length of the interviews ranged from 45 to 55 minutes interview discussing the participants' experience.

The model validation will take place in this research site. This was possible after obtaining the CEO's approval to conduct workshops, focus groups, and post implementation interviews to see the practitioners' perception on the model.

Participant	Role and Responsibility	Location
OppMan-LebDevCo1	Operations Manager	Mount Lebanon
ClientEng-LebDevCo1	Client Engagement	Mount Lebanon
BuisDelHead-LebDevCo1	Head of Business Delivery	Mount Lebanon
DevelopDir-LebDevCo1	Development Director	Mount Lebanon
PO1-LebDevCo1	Product Owner - Project Manager	Mount Lebanon
PO2-LebDevCo1	Product Owner - Project Manager	Bekaa
ProjMan-LebDevCo1	Project Manager	Mount Lebanon
DataIntegration-LebDevCo1	Team Leader Data integration	Mount Lebanon
ProjMan2-LebDevCo1	Project Manager - Developer	Mount Lebanon
POBank-LebDevCo1	Product Owner of Banking Sector	North Lebanon
Dev1-LebDevCo1	Developer	Mount Lebanon
Dev2-LebDevCo1	Developer	Mount Lebanon
DevLeader-LebDevCo1	Team Leader - Developer	Mount Lebanon
BuisDelHead-LebDevCo1	Business Delivery	Bekaa
Dev3-LebDevCo1	Developer	Mount Lebanon
Dev4-LebDevCo1	Developer - Data modelling	Mount Lebanon
Dev5-LebDevCo1	Developer	Mount Lebanon

Table 3.3: Overview on LebDevCo1 participants

Lebanon: LebDevCo2

LebDevCo2 is a software development company that provides technological solutions for clients in Lebanon and abroad. The headquarters is in the capital Beirut. Agile is implemented in LebDevCo2 for all software development projects with the customers. LebDevCo2 develops on-demand and customized software. LebDevCo2's branch in Beirut is designed to serve customers in Europe and the Middle East. The need to be adaptable according to the customer's needs drove LebDevCo2 to become an agile implementing software development company. Developers and product owners felt the need to implement agile since their products are custom made and customer involvement is key in the project success. Below is a table that summarizes the participants from LebDevCo2, their roles, responsibilities, and respective location.

A total of seven interviews were conducted in LebDevCo2. Table 3.4 provides an overview on the participants' role, responsibility, and location. The data collected was obtained from semi-structured open-ended questions. The length of the interviews ranged from 40 minutes to 55 minutes interview discussing the participants' experience.

Participant	Role and Responsibility	Location
Dev1- LebDevCo2	Developer	Beirut
Dev2- LebDevCo2	Developer	Beirut
Dev3- LebDevCo2	Developer	Beirut
PO1- LebDevCo2	Product Owner	Beirut
Designer- LebDevCo2	Designer	Beirut
TeamLead- LebDevCo2	Team Leader	Beirut
PO2- LebDevCo2	Product Owner	Beirut

Table 3.4: Overview on LebDevCo2 participants

Lebanon: LebDevCo3

LebDevCo3 is a Lebanese based software development company that develops apps and websites for its clients. LebDevCo3 has its main headquarters in Mount Lebanon. LebDevCo3 has a major product where it develops platforms that relate teenagers, specifically school students, to the various universities. They aid young adults, high school through college, to document their achievements and activities in a key portfolio. The company has 30 employees, divided between full time and part time employees.

A total of seven interviews were conducted in LebDevCo3. Table 3.5 provides an overview on the participants' role, responsibility, and location. The data collected

was obtained from semi-structured open-ended questions. The length of the interviews ranged from 45 minutes to 1 hour and 20 minutes interview discussing the participants' experience on information-sharing in an agile environment.

Participant	Role and Responsibility	Location
CEO-LebDevCo3	CEO	Mount Lebanon
PO-LebDevCo3	Product Owner	Mount Lebanon
ContentCreat-LebDevCo3	Content Creator	Beirut
MarketingMan-LebDevCo3	Marketing Manager	Mount Lebanon
Dev1-LebDevCo3	Developer	Mount Lebanon
Dev2-LebDevCo3	Developer	Mount Lebanon
TeamLead-LebDevCo3	Team Leader	Mount Lebanon

Table 3.5: Overview on LebDevCo3 participants

3.6 Data collection (Phase 1)

This section describes the data collection process. It involves recruiting participants in each research site and observations conducted at LebDevCo1, model implementation site. Data collection in Grounded Theory is guided by a process called Theoretical Sampling. Under this process, emerging theories identify what data should be collected next [80].

3.6.1 Recruiting Participants

After gaining the ethics approval STR1819-30 (Appendix A.1), participants were recruited in each research site. The contact person would schedule the interviews. As a researcher, I would recommend that the practitioners are from different functional areas of software development.

The unit of analysis in this study are the practitioners. The study participants have different levels of experience ranging from 6 months to 16 years. The average level of experience for all participants is 4.3 years which leads to high quality of data and deep insight into agile implementation.

This research is based on 49 participants, of which 31 are from three Lebanese research sites. All participants were practising fundamental agile software development methods, such as sprint planning, user stories, frequent release of working software, and iterations. The data collected was obtained from semi-structured open-ended questions. The length of the interviews varied from one participant to the other depending on the participants' personality. The most effective way to optimize the data collected from interviews is to record and transcribe data manually [4]. All interviews were recorded after obtaining the practitioners' consent in order to ensure that no data is lost and that I concentrate on the conversation rather than jotting down notes. The practitioners' consent was either given orally through Skype or written through a signed document. Then, interviews were transcribed manually since it ensures correct transcription and reminds the interviewer of the social and emotional aspects that occurred during the interview [166].

The conducted interviews followed an open-ended questions interview guide that enabled participants to raise any issue that came up, even if it was not mentioned in the guide (Appendix A.2). The interview guide contained sections on interviewee information, agile implementation, agile tailoring, communication, agile roles, agile artefacts, agile ceremonies and challenges faced.

3.6.2 Transcribing Data

Data was transcribed verbatim. The transcripts capture what is said by both the interviewer and the interviewee. It also captures how it is said through indicating any feelings such as laughter and pauses. In the research sites DevelopCo and HealthCo, all interviewees talked in English all the time. In LebDevCo1, LebDevCo2, and LebDevCo3 interviewees expressed their opinion in English. Occasionally, interviewees used a couple of phrases in French or in Arabic. Note that this is due to the fact that Lebanon is a francophone country. These phrases were transcribed in the respective language and then translated to English. Below is an example of an interviewee who expressed their opinion in French language: Operations Manager expressed: “Les paroles sont volé les cris s’inscrit [word of mouth changes but written documents are permanent].”

3.6.3 Observations

In addition to the interviews, observations were made in LebDevCo1. LebDevCo1 was chosen to be observed since I gained approval from the CEO to implement the developed agile model. Observations were made on the agile practices, roles, artefacts, and ceremonies, conducted by the participants as well as inter-team communication. Notes were taken all throughout the observations. Thorough observations were recorded for two teams in LebDevCo1 before and after the agile model implementation. This provided greater insight into the collected data and fortified the validity. In addition, informal conversations were part of the data collected. They took the form of face-to-face conversations, phone calls, or text messages. To save these conversations, I either took notes during conversations, saved the written text messages, or recorded the conversations.

3.7 Data analysis: Within-Case Analysis based on Grounded Theory (Phase 2)

Data analysis for each research site was done individually. The following sections describe the Grounded Theory data analysis procedures that were conducted: transcribing the collected data, performing open coding, applying constant comparison methods, identifying core categories, and memoing the data, following the model in Figure 3.3 [93]. From the analysed data, interview concepts arise. These concepts are then grouped into categories which then form the main categories.

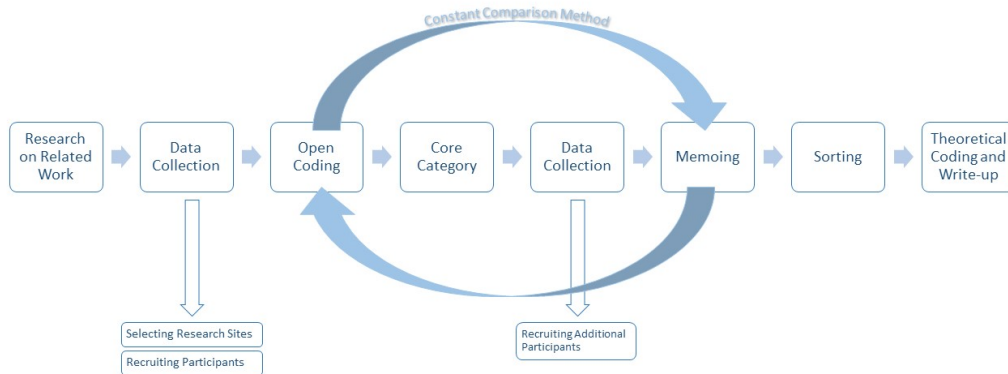


Figure 3.3: Grounded Theory components

Grounded Theory is chosen as the research methodology for various reasons. First, Grounded Theory has been used to study agile software development by numerous researchers [18, 93, 126]. Second, there is a compatibility between the topic, agile implementation in software development and the practitioner's interaction, and grounded theory which uses qualitative research methods to study social interactions. By using open-ended research questions asked in the interviews, the topics

will be generated by the participants. Third, Grounded Theory aids in answering our research questions that revolve around understanding the practitioner's perception on agile implementation rather than theory analysis or theory extension. It allows me to address the complexity of human behaviour in software development context [154].

3.7.1 Open Coding

All transcribed data was imported to an analysing tool Nvivo 11 (NVivo qualitative data analysis software, 2015). The University of Salford provides students with free license and training on Nvivo. Nvivo allows for efficient and quick data sorting and organizing. Open coding includes the analysis of data and exclusion of prior judgment to produce the maximum number of concepts and key points. All interviews were coded, thus deriving categories on a high levels of abstraction, along with patterns of behaviour [4]. Then, each key point is assigned a code title that summarizes it.

Line-by-line open coding approach was used on the transcribed interviews, which is more effective and useful than word-by-word coding approach. When coding line-by-line, data can be inspected and a special incident can be found in a word or line, or through several lines [4]. This coding process was applied through highlighting the selected sentences using the Nodes option in Nvivo 11. Using the constant comparison method, each interview was reviewed more than once to confirm that no data is left unnoticed. Figure 3.4 examines one example of comparison of quotes between and within categories of inter-team communication tools preferences.

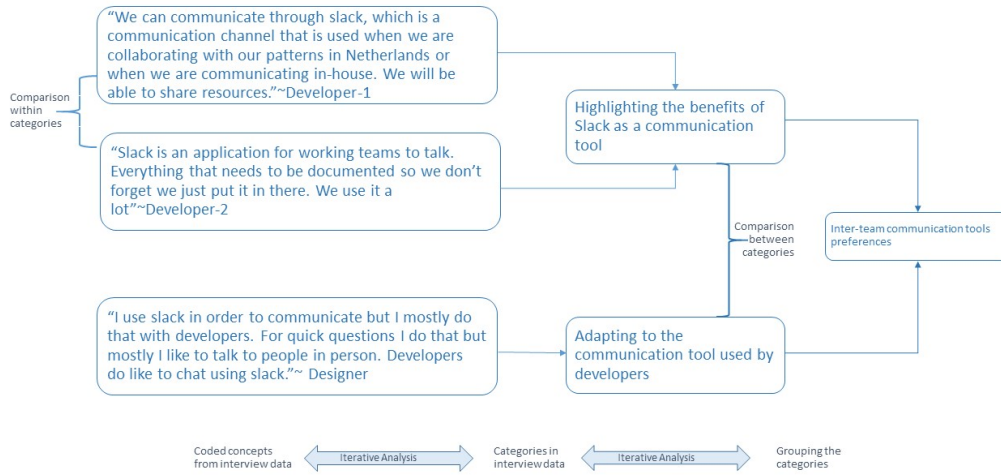


Figure 3.4: Iterative analysis

A core category shows the main concern of participants. After identifying the main problem, the researcher moves to selective coding. This process entitles the researcher to limit the coding to the data that corresponds to the core category chosen. This process reached a halt when no new categories were created, and theoretical saturation was reached.

3.7.2 Constant Comparison Method

Codes from each interview were compared from codes arising from the same and other interviews. This constant comparison technique enabled the grouping of codes that constitute concepts (a higher level of abstraction) which were then grouped, using additional constant comparison, into categories (an even higher level of abstraction) that were then coded. As more data was collected, previous data was revisited and compared with the newly collected and coded data.

Constant comparison was a key technique in this research. As a software engineering researcher, constant comparison enabled me to organize my analytical thinking skills. It also increased the validity of this research through guaranteeing that no data is left unnoticed. This is true since each data collected is visited numerous times and compared with previous data.

3.7.3 Memoing and Sorting

The next step was the ongoing process of writing memos under Grounded Theory process. Memoing is considered to be the ‘backbone’ of theory generation [80]. Memos were first written in their draft version, to guarantee that they are written in the “passion of the movement” [4], and then revisited and written in a formal manner, to insure clarity through using correct and revised English. Quotes from the interviews were used as evidence in the writing of memos along with the field notes taken during the interviews.

Next, the written memos were sorted on the conceptual level to relate the created categories to the core one. This was followed by relationship creation between memos through drawing out the relationships using a paper and a pen [95]. It took the use of several relationship diagrams to understand the data, relate the subjects, and build up the overall modelling of the relationships between the categories. These included swim-lane diagram, information flow, Technology Institutions and Capabilities model, and mapping of roles, artefacts and ceremonies.

3.7.4 Theoretical Coding

Consequently, theoretical codes were highlighted from the emerged concepts and categories. Relationships were analysed between the different categories and their respective properties to reach our theory. The writing-up of the findings came as a result from the sorting and theoretical coding. Our study revolves around agile implementation in software development companies.

3.8 Data Analysis: Cross-Case Analysis (Phase 3a)

In order to compare and contrast agile implementation in Lebanon on one side and in the other studied companies on the other, I perform cross-case analysis. Figure 3.5 demonstrates the cross-case comparison performed. The Lebanon case study was compared with both DevelopCo and HealthCo. A cross-case analysis enables researchers to compare and contrast the studied case. A cross-case analysis achieves high levels of confidence in regards to rigour [112].

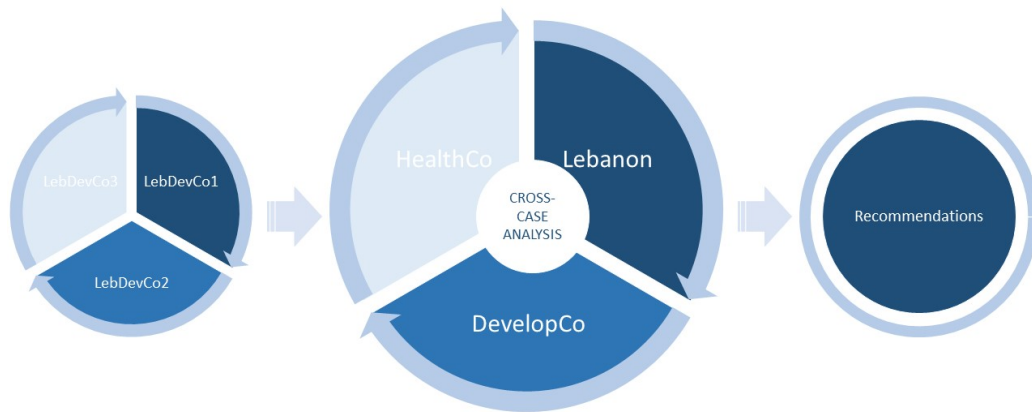


Figure 3.5: Cross-case illustration

Cross-case analysis method is used to compare the commonalities and differences in two or more case studies. “*Cross-case analysis is a research method that can mobilize knowledge from individual case studies*” [99]. It is a systematic method that uses diverse forms of qualitative data as evidence. Graphs and tabular displays are used to facilitate the data comparison analysis [99].

Variable-oriented and case-oriented are designated as the two main approaches to

cross-case comparison [99]. In **variable-oriented** approach, variables are the main points of comparison. The main aim is to generalize between the different studied variables. Researchers aim to identify themes and patterns among the studied cases [99]. The larger the number of studied variables the larger the number of case studies needed [57].

In **case-oriented** approach, seek to understand complex units [57] rather than establishing relationship between variables. Researchers concentrate on a small number of cases and focus on each case as an interpretable whole.

This study follows the case oriented approach as a guideline for our cross-case analysis. Porta (2008) identifies elements that guide the approach [57]. Table 3.6 introduces each element, provides a description, and links it to this study.

Element	Case-based Approach	Current Study
Cases as	Names with capitals (complex units)	Within-case analysis was performed before the cross-case comparison
Concepts	Constructed during the research	The qualitative research was guided by grounded theory allowing for new concepts to arise
Independence of cases	Addresses systematic process analysis	Applying a qualitative inductive research strategy guided by grounded theory
Number of cases	Keep N low	Number of cross compared cases is 3 (DevelopCo, HealthCo, and Lebanese companies)
Number of variables	Increase number of variables in order to make the description thicker (full accounts; case knowledge)	Studied parameters involve communication, roles, artefacts, ceremonies, challenges, and influences
Case selection	Tend to select paradigmatic cases	Research sites where selected according to specific criteria
Diversity as	Understanding through differences – exploring diversity	This research studies agile implementation in Lebanon through comparing and contrasting it with the studies research sites
Use of time	Processes and temporal sequences; eventful temporality	The model validation process is observed through the whole implementation period

Table 3.6: Elements identified in case-based comparison

3.8.1 Chosen Approach to Conducting Cross-Case Analysis

Researchers present several techniques and approaches to conduct a cross-case analysis [99]. Miles and Huberman's cross-case techniques involve the use of meta-matrices for segregating data and timely order display of data [112]. The process begins with a within-case analysis, then coding and tabulating the results of each case individually, then results are combined in a single table, and finally a cross analysis is conducted highlighting the similarities and differences. This technique involves highly disciplined procedure that are complicated and sometimes unnecessary [63]. Our approach is informed by Cruzes's approach that is based on Miles and Huberman's approach. Miles and Huberman applied their synthesis method within a multi-case study [63], Cruzes applied their synthesis across two single case studies [56], while we aim at conducting a cross-case comparison between the Lebanese case study on one side and the DevelopCo and HealthCo case studies on the other.

The method comprises of three iterative steps data reduction, data display, and conclusion drawing and verification [63]. Data reduction is the process of transforming the results from the studies to focused and simplified form. Data display involves the use of plots, tables, graphs, and matrices to organize the information and draw conclusion. Conclusion drawing does not have to be done at the end. Instead, researchers start building up conclusions from the start of data collection through highlighting explanations and propositions. The process of conclusion verification shifts towards the mid-end of the research process. Figure 3.6 illustrates the steps of the methodology used.

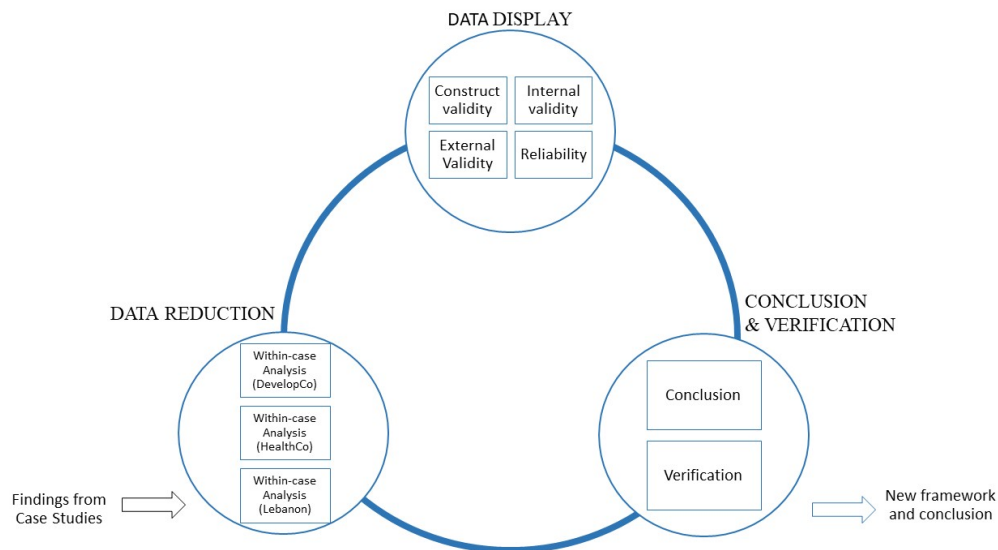


Figure 3.6: Cross-case analysis informed by Cruzes’s approach [56] that is based on Miles and Huberman’s approach [112]

Data Reduction

Findings from the case studies fed the data reduction stage that began with a within-case analysis of each. Data analysis involved gathering evidence from the studies through summarizing and coding broad themes. Memos were used to write the results from each case study. Then, a detailed case study write-up was developed for each case separately.

Data Display

The produced results fed into the second phase, data display. Here, results from each case study were compared and contrasted while understanding the differences and exploring the diversities. The data compared is displayed in a tabular form

to portray the commonalities and differences of communication and agile roles, artefacts, and ceremonies.

To understand the difference in communication forms, platforms, and inter-team communication across boundaries, I looked into the qualitative description of each study. Then, a comparison graph was drawn to highlight the commonalities and differences. To compare and contrast the agile roles, artefacts, and ceremonies, I used the Design-Reality Gap. This concept was developed from the literature on social construction of technology [160]. The design-reality gap provides a general framework for analysis of change. The smaller the gap the greater the chance of success [21, 86]. Since agile implementation is practically a newly applied concept in Lebanon, the use of this model deemed fit. I adjusted the model to fit the agile roles, artefacts, and ceremonies.

Conclusion Drawing and Verification

Then conclusions are drawn to further refine the above steps. Conclusion development start from the early stages of data reduction and data display. They develop throughout the process and sequence of the study, as per the case-based approach. The cross-case method allows the inference of new knowledge through analysing agile implementation in Lebanon. The verification of results was possible through applying the developed model in LebDevCo1 research site.

3.9 Model creation process (Phase 3b)

The model created is based on the concept of swim lane diagrams. Coming from an industrial engineering background, I realize the importance of determining the most effective ways to improve productivity and quality of method implementation using people and processes. Thus, swim lane diagrams are used as a mapping tool to understand the process flow in relation to the people performing it. In this study, the swim lane diagram is used to capture how agile is implemented in different

research sites. The different swim lanes are then analysed in Phase 3a as part of the cross-case analysis. Further, swim lanes are also used as part of the planned method to capture the improvement in agile implementation in the chosen research site. The swim lane for one research site is mapped over different time intervals, baseline state (current model developed in Phase 2), proposed state (pre-intervention model developed in Phase 3b), and final designed state (post intervention model developed in Phase 4), to show the effect of agile implementation on the process flow.

First, to understand the process flow in each research site, a **swim lane diagram** is developed. Swim lane diagrams are used as a modelling tool for process planning in software development [84, 97]. Every actor or team is assigned a swim lane (horizontal section). Then, all the activities are positioned in the corresponding lane. Arrows are used to indicate the sequence of activities which may be traced [30] showing the start of agile iteration and end.

Swim lane diagrams may be used as validation tools [124]. Such diagrams are proved to be effective and efficient in conveying information to researchers and stakeholders [97]. In this research, swim lane diagrams are used to determine the completeness and accuracy of agile implementation. Data display is performed through constructing an information flow model for each company. Information flows within each company were analysed and modelled using concepts in the findings. Input from other findings were also included using the constant comparison technique.

Swim lane diagrams are compared in two phases in the study, Phase 3a and Phase 4. In Phase 3a, swim lane diagrams allow us to visually distinguish the difference in the interaction and information flows across the different research sites, thus providing an input for the data display step in the cross-case analysis. The cross case analysis allows for the comparison of LebDevCo1's swim lane with not only the literature review of the workflow in agile but also to 'real life' cases, DevelopCo and HealthCo. In Phase 4, the swim lanes are developed for the same research site, LebDevCo1, across different time intervals, baseline state, proposed state, and final designed state. This swim lane diagram timeline allows the illustration of the

effect of introducing agile and tailored agile method to the flow.

Second, the design-reality gap lens is used to analyse the difference between the actual and designed agile implementation. The design-reality gap was developed from the social construction of technology and contingency in organizational change [21]. The design-reality gap model argues that organizational change holds assumptions. Thus, the design expectations may differ from the reality upon implementation creating a gap [87]. It may be used as a project evaluation tool [21]. Thus, the design and reality gaps are assessed longitudinally, post and pre intervention. In order to perform this evaluation, it is deemed necessary to identify the key dimensions. This study identifies 3 roles, 4 ceremonies, and 5 artefacts previously described in Section 2.1.

Third, the TIC lens is used to present the inhibitors and excitors affecting the agile implementation process. These were derived from the within-case analysis of the factors influencing agile implementation in Lebanon. The TIC lens is used as a driver to introduce the agile methods and tailoring inserted in the proposed model. It is also used as project evaluation tool. Thus, the TIC lens is assessed longitudinally, post and pre intervention. The difference in the inhibitors, excitors, vicious circles, and virtuous circles are registered at the two different stages and improvements are recorded.

Also, the model takes into consideration agile tailoring to fit the Global South context (Lebanon). The post-colonial theory lens has been used to examine the phenomenon of off-shoring in the digital economy [140]. Southern theory focuses on the possibilities of critical thinking taking place in, of, and about the Global South. In this view, those living in the Global South are all too often intellectually subalternised by the universal character of theoretical understandings originating in the Global North [50, 102]. This theory is used to suggest that the complex mixes of pre-colonial tradition, colonial imposition, and post-colonial adaptation, may produce unique customisations of otherwise Globally Northern impositions. This process may in fact be requisite both for successful adoption and for effective integration of new technologies and practices in various parts of the Global South.

3.10 Agile Tailoring to the Lebanese SME Context (Phase 4)

The intervention period is used to verify and validate the data analysis. I was able to get permission to implement the model at LebDevCo1 after signing a confidentiality agreement with the company. The CEO gave me permission to guide the intervention process, conduct recorded observations, and perform post-implementation interviews. First, the results from the data collection and analysis process were introduced to the CEO and top managers at LebDevCo1. Then, series of workshops and presentations on the agile methods were presented to two teams who will be involved in the process along with the involved stakeholders. Team A includes 6 members and the Product Owner (PO) and Team B includes 7 members and the PO. Three members from Team A, four members from Team B, and the Product Owner were from the practitioners interviewed in Phase 1. The workshops extended all through February 2020.

Initially, LebDevCo1 gave me 3 iterations to validate the data. After the end of the second iteration, all Lebanon was on lock down due to COVID-19. Thus due to force majeure, on March 20, I was forced to pause the validation process which was resumed back in June. According to plan, the implementation process should proceed for another iteration. At the end of this iteration and due to impact observed by the management, I was able to negotiate yet another three iterations. As a result, the implementation period extended to a period of 16 weeks.

Observations were conducted all throughout the implementation period with recordings being used to extensively analyse the process. Due to the confidentiality agreement, all recordings were to be deleted after analysis. Post-implementation interviews were conducted with the PO and all 13 members from both teams. All interviews were recorded, transcribed, coded and memoed. Impact evaluation methods were used to provide evidence based tools for the impact created [141].

The input for the intervention, Phase 4, is the proposed model with suggestions on

agile methods to be implemented. The intervention process allowed for extensive observations, direct relation with practitioners, and access to artefacts. This initiated new ideas to tailor agile method to fit the context. These ideas were proposed to practitioners and then tested. With new tailored agile methods implemented, changes were added to the model. The proposed model underwent numerous improvements to reach the final designed model. To validate the final designed model, data was collected during the intervention phase by taking notes during observations and recording stand ups. Also, data was collected after the completion of this phase by interviewing practitioners who took part in this intervention. These interviews were then compared with the initial data collected and the improvements were recorded.

3.11 Summary

This chapter summarizes the methodological issues of this study. All the aspects of the methodological issues are portrayed in figure 3.7.

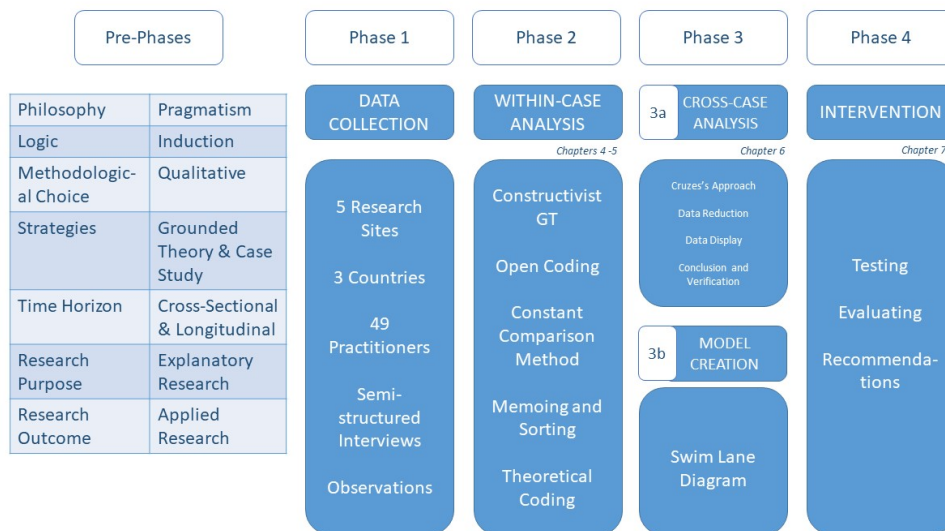


Figure 3.7: Summary of methodological issues in this study

Chapter 4

Approaches to Agile Implementation

4.1 Introduction

This chapter is divided into three main sections that present the findings from the collected data at DevelopCo, HealthCo, and chosen Lebanese research sites respectively. The study at the first two research sites is characterized as a preliminary study used to gain experience in conducting qualitative analysis and using Grounded Theory. Also, it is used to gain real life experience on agile implementation in software development companies. The findings from DevelopCo are shown first followed by the findings of HealthCo.

This chapter also presents the analysis of the data collected in Lebanon. It goes deeper into details than the first two preliminary studies. It presents the factors that influence agile implementation and tailoring in the chosen Lebanese research sites. This chapter represents phase 2.

4.2 Agile Implementation in DevelopCo

DevelopCo's section presents the swim lane diagram, portraying the main activities in each section while highlighting the network of information flows and communication channels, and indicates the difference in agile implementation across the company's two geographical locations. Then, the communication bottlenecks between teams is shown. Also, the collaboration difficulties between the customer and the teams are described. This section also looks into the most recurrent inter-team communication tools used and the differences in practitioners' perception and advocacy of these tools. Finally, the approach taken towards non-functional requirements is described.

4.2.1 DevelopCo Process Flow

Figure 4.1 presents four swim lanes displaying the main actors in the agile software development. The model starts with the customer's input on product requirements and ends with the retrospective. Activities that are placed on the borders of the swim lanes indicate that the activity is performed by the actors sharing the border. As part of the agile process, the diagram shows how the customer is involved in the product development process. The customer is included in the model since the evaluation and completion of the product backlog is based on their input along with the success of the sprint. This is based on the customer's evaluation of the team's demo. Also, it portrays how common the inter-team communication and the product owner involvement are in the process. Additionally, it illustrates the scrum master's role and highlights how the scrum master acts as a link between the different actors.

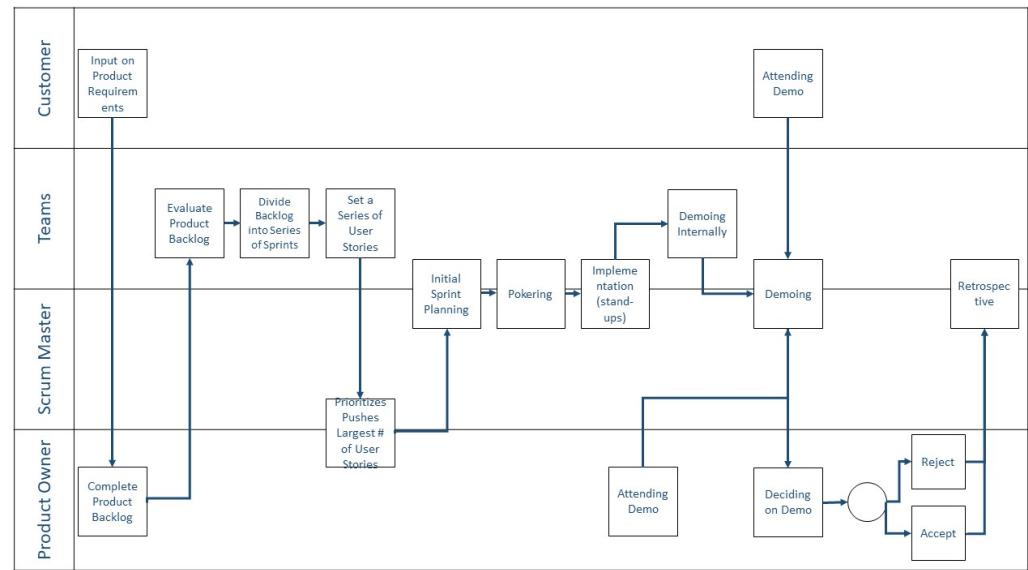


Figure 4.1: Swim lane diagram portraying the process flow of DevelopCo

4.2.2 Geographically Separated Inter-Team Collaboration in relation to Agile Implementation

The inter-team communication in DevelopCo is spread across two geographical locations, The Netherlands and Kenya. The agile implementation process started in The Netherlands first. Then, practitioners from The Netherlands informed practitioners in Kenya on agile implementation. This section observes the difference in agile implementation between the practitioners in the two countries. Director2-DevelopCo expressed their view on agile implementation: “*Agile may be applicable to every culture, but within each culture it is not applicable to all people.*”

Struggles to implement agile in The Netherlands

Employees with minor or no agile implementation experience find it difficult and time consuming to grasp the agile way of thinking. PR-DevelopCo finds it demanding to communicate and use agile concepts: *“It is hard on me to adapt to the words used, thus it is hard for me to communicate.”* HR-DevelopCo pinpointed that it is harder to implement agile in operational work as opposed to software development: *“There is always going to be a discrepancy, because I am not building a software. It is different when you are applying agile to an administrative process.”*

Employees who have been implementing agile for a longer period highlighted several times the benefits of agile such as self-organizing teams and the clarity of user stories. On the other hand, other employees found it hard to implement agile especially if they had previous work experience in a different field. The ability and will of every employee to forget any preconceived and practiced ideas about project management and start implementing agile differs. TechnicalLead-DevelopCo highlighted: *“One learns how to do certain projects in a certain ways, and if one wants to implement agile they need to unlearn things.”* In the Netherlands, employees have to unlearn preconceived ideas on project management and software development that are non-agile. One specific example was given by Director2-DevelopCo on one of the employees who came from a government sector which is characterized by: *“Long term planning, bureaucracy, and perfecting both the value and non-value-adding activities.”*

Struggles to implement agile in Kenya

At the initial agile implementation stage in Kenya, the employees at The Netherlands instructed their Kenyan partners on agile implementation. The Kenyan partners were able to learn the agile process with a fresh mind-set. The process was a mutual benefit, where the employees at The Netherlands learned and enhanced the agile process back home while directing the Kenyans. The TechnicalLead-

DevelopCo said: *“After every sprint in Kenya we learned new things and restarted the game.”* This mutual learning process affected the internal communication in The Dutch office with every practice of long-distance communication with the Kenyans. In addition, the office in Netherlands updated the way Kanban boards were designed following a method created by the Kenyan partners. The TechnicalLead-DevelopCo said: *“In Kenya, they came up with some ways to organize their Trello board [digital Kanban board] for their office team and management team. We adapted and then changed that for a more suitable way for our office.”*

Also, change was portrayed in the improvement of user story-writing. The Kenyans required detailed explanation in order to understand the work that needed to be done, and user stories were hence written with extensive description and detail. This became a habit in The Netherlands too, since it clarified more the user stories and decreased the number of questions asked later. The TechnicalLead-DevelopCo explained: *“Because we added extra description in Kenya, we found out that we need to give a little more information that they can look up afterwards instead of telling that in the sprint planning next to the user story.”*

4.2.3 Communication Between Practitioners

Communication at inter-team boundaries faced several challenges indicated by practitioners. These include lack of inter-team communication, lack of collaboration among team members, and absence of inter-team communication in the presence of dependencies.

The first challenge is the lack of continuous update among teams. Teams that are facing setbacks assume that other teams are aware of their situation thus do not feel compelled to explain their status. Developer2-DevelopCo said: *“Teams are like: we know so the rest knows it as well. They are just assuming.”* The TechnicalLead-DevelopCo affirms the above by stating: *“They [team members] assume that if they mention that something did not go well then you know exactly what they are talking about even if they do not mention details.”* When such cases

are repeated, trust between teams starts to decline. This causes significant problems in the performance of teams leading to poor performance in the overall project. Developer2-DevelopCo describes an incident where team B was dependent on team A's successful sprint completion. Unfortunately, team A failed to complete their sprint and did not communicate this to team B, who in turn could not initiate their own sprint. Developer1-DevelopCo said: *"If the scrum master had told the other team, they could have done a new sprint planning. Now they were just waiting and two teams were set back by that."*

Secondly, with tight deadlines in the workplace, team members focus on completing their own user stories and tend to neglect requests from members of other teams thus weakening the collaboration between teams. Sales-DevelopCo describes an incident that occurred with one of the developers under that context:

"She started working at the customer but the screening was not completed yet and she needed to hand a document. I emailed her, called and she was not responding for several days and that was really frustrating. It took three days to complete that."

Thirdly, the interconnection of sprints cause high level of dependencies. Reliability between sprints may hinder the process, especially if the work done was incomplete or needed rework. An unsatisfactory sprint may contain or block the work flow. Developer2-DevelopCo explained a similar case:

"In the last sprint there was a team who picked up a bit too much, they underestimated their user stories and another team was dependent on what they were supposed to make, but they were not informed of the delay. So they were waiting for their sprint to start for some time. So if they had told them [about the unfinished work] they could have done a new sprint planning, plan different stuff and finish other tasks."

4.2.4 Team and Customer Collaboration

Agile software development requires the customer's input to the project; it is customer driven. Thus, understanding the customer needs and building on their requests are of high importance. Here rises the need to analyse the challenges at the boundary between the teams and the customer. Such issues include communication with customers through agencies, dealing with unclear customer requirements, reaction to customer feedback, and collaborating with non-agile companies.

Communicating with Customers through Agencies

Agile software development integrates customer feedback through its practices. In certain cases, a third party acts as a communicator between the company and the customer; consequently, communication is faced with interruptions. TechnicalLead-DevelopCo said: *"When the development is through an agency, the communication is more difficult."* Knowledge sharing becomes difficult and unprecise, leading to frustration of team members. Restrictions and limitations that accompany the involvement of an agency hamper the team members' performance and enthusiasm.

Unclear Customer Requirement

Customer involvement and input happens all through the agile process. Lack of clarity in customer requirements leads to unclear product backlogs, causing time delays in delivery. PR-DevelopCo explained: *"The lack of information in the project we are doing is a factor that will negatively affect delivery time."* The team members' experience in formulating and asking the accurate questions at the beginning of the project prior to forming the product backlog is vital. Director1-DevelopCo highlighted the importance of the scrum master's role in clarifying the customer's requirements: *"The scrum master can clarify the story or feature with the customer and product owner and communicate that to the team."* Sometimes the team members or the scrum master base their product backlog solemnly on

customer's briefing and fail to investigate further. Director2-DevelopCo said: *"I think the most common negative effect on the workload is the lack of clarity and understanding of what is required."*

Reaction to Customer's Feedback

The success of the team's demo to the customer and the retrospective are related to the customer's approval of the sprint. Some employees understand that continuous customer involvement is core in agile software development, and customers tend to change or refine their requirements frequently. On the other hand, some team members process the customer's feedback during the demo personally and turn the retrospective into a session to discuss solemnly the demo instead of the sprint as a whole. Director2-DevelopCo indicated: *"In our last demo we had problems with the customer that resulted in negative feedback, and then in the retro everybody put negative improvement stickies related to that particular incident."* Some of the employees become directly affected by the customer feedback presented in the demo, which stretches further in affecting not only the retrospective but also the performance in the coming days. Developer-2 explained how negative feedback is dispiriting yet is essential to reach the customer's requirements and how team members may require a recovery period:

"It is usually a bit demoralizing to start over again. But you have to start over and make sure that this is what they want next time...Usually the demo is on a Thursday and Friday is usually a personal sprint day, while Monday you can just start over. Friday would be like a day to recover."

Collaborating with Non-Agile Implementing Companies

Not all customers understand agile and are willing to collaborate with the company on an agile basis. In such case, being fully committed to implement agile, DevelopCo works on incorporating the non-agile requirements into the agile pro-

cess through user stories. Director1-DevelopCo explains: *“We just take an agile approach when dealing with non-agile.”* Director 2-DevelopCo describes a project of a highly regulated nature that requires setting milestones and documentation to develop a business strategy for a government department. Such customers may hinder the completion of the project, since giving regular feedback is not implemented in their system.

4.2.5 Inter-team Communication Tools

In this section an overall picture of inter-team communication is presented. Then, the chosen tools, instant messaging and virtual Kanban boards, are discussed to highlight the difference in practitioner’s perception in regards to these tools.

Face-to-face communication is used during agile ceremonies such as daily stand-ups, demos and retrospectives, as well as regularly during the day. One of the main advantages of face-to-face communication is the live feedback through body language, facial expressions, tone, reactions, and feelings. Director2-DevelopCo said: *“When most people use a document we try to use an email. When most people use email we try to use chat message. When most people chat we try to talk to each other.”* Contrary to developers, the PR, Sales, HR, and designers consider face-to-face to be their preferred form of communication and tend to use it frequently. Designer-DevelopCo indicated that face-to-face communication is the initiation of any project design: *“Without face-to-face communication I cannot start with my design.”* Due to geographically separated teams, video conferences become a regular mean for communication. Challenges rise especially with weak network signal, thus making remote communication between Kenya and The Netherlands inexpedient. Designer-DevelopCo indicated that *“Sometimes in Kenya they are a bit slower and they have trouble with internet connection a lot. Sometimes the sprint would be affected by that.”*

Overall Picture of Inter-Team Communication

Different communication forms and tools were revealed by practitioners during interviews including instant messaging tools, face-to-face, communication, emails, video conferencing tools, and Kanban boards. Appendix A.3 includes practitioners' responses on the communication tools used. For each practitioner, a quote describing the communication tools is extracted and the respective communication tools are indicated. From Table A.1 in Appendix A.3, communication and coordination enhancement between teams is mostly achieved through instant messaging tools (or social networking platforms) and virtual Kanban boards. The coming sections discuss the above mentioned tools since they are the ones recurrently mentioned by the practitioners.

Instant Messaging Tools (Social networking platform)

In this company, Slack is used as a social networking platform for employees to communicate together (Slack, version 3.3.1). This application serves as a digital workspace for daily communication and knowledge facilitation. Developers were characterized as advocates for regular Slack usage, while designer, sales, public relations, and human resources teams were reluctant towards Slack usage.

The developers advocated the daily usage of Slack as means to communicate with members of the same teams, members of other teams, and teams located in different geographical locations. Developers highlighted the numerous benefits of Slack. Slack allows open communication between members of the same team, across teams, and within the company as a whole.

In addition, the TechnicalLead-DevelopCo highlighted one of Slack's benefits as a platform to request and receive support when needed. Slack also has tagging and notification options that can be used to notify specific team members or a whole team. Developer1-DevelopCo noted that *"If we have a group that is specific for a certain project, I am able to tag the members of the team to draw their attention to*

something... everyone will get notified as soon as I post something.” Contrary to developers, public relations, human resources, sales and designers were reluctant towards the usage of Slack. Non-developers tend to use Slack solemnly as means to communicate with the developers. Designer-DevelopCo said: *“I use Slack in order to communicate but I mostly do that with developers.”*

Virtual Kanban Board

Virtual Kanban boards are tools designed to visualize the current work in progress of all teams in agile software development. Employees recognized the importance of the virtual Kanban board, Trello (Trello, version 2.8.3). Trello boards aid in managing the sprint process through showing the sprint backlog list, identifying the work in progress, indicating the accomplished tasks and limiting work overload. TechnicalLead-DevelopCo indicated how Trello boards reduce the number of regular meetings through transparently updating the progress of each team member: *“Trello keeps an overview of all the project, sprints, and teams at the same time... With scrum boards in the rooms, one can see what the status is without too much hassle.”*

Moreover, Trello boards allow team members to check the work status of different teams by simply examining the board of the respective team. Trello boards highlight the status of user stories. TechnicalLead-DevelopCo explained: *“We use Trello and this is where we set all the user stories for the teams...when you pass by you can see straight away what the status is of the sprint.”* Designer-DevelopCo stated how convenient and practical Kanban boards are: *“The Trello board is more convenient to keep each other up to date and also as a reminder of how much we need to do. It is more like a practical thing.”*

4.2.6 Approach to Non-Functional Requirements

DevelopCo works on maintaining an agile software development process. As Director2-DevelopCo puts it: *“There is nothing I can think about in the company that is not agile.”* Director2-DevelopCo describes how being agile requires continuous learning and improvement:

“One of the things I learned about being agile is that you cannot be 100% agile; as time passes by, we realize always how we can get better. We think we are 100% agile and then 2 weeks after, we realize we can make it even better.”

DevelopCo has worked in regulated environments such as a project designated to develop a business strategy for a government department. The project was in a highly regulated environment requiring documentation and milestones. Director2-DevelopCo describes the project: *“We have a project with the German Government, which is completely non agile. The project description is non agile, the documentation is non agile, and the terminology is non agile.”* In such cases, DevelopCo treats all non-functional requirements into user stories incorporating them into the agile process. Director2-DevelopCo explains: *“We just take an agile approach when dealing with non-functional requirements. We treat them as stories.”*

DevelopCo has even abolished the manager role and only has scrum masters and product owners. In addition, when dealing with non-agile implementing companies, the company takes an agile approach through creating user stories to cope with the circumstances. Director1-DevelopCo clarified: *“If your customer is non agile, then the feature of the product or project you are building just becomes another agile task.”*

4.3 Agile Implementation in HealthCo

This section presents the findings from HealthCo. First, the swim lane diagram is presented, portraying the main activities in each section and highlighting the network of information flows and communication channels. The employee development and training performed by the company is described. Then, the communication bottlenecks between teams and the difficulties between the customer and the teams are presented. Also, a general overview on the inter-team communication tools used are explored. Finally, the approach taken towards non-functional requirements is presented.

4.3.1 HealthCo Process FLOW

Figure 4.2 presents five swim lanes displaying the main actors in the agile software development in HealthCo. The diagram shows the high influence of the Healthcare Regulatory Agencies on the product development and how they can insert changes to product specifications due to updates in universal medical regulations. As part of the agile process, the diagram shows how the customer is involved in the product development process. Initially, the evaluation and completion of the product backlog is based on the customer's input; lastly, the success of the sprint is highly based on the customer's evaluation of the team's demo. Furthermore, it displays the scrum master's role and highlights how the scrum master acts as a link between the different actors. It also portrays how common is inter-team communication. The product owner involvement in the process is steady especially with the weekly backlog grooming sessions that affect the two implementation phases in each sprint.

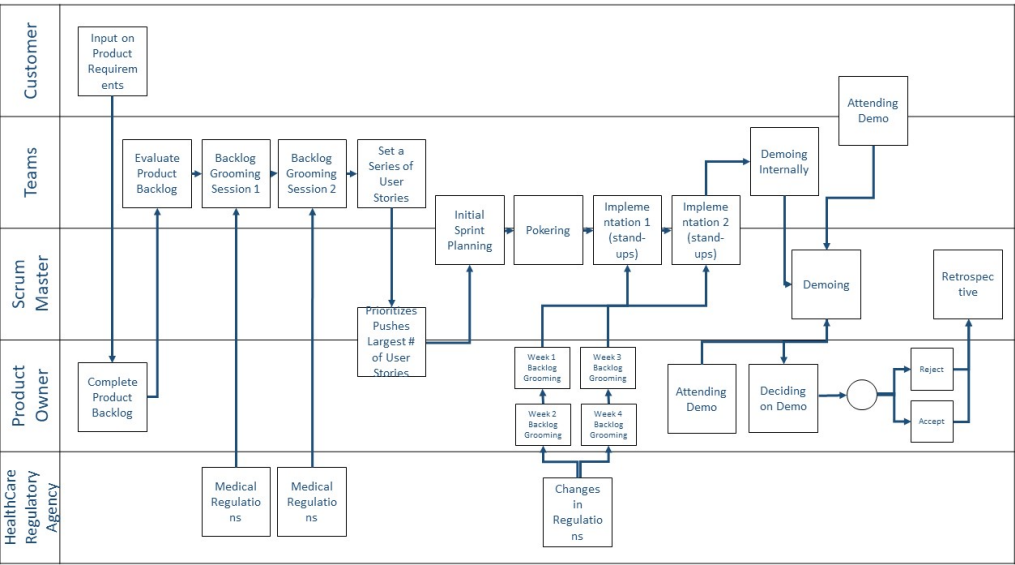


Figure 4.2: Swim lane diagram portraying the process flow for HealthCo

4.3.2 Employee Training and Development

HealthCo puts high efforts on developing the soft skills of its employees. The company concentrates on each person individually. TeamManager-HealthCo said: “The base value system is completely different for every individual in the team thus we do take care of individuals.” He/She explains further the case of Indian employees and why the company believes in individuals: “It’s not a matter of money in India, at least in my opinion. People really compromise and give value for their career.”

Training is also conducted for computing and technical skills when needed. For instance, Develop-HealthCo described a case where they needed training and he asked the scrum master for that: “We need to develop something in Java and I don’t know Java, then I discuss it with the Scrum master and he organizes the training sessions.” The developer also indicated how much the senior management

encourages the advancement of employees. Architect1-HealthCo described a similar incident: *“When a senior developer is performing well and is capable of handling certain things well, then there’s a good chance that he gets into the architect role.”*

4.3.3 Communication Between Practitioners

At HealthCo, communication is encouraged across teams. The agile coach, who is also the Senior Manager, highlighted the importance of communication regardless of the geographical location: *“We encourage people to talk to each other as much as possible regardless of which site they are.”* Communication is organized through different group meetings that include, but are not limited to, sprint meetings, weekly reviews, retrospectives, a common framework called ShareGo, and face-to-face communication. In addition, even though it is not as recommended as a platform for communication in the organization, some employees use WhatsApp in order to connect with other members in the team for any logistic reason or personal reason. TeamManager-HealthCo indicated: *“We have a WhatsApp group for our team to stay connected. If somebody is not coming, they may just text what’s the status and everybody will get it.”*

On another note, when transferring information from one actor to another, some of this information is lost or altered unintentionally. This impacts team performance and work-in-progress. This is a case that ProjectManager-HealthCo faced with newly recruited employees: *“I believe when you transfer, 50% of information is lost and it will become a very challenging system if you have got all new people.”* High interaction between team members is needed in order to meet deadlines. Teams with lesser loads than other tend to take the tasks of other teams in order to facilitate the process. The SeniorSoftEng-HealthCo described an incidence where teams helped each other through transferring the workload from one team to another.

Inter-team Dependencies

When working on large-scale projects inter-team dependencies become a natural process. In the healthcare sector, some dependencies occur in the same geographical location, for instance different teams on the same floor, and other dependencies occur between teams located overseas. Architect2-HealthCo located in India explained a situation where the algorithm team was located in the United States of America. Moreover, he gave another example of dependencies between different teams that are geographically separated: *“Our guy [concept coach] who is sitting in [redacted German city name] is interfacing with the clinical coach and the team here [India] for all the processes part.”*

Furthermore, the employees reported different types of dependencies between teams such as technical or workflow dependencies. For instance, Architect1-HealthCo described a dependency between the architect team and the platform team: *“Now, we are reaching the feature phase. Are we going to deliver this new tool within the feature phase or not? So here we have a dependency to the platform team.”* Moreover, there are cases where the project has to be delivered in a shorter amount of time than previously planned. This will create pressure from the upper management on the employees causing some difficulties.

4.3.4 Team and Customer Collaboration

Customer involvement in the agile process is vital. When implementing agile, customers become major influencers in the software development process. The agile mind set puts the customers' needs first. According to the ProgramManager-HealthCo: *“A team that does not implement agile would say: I have implemented everything, and it is working, why you want to change it.”* On the other hand, an agile implementing team would say: *“It is not a question about whether it is working or not, [the question is] whether the end users are liking it or not.”*

Communication is directly related to customer satisfaction. In the healthcare soft-

ware development sector, there are many stakeholders involved. SeniorDeveloper-HealthCo highlighted three main communication means that affect the customer satisfaction. First, a line meeting is conducted that involves all stakeholders, *“A line meeting is usually for half an hour or max one hour wherein all the stakeholders are available.”* Second, an agile presentation is held and the Senior Developer described it: *“During our agile presentation, hardness level presentations, they get to know what item is being done, what the complexity is, and what the objective of this feature is.”* Third, the retrospectives aid in achieving customer satisfaction. Retrospectives are devoted for the improvement of the agile software development process and for adaptation to changes that arise, such as the customers’ needs. SeniorDeveloper-HealthCo said: *“We evaluate. Everyone is involved, managers, the product owner and each team: what we have done, what went wrong, what went good? And what are the improvement areas that we need to identify?”*

4.3.5 Inter Team Communication Tools

Employees revealed two main communication tools that facilitate the communication process. In addition to face-to-face communication, practitioners used an information sharing platform design for their company and the physical Kanban board.

ShareGO is an information sharing platform used in the company for over 10 years to share documents and reports; it also acts as a software development network. We altered the name of the information sharing platform to ShareGO to protect the anonymity of the participating company. First, the reports are created after the sprint. They include the sprint details and individual sprint reports. Second, documents are shared and include, according to Architect2-HealthCo : *“burn-down charts, histograms, and other alternatives, that talk about how much of the backlog is done.”* On another hand, ShareGO is used as a software development network implementing a tracking system that is able to inform employees of defects on certain codes even if they are developing a completely different one.

The employees also use a physical Kanban board that aids in tracking the work of the different teams and employees. According to Developer-HealthCo: *“We can see all of the tasks which we have done for the current iteration.”* Furthermore, the physical board identifies the different backlogs and the priority of each. The SeniorDeveloper-HealthCo indicated: *“During the iteration we pick up the items, and it is those items that we put on to our board so that the board has a card, and each card has a title.”*

4.3.6 Approach to Non-Functional requirements

The health care sector in HealthCo does not follow pure agile software development methods. Instead, a mix of methods are used to guide the process. SeniorManager-HealthCo explained: *“We have found a little bit of midway between a very pure agile development and the old style of waterfall development.”* ProjectManager-HealthCo emphasized this by saying: *“So I won’t say that we are strictly following agile.”* In addition, the ProgramManager-HealthCo estimated the following: *“The Agile we are implementing is not through 100% of the lifecycle of the product, it is only through 70% or 80%.”* This is translated in the way HealthCo deals with non-functional requirements.

HealthCo mixes agile and traditional methods to deal with non-functional requirements. First, HealthCo associates different roles and titles to its employees reaching 16 titles. These roles include: product manager, line manager, project manager, sub-segment head, test manager, team master. . . Developer-HealthCo said: *“If we go by the pure Agile Scrum methodology, then we shouldn’t have any other role other than the product owner and the Scrum master.”* On the other hand, some employees take on double roles. For instance, the senior manager holds two responsibilities: line coach and project coach.

Second, the employees perform documentation regularly. One reason for the usage of documentation is the quality assurance. Architect1-HealthCo also indicated the relation between quality assurance and documentation: *“We deliver monthly*

reports to the technical quality person who delivers quality assists and technical assessment monthly and sometimes weekly.” Extensive documentation is used since there are some international regulations that the company has to comply with. For instance, Developer-HealthCo mentioned the FDA: *“There are documents because we are in the medical software [development industry] and are based on the FDA, Food and Drug Association of America. They have test specifications, safety specifications, design specs, and safety integration test specifications.”* Architect2-HealthCo pointed out several times where documentation should be used: *“The requirement spec at the beginning of the sprint, test validation at the end of the iteration, release software requirements, and ready and release documents.”* The SeniorManager-HealthCo specified: *“The large number of documentation complicates the tracking of non-functional requirements.”* The large number of documents create challenges in identifying dependencies between non-functional requirements. SeniorManager-HealthCo continued this list to reach 8 documents that need to be written on plans, testing, and results. ProjectManager-HealthCo proceeded to add how documentation differs from one country to another: *“Documentation in Germany is more of self-declaration to get your CE labelled... in China documents are converted into Chinese requirements.”*

4.4 Agile Implementation in Chosen Research Site in Lebanon

This section describes the baseline in each of the studied companies and provides an overview on agile implementation. Then, the external influences on agile implementation and agile concepts are presented. These influences either impede or enhance the agile implementation. Also, the customer involvement in the process, agile in regulated environment, and the approach to non-functional requirements are explained. The factors identified by practitioners that require improvement in the project are presented.

4.4.1 Description of the Baseline in each Research Site

The below section presents the results developed from the interviewees done at the studied agile implementing software development companies in Lebanon. It presents an overview of the agile methods used in each research site. From this, several similarities are detected.

Agile Implementation in LebDevCo1

Agile is implemented in LebDevCo1 in the business lines related to banks and financial institutions. The introduction of agile was deemed necessary with the creation of prototypes and new products. OppMan-LebDevCo1 expressed:

“We implement the main core values used in agile: prefer doing the interactions over focusing on processes and tools; to give the clients a working software rather than a comprehensive documentation; to engage the customer in reviewing periodically the delivered parts and to iterate the delivered parts.”

Implementing agile software development was introduced to LebDevCo1 through research done by managers. The managers felt the need to be more flexible with the customer's requirements and deliver faster. Thus, they chose to introduce agile methods to their traditional Waterfall work method. OppMan-LebDevCo1 said: *“It is a combination of our research and what we feel is really needed to do the work properly.”* DevelopDir-LebDevCo1 explained how time and knowledge stand in the way of familiarizing employees with all agile methods, *“Sometimes we don't have time and education and we just need to keep things running.”*

LebDevCo1 is selective with the choice of agile methods according to the background and attained experience. OppMan-LebDevCo1 said: *“Our experience has taught us that we should use these specific core values from agile.”* The agile methods used and their corresponding benefits were emphasized by the employees. The

concepts of sprint, demos in front of customer, and open communications where the major highlighted benefits of agile. OppMan-LebDevCo1 indicated the importance of communication and how it decreases rework and bottlenecks at the end of the process: *“I ask the team to emphasize on interacting with each other and not rely only on tools and processes.”* DevelopDir-LebDevCo1 observed that adjusting the product after each iteration decreased the time required to complete a project: *“We conduct a demo and we get the comments of the customer and we adjust thus saving time.”* BuisDelHead-LebDevCo1 and ClientEng-LebDevCo1 emphasized on the benefits of using iterations, breaking down the project into increments, and working in parallel. The former explained how agility increases accuracy and quality of delivery: *“Your prediction and estimation would be continuously reviewed in short term.”*

Agile Implementation in LebDevCo2

Agile is implemented in LebDevCo2 for all software development projects. LebDevCo2 develops on demand and customized software. LebDevCo2’s branch in Beirut is designed to serve customers in Europe and the Middle East. The need to be adaptable according to the customer’s needs drove LebDevCo2 to become an agile implementing software development company. Developers and product owners felt the need to implement agile since their products are custom made and customer involvement is key in the project success.

Agile methods are implemented such as sprint planning. Stand-up meetings are not always executed. According to Dev1-LebDevCo2, stand-up meetings are only done at the client’s side: *“At the client’s side, each day we have a stand-up meeting. When we are in Lebanon we cannot always do the stand-up meetings.”* This relation between implementing agile and the geographical location of the client was highlighted again by Dev2-LebDevCo2: *“Since we are working with a remote client, we are trying our best to apply agile.”* In addition, agile implementation is subjective and the choice of implementing certain agile methods varies according to each team. PO1-LebDevCo2 explained: *“One team would conduct one sprint*

planning meeting at the beginning of each sprint; while another team will conduct a sprint meeting and several refinement meetings throughout each sprint.”

Agile Implementation in LebDevCo3

Agile software development is implemented in LebDevCo3 which specializes in designing application and platforms for teenagers and young adults. One of their main projects is the development of an application that aids education seekers document their achievements and activities in a portfolio mode so that they can apply to college, find financial aid, and connect to each other.

LebDevCo3 employees aim at becoming 100% agile. Major resistance towards this implementation initiated from the board. The board of the company wanted to see the overall image and were not convinced with the incremental nature of agile. As PO-LebDevCo3, employee responsible for agile implementation, indicated:

“I tried to apply purely agile, even when the board wanted to see the global image. They want to set goals and see the bigger picture; they want to put their hands on features they see are important.”

Scrum was applied and ceremonies such as sprint planning and demoing were implemented. MarketingMan-LebDevCo3 described the morning meetings: *“We do morning meetings with the CEO that are supposed to be stand-ups. We go into extreme details with the CEO.”* Open communication was encouraged and several communication platforms were used to achieve this. These include Jira, Slack, Google Drive, and emails.

Employees in this company assumed that agile methods may be implemented if there were no resistance from the upper management. ContentCreat-LebDevCo3 expressed: *“The upper management’s continuous involvement in the agile implementation process and changing every detail hindered the application.”*

4.4.2 Overview of Agile Implementation

This section shows the characteristics that identify how agile is being implemented in the chosen research sites in Lebanon. The below presents the characteristics that either impede or amplify the agile process.

Lack of Understanding of Agile Methodology

Being a new concept in the Lebanese software development industry, understanding agile was a major struggle for employees. First of all, implementing agile was suggested by either employees who found the agile concept while researching or the CEO who heard about the concept in a different company. TeamLead-LebDevCo2 explained: *“While I was researching for a project, I found out about agile. I presented this concept to the team leader and he liked the idea. So the company decided to turn into agile.”* In another case, the CEO wanted to implement agile in the company after being introduced to this concept by a friend. The CEO revealed: *“I have an idea, general overview, but I don’t know exact details.”* The CEO wants to implement agile yet has initial knowledge and no prior experience with agile.

Second, of the three research sites, non had a certified agile coach who guides employees through agile implementation. PO-LebDevCo3 was responsible to implement agile in LebDevCo3 with no prior experience or certification. In LebDevCo1, agile was tailored according to trial and error methods. OppMan-LebDevCo1 explained: *“Unfortunately, we base our work on research that we do on our own. We never had a coach specialized in agile methodology. We do our own research and we let ourselves be guided at the same time by the real needs on the ground.”* The employees did not thoroughly understand the concept of agile. For instance, some interviewees had no knowledge on what agile ceremonies and artefacts were. They knew general concepts such as individual interactions over processes and working software over comprehensive documentation. Third, understanding agile and its importance was especially difficult for non-developers. PO-LebDevCo3 explained: *“The business development team and the marketing*

team did not understand the importance of agility.”

Implementing Agile in a Francophone culture

After being colonized by the French (from 1920 till 1943) and having the majority of population speaking and studying in French, practitioners find it difficult to learn and fully understand new agile methodology. OppMan-LebDevCo1 indicated: *“It was hard for me to understand the concept of agile especially that most of the research is done in English.”*

When introducing the agile concept of a ‘working software over comprehensive documentation’, several practitioners reported how it was such a foreign concept for them. The Francophone culture advocates documentation and this became an innate habit from early stages in their careers. TeamLead-LebDevCo2 explained: *“From when I was just a assistant at the office, we performed documentation. With agile this become different.”* DevLeader-LebDevCo1 explained how sometimes they get lost between the different cultures.

Combination Agile and Waterfall Methodology

None of the research sites under study fully applied agile. Sites used a combination of Waterfall and agile methods. Below is a description the methods used in each site.

LebDevCo1 follows a combination of Agile and Waterfall methodologies. As mentioned by OppMan-LebDevCo1: *“Inside the agile iteration we use the Waterfall methodology by respecting the sequence of processes.”* The sequence is explained by collection of requirements, design, development, testing and delivery. The use of agile depends on the type of life cycle followed. There are two groups of life cycles. The first group includes the repetitive project which is characterized by the similarity to previously executed projects. In this case, the teams follow a

traditional model and as PO1-LebDevCo1 described: *“The requirements we take them as Waterfall.”* The second group involves an adaptive or iterative life cycle characterized by novelty and innovation.

LebDevCo2 uses agile methodology when dealing with clients in Europe. Agile ceremonies are performed at the client’s side. Dev1-LebDevCo2 described: *“In Switzerland, at the client’s side, each day we have a stand-up meeting. When we are in Lebanon we cannot always do that.”* Team members will use other communication platforms to update each other; TeamLead-LebDevCo2 described: *“We will communicate what we are doing with the teams through chatting.”* On the other hand, when dealing with clients in Lebanon, LebDevCo2 does not implement fully agile methodologies.

LebDevCo3 aims at implementing agile but faces resistance from upper management. This lead LebDevCo3 to implement a combination of agile and Waterfall methodologies. The employees aim at implementing agile in order to enhance communication and enhance the productivity of the team members. On the other hand, the upper management, according to the employees, wanted to see a global picture of the requirements and interfered in the details. The overall requirements were given in the Waterfall method. Dev2-LebDevCo3 expressed: *“For me, micro management from the upper management, is the hardest challenge.”*

4.4.3 Influences on Agile Implementation Exerted by Non-team Members

This section describes various factors that affect agile implementation. These factors are exerted by the upper management and political and economic situation in the country.

Upper Management's Micro Managing

Dealing with the pressures of the upper management is one of the main challenges faced by employees. PO-LebDevCo3 said: *"Honestly, the biggest challenge faced is the upper management."* Practitioners revealed how the CEO was micro managing and continuously involved in the process. TeamLead-LebDevCo3 explained: *"The CEO changed his mind every 5 minutes and he micro managed."* ContentCreat-LebDevCo3 and MarketingMan-LebDevCo3 second that.

In regards to communication, the CEO requested to know all the details of every project. So the CEO would request a morning meeting to be informed of all the details. These morning meetings will last longer than advised. Details on stand-up meetings are available in section 4.4.4. Another request by the CEO would be to track all tasks being executed on JIRA. According to MarketingMan-LebDevCo3, *"We used to put things on JIRA but the CEO would never access it."*

Practitioners complained from the CEO's interference in how they lead their team and plan and execute their upcoming tasks. Dev2-LebDevCo1 described: *"I am not given the space to implement the work independently. So I had the CEO assign tasks to my team without me knowing. He was deciding our next steps even if I did not know."* Thus, Dev2-LebDevCo1, team manager, and Dev3-LebDevCo1 a member of this team described how they would use instant messaging tool (WhatsApp) in order to communicate without being interrupted by the CEO. They would plan their overall tasks and daily to-do lists.

Nepotism

Nepotism hinders the process and affects the employee motivation. Certain practitioners felt that no matter how hard they work, there are certain positions they cannot reach. Designer-LebDevCo2 described a personal experience (Designer-LebDevCo2 was reluctant at first, but decided to describe the situation after being assured that all data will be anonymized):

"I have been a designer for quite sometime in this industry. Yet I know I will not be able to progress in this company since the head of the designing team is the direct relative to the owner. This sometimes demotivates me."

Current Political and Economic Situation

The current political and economic situation in Lebanon has highly affected the software development industry along with the Lebanese people's motivation. Throughout the interviews, the employee morale and enthusiasm was highly affected. Lebanon has faced several difficulties, wars, and clashes throughout the years. But the economic situation is draining the people's enthusiasm and passion towards work. DevLeader-LebDevCo1 explained that employees come to work now with a full load of external worries. DataIntegration-LebDevCo1 expressed: *"I would like to see my country improve. When the country improves we improve."*

The upper management and employees feel that there is no other option but to succeed in their line of work. Employees are facing an extremely tough economic and political situation. Also, Lebanese are experiencing difficulties in being accepted to jobs outside Lebanon due to certain imposed sanctions. Thus, employees are eager to succeed in their current jobs.

Allocation of Tasks by Managers to Team Members

Managers and team leaders are responsible to assign tasks to team members. ProjMan-LebDevCo1 explained: *"We assign the global tasks for each team and then each team leader will assign the tasks to his team."* Some team leaders discuss the task distribution with their team members. DataIntegration-LebDevCo1 explained:

"I discuss with them the tasks. We can divide it the way they [team members] like. But I do the final picking. It is important that the distribution of tasks is

equal.”

This creates occasional tension between the team leaders and project managers from one side and the team members from the other. Team members would prefer they choose the tasks they want to work on. As Dev-LebDevCo3 describes: *“I dislike the idea that our leaders allocate the tasks. I would like to choose the tasks I have to do.”* Dev-LebDevCo2 added: *“It would motivate me if I chose my own tasks from each sprint.”* When tasks are allocated, some team members object. Yet some team leaders do not take this objection into consideration. ProjMan-LebDevCo1 expressed in Arabic: *“They don’t have the opportunity to say no.”*

Hierarchy in the Structure of the Organization

The structure of the organization is hierarchical which leads to bureaucracy. This delays the development process. The hierarchy reduces internal innovation due to the centralized power structure. ContentCreat-LebDevCo3 explained: *“The need to report everything to everyone makes it hard to be creative.”* Some managers even become territorial thus reducing collaboration with all members.

4.4.4 Agile Implementation in Relation to Ceremonies and Practitioners’ Roles

The practitioners’ perceptions on agile ceremonies and roles implemented are presented in this section.

Improper Implementation of Agile Ceremonies

Practitioners were asked on agile ceremonies conducted at their respective companies. Sprint planning, retrospectives and stand-up meetings were the core of the

discussion. Sprint planning was done by all three companies under study. All practitioners identified its importance towards the success of the sprint. Practitioners' perception on retrospectives and stand-up meetings are discussed in details in the below sections.

Stand-up Through the interviews, it was obvious that stand-up meetings, if implemented, were not executed correctly. Morning meetings with the CEO, top managers, and team leaders replace stand-ups. PO2-LebDevCo1 explained: *"In this meeting, we put all our active items. The CEO starts and then asks each team leader what they have for the day."* DataIntegration-LebDevCo1 indicated that the morning meetings take around 30 minutes and: *"They should have a different formulation. We have to discuss what we are doing on this day. We usually discuss the project in general. It is sometimes repetitive."*

Similarly, MarketingMan-LebDevCo3 said: *"It is a tiring way to start the day"*. Similar to the CEO of LebDevCo3, LebDevCo1's CEO took the lead in the stand-up meetings. As per PO-LebDevCo3: *"He [CEO] would start asking what we are doing and enter a debate. He could not put the limit to his discussion."* After an all-inclusive meeting, the product owners and team leaders were not able to conduct specific stand-ups with their team members: *"It was impossible for me to do that again with my team."* TeamLead-LebDevCo3 gave a thorough description and overall view of the daily morning meetings:

"The people felt that it was agony to do the daily stand up. Sometimes it would take an hour for us to finish a stand-up. We would discuss things that should be discussed in meetings because they require more depth. A daily meeting should be dedicated for everyone to align."

Of the interviewed practitioners, two were against conducting stand-up meetings. The two practitioners indicated that conducting stand-up meetings on a daily basis is counterproductive since most tasks take more than one day to finish. Thus, it becomes repetitive to conduct them. Dev5-LebDevCo1 explained: *"Sometimes we will have tasks that will take several days, so it is counterproductive to do them"*

daily.” ProjMan2-LebDevCo1 added to the above mentioned idea: *“Stand-ups should not be on a daily basis. Because sometimes a business that is clear will require 3 days to develop and no need for updates.”*

Retrospectives Through our interviews, I detected confusion around retrospectives. Retrospectives are defined as meetings held at the end of each sprint iteration. During retrospectives, team members will reflect on the sprint and their performance. Team members will constructively and collectively analyse their performance. When asked about retrospectives, 21 interviewees did not know what retrospectives are. So, as the interviewer, I had to explain what happens during retrospectives and how they are conducted. Dev1-LebDevCo2 responded: *“We did not do retrospectives; we did not have formal evaluations. We just informally saw what was wrong.”*

As for the interviewees who responded to the question regarding retrospectives, they lacked a clear understanding of what they are. Every response was different from the other and misconceptions on retrospectives were clear even in the same company. ProjMan2-LebDevCo1 explained: *“We do our weekly meetings and we learn from our mistakes. We learn from each other so that we would not face the same things. We do not wait till the end of the sprint.”* Dev5-LebDevCo1 added: *“We do weekly meetings mostly to clarify the summary, reports, and changes that have been done during this week.”* Another point of view was given by Dev1-LebDevCo1: *“We do an evaluation meeting each 2 days, not at the end of each sprint. We stop, we change, if we need to; it is a continuous work.”*

While interviewing practitioners in LebDevCo3, a difference between the employees and the CEO’s input on retrospectives was clear. The CEO expressed: *“We do not do retrospectives, we evaluate the performance informally.”* On the other hand, the employees said that they perform retrospectives in each team. Again, inconsistency was detected; where one team would conduct retrospectives each week, and the other team will conduct them after each sprint. ContentCreat-LebDevCo3 explained: *“We used to evaluate our strategy. Is it working? Should we adopt*

another one?" Practitioners explained that each team member will react differently during retrospectives. PO-LebDevCo3 said: *"Some people accepted the idea of retrospectives and work evaluation."* Others found it *"mundane"* as per TeamLead-LebDevCo3. PO-LebDevCo3 gave a through explanation:

"I would personally come up and say that something went wrong. Because if you don't look into sprints and what went wrong, it won't work. The problem is that there were people who resisted talking because they were afraid."

Troubled with Public Self-Reflection The idea of public self-reflection make Lebanese practitioners feel vulnerable. Critical self-reflection is difficult to share. This is directly related to two factors, personality of the Lebanese practitioners and the fear of showing mistakes in front of the upper management. Dev1-LebDevCo1 indicated: *"I feel like evaluation meetings are a way to evaluate our performance with respect to other employees."*

Practitioner's Roles in Agile

Agile advocates three roles, product owner, scrum master, and self-organizing team. The role of the product owner is present in all research sites. PO1-LebDevCo1 explained: *"I work on the product concept and follow up with different stakeholders."* On the other hand, none of the research sites had a scrum master. This is due to the lack of knowledge on agile methodology.

Teams lack self-organizing skills. Instead teams are managed by team leaders or managers of the company. Their work is hindered by the organizational structure characterized by hierarchical values, referred to in section 4.4.3. For instance, team members are assigned tasks by their managers as described in section 4.4.3.

Developers Sense they are Pressured Interviewed developers revealed the amount of pressure they face with the increase in their responsibilities. They

understand the business requirements, meet with client, design the screen,, and apply tests. When a developer gets involved with the business requirements, they can give a more valid, directed and targeted solution. Dev3-LebDevCo1 indicated: *“It is not our task to do everything. In other companies, the developers just take the requirements and do their job.”*

Communication between Practitioners

Communication is highly encouraged at the studied sites. Diverse communication channels are open for employees at the company. Weekly face-to-face meetings are held specifically for each project and reports are shared with all stakeholders.

Face-to-face communication is always present among members who are in the same location. When practitioners need to communicate and are present in different geographical locations, they use online video conference call applications available.

DevelopDir-LebDevCo1 highlighted the importance of delivering the message: *“The target is that they deliver the message not the medium.”* And OppMan-LebDevCo1 agreed. The usage of communication channels depends on the needs and message being delivered. If it is a simple update, then instant messaging platforms are used. If it is formal, then emails are sent and meetings are held. The DevelopDir-LebDevCo1 explained: *“Sometimes, I have to send just an informal message; I use WhatsApp.”* PO2-LebDevCo1 added: *“We have a WhatsApp group for the team. We use it to communication and if someone is on vacation.”* CEO-LebDevCo3 indicated that slack is used for documentation which contradicted with the input of PO-LebDevCo3 who indicated: *“Slack was more of a communication platform than a documentation one. The thing that is written on Slack is not that official.”*

Communication between Teams with Members in Different Geographical Locations

Interviews revealed that some team members where located in different

geographical locations while working on the same project. Two research sites, LebDevCo2 and LebDevCo3, outsourced people to perform various tasks. These included graphic designers, advertising agents. . . . Another research site, LebDevCo1, had branches located in different districts in Lebanon. Communication between these members is smooth. Surprisingly, from all the interviewed practitioners, who worked with outsourced employees or with team members in different company branches, none claimed that they were facing communication difficulties due to geographical separation.

Different communication platforms were used to communicate with employees who were outside the office such as Slack, email, calls, and video conferencing. Dev2-LebDevCo3 said: *“Every time we needed something from someone we used Slack, whether they are in the office or outside. The communication was very easy.”* Designer-LebDevCo2 used emails to send reports and communicate with their outsourced employees.

LebDevCo1 has three branches in Lebanon and teams are distributed in different branches but the management is centralized. From the HQ to the North Office, it takes approximately 1 hour and 40 minutes. From the HQ to Zahle office it takes approximately 1 hour 30 minutes. Developers, business delivery units and data modellers are spread across the three different branches. Dev5-LebDevCo1 said: *“We work remotely because we have the technology and the communication tools to communicate and prepare the data.”* The new technological platforms helped in the process where team members can work remotely and communicate through online platforms, video calls, screen sharing, and VIOP phones. Practitioners also perform site visits, only when needed. ProjMan2-LebDevCo1 explained: *“If needed the team members come and stay at the main branch.”* Managers try to minimize the movement between branches as much as possible in order not to lose time. Especially since the branches are located in different districts, no public transportation is available in Lebanon, and traffic jams in Lebanon are constant.

BuisDelHead-LebDevCo1 gave a personal example where she extended her maternity leave and lead her team completely remotely from her home. BuisDelHead-

LebDevCo1 claimed:

“We are delivering. We are connected. . . We don’t have this feeling that there’s something wrong or this person is not with us if they are working remotely. Sometimes we don’t know if the person is in the North District or in the headquarters or elsewhere.”

Communication Vertically Across Hierarchical Positions A detected challenge is the information sharing vertically across the hierarchical positions. Some practitioners find it difficult to communicate with others who are in higher positions. Dev5-LebDevCo1 indicated: *“I find difficult to communicate especially with my managers who are in higher positions.”* Employees find it difficult to inform the upper management if something does not go as planned. CEO-LebDevCo3 stated: *“There was resistance from the employees to come to me and say that something was going wrong. Once I found out when it was too late and nothing can be done.”*

Vertical communication is hindered by fear mainly. Due to the economic situation, employees fear that one mistake may cause them to lose their job. Dev1-LebDevCo1 expressed: *“Most of the time I try to fix the problems myself because I am afraid if I did a mistake I might lose my job.”*

4.4.5 Time Factor and its effect on Agile Implementation

This section looks into the time factor and its effect on agile implementation.

Struggle to Respecting Time

Respecting time is considered one of the major drawbacks in the Lebanese industry. Stakeholders show low discipline when it comes to respecting time. Respecting deadlines and meeting due dates is a challenge mentioned by several interviewed

practitioners. PO1-LebDevCo1 explained: *“Respecting the deadlines and ETAs is a daily challenge.”* Unexpected problems and errors may hinder the process. PO2-LebDevCo1 described a case where the server was down for 3 days which caused “a lot of stress”. ClientEng-LebDevCo1 encountered this difficulty with employees who don’t specify a specific delivery date: *“Colleagues would say this will be done next week. Monday is next week and so is Friday. I know it’s an estimate but just give it in the most precise way.”*

Estimating Time for Completing a Task

The time estimation and prioritization of tasks is done by the product owner. The first high level estimation is done by scanning the requirements. The team’s experience, clarity of requirements, and complexity of the task are taken into consideration. Then, a percentage of time increase is added to mitigate the risks. The major risk that needs mitigation, according to practitioners, is the client’s cooperation throughout the project flow. Dev2-LebDevCo1 described such cases with the client: *“The client will have a delay in answering back our requirements. . . the executives will not understand that there had been a delay from their end, not because of us.”* Further findings on the client’s effect on project execution timeline is discussed in Section 4.4.6.

Task estimation is performed by managers and a difference was detected in the time estimation of tasks between different practitioners. Certain team leaders took into consideration the employee that a specific task is allocated to and adjusted the time accordingly. ProjMan-LebDevCo1 expressed how time estimation respects the developer’s capabilities: *“We estimate based on our experience and the person we are dealing with. For example, with one person it [the task] will take 1h for another it will take 4 hours.”* PO2-LebDevCo2 did the same. On the other hand, other practitioners evaluate the task regardless the task allocation. ProjMan2-LebDevCo1 claimed: *“When we are asked about the level of effort from the business department, we don’t tailor it to the person who is going to do it. We say it need a senior or a junior member to do.”*

4.4.6 Customer's Involvement in the Agile Process

This section describes the practitioners view on the customer involvement in the agile process. It also shows the effect the client's feedback has on practitioners.

Practitioners' view on Lebanese Clients vs Western Clients

Practitioners expressed the difference in working with different clients. They characterized them into two groups: the Western clients, North America and Western Europe, and Lebanese clients. Practitioners felt that it is easier to work with the Western clients in comparison to the Lebanese clients. DevelopDir-LebDevCo1 explained how transparency is key with the Western market: *"Working with Western market, is easy. What you have to do is to be transparent and say things as they are. Working with the Lebanese market is harder."*

This is due to factors pointed out. First, the Western clients value time and provide a logical estimate for task completion. ClientEng-LebDevCo1 explained: *"In their business culture, they are committed to the set dates. Also, they are very reasonable in discussing the time required to complete each product."* Lebanese clients tend to put unrealistic deadlines for their suppliers. Second, they are structured, well organized, and cooperative. BuisDelHead-LebDevCo1 explained: *"They did their due diligence by preparing the requirements. And they work with you in order to have their work done properly and with high quality. You will not find them as clients you'll find them as partners."* Third, they have a collaborative nature and seek to have a successful product. DevLeader-LebDevCo1 said: *"If you have something, a bottleneck in certain place, and you give a good argument, they will understand and they will help you solve it."* Fourth, Western clients are specific about their requirements and all the associated aspects. DevelopDir-LebDevCo1 explained: *"They know exactly what you can do and what they want. They know their budget and give you the righteous amount for your work."*

On the other hand, dealing with Lebanese clients was harder and less efficient.

Participants did not generalize about all clients in Lebanon. They either stated the former or named specific inefficient clients, in their opinion. Dev1-LebDevCo1 expressed: *“It is about some Lebanese clients and their culture not all of them. We struggle to achieve communication and get the requirements.”* PO1-LebDevCo1 explained that delays in project time do not only affect the client, but also affect the software development company that is keeping resources dedicated for a certain project longer than planned. The coming paragraphs explain several reasons why agile software development companies find difficulty in achieving results with Lebanese clients.

Lebanese clients show low levels of responsiveness to the requests from the software development company. This causes major delays in the due date. ContentCreat-LebDevCo3 explained: *“The collection of data was the most difficult part.”* BuisDelHead-LebDevCo1 expressed how clients delay in presenting the required information: *“The client will delay in answering back our requirements to get their feedback. So this will lead us not to meet a deadline at a certain point.”* With clients responding poorly, gathering needed information was a challenging process. ContentCreat-LebDevCo3 said: *“The collection of data was the most difficult part.”* CEO-LebDevCo3 explained a case where their client was a university in Lebanon. They faced large delays in project due date due to the irresponsiveness of the client: *“We knew that the educational system is slow by design but we were not expecting 3 months of delay. Every time we requested data or a specification it would take days and weeks for them to respond.”* Dev2-LebDevCo3 described a case they are currently facing: *“In this project we are expecting data from a client and we have been expecting it since Tuesday and it is Thursday from the second week now and we haven’t received it.”*

Managers expressed how certain Lebanese clients stated their initial commitment to the development process; but through the process, clients’ enthusiasm decreased and their commitment became minimal. ClientEng-LebDevCo1 claimed: *“The other thing that I’ve noticed with clients in Lebanon is the availability and commitment of their staff. It is not always as promised.”* PO1-LebDevCo1 also explained that some employees at the client’s side have multitasks thus: *“Sometimes the cus-*

tomers cooperate and sometimes they have other priorities.” ProjMan-LebDevCo1 went further and indicated that sometimes you should analyse how the client is feeling in order to ask for data: “The clients are very moody. You have to know the mood of the client today and how you should approach them.”

Practitioners indicate how they try to find innovative ways to keep the client engaged and motivate the client to respond promptly. CEO-LebDevCo3 expressed: *“We have to sometimes create ways to put pressure like sending emails, calling and texting in the same day.”* Some practitioners accept the irresponsiveness of the client and try constantly to let the client understand how they are involved in the agile process:

“Our project managers are pushing the client to validate this process to sign off the requirements, to do the testing, to do a regular meetings, weekly calls, weekly meetings, face to face meetings... We are pushing the client and we are guiding the client to follow this agile model.”

Regardless, the clients tend to blame the team leader or product owner when a delay in the project occurs. BuisDelHead-LebDevCo1 explained: *“Sometimes the executives from the client side will not understand that there has been a delay in a certain step not only because of us but because they did not give the requirements.”* Thus, practitioners will aim to document everything and keep track of all requests. DataIntegration-LebDevCo1 explained: *“Everything should be emailed, everything should be documented, even if you are friends with them.”* Occasionally, the product owner will aim to send a reminder to all stakeholders; *“We will send a reminder email in which all the stakeholders are aware that the reason behind this delay is your [the client] end and not us.”* Dev4-LebDevCo1 explained.

Furthermore, at the client’s side, employees require the presence and interference of the manager at every bottleneck. TeamLead-LebDevCo2 explained a situation where a change in the colour used in the design was needed and the client executive requested a meeting with the manager regarding this issue. OppMan-LebDevCo1 indicated: *“I follow up personally most of the time until this matter has been resolved. The clients always request the manager.”*

Reaction to Client's Negative Feedback

As a normal process of any software development project, customers will request changes and may present negative feedback to customers. Even though the product owners clarify all requirements at the beginning and present a detailed document containing all deliverables, methodologies, technologies and timeline, the customer's reaction is unpredictable. DevelopDir-LebDevCo1 explained: *"We send the functional requirements and we get the sign off before the delivery. This does not mean we do not have some bottlenecks or objections from the customer."*

On their reaction to the customer's negative feedback, most practitioners revealed that they try to tackle it objectively. First the facts are weighed and a root cause analysis is performed. A corrective plan is put into action and decisions are taken and executed accordingly. BuisDelHead-LebDevCo1 explained:

"We see if we have to solve the problem from our side, if we have a problem from inside our company. Or if it's from the client side, we call and prepare the documentation and the material that proves that it is not from our side."

Most practitioners indicate that they *"Don't take it personally"* and they don't even have time to react or get affected by the feedback. Two practitioners ProjMan2-LebDevCo1 and TeamLead-LebDevCo2 indicated that they get affected and it takes a day to get back to full functionality. TeamLead-LebDevCo2 indicated: *"Constant negativity is nerve racking."*

4.4.7 Agile in Regulated Environment

Software development companies are faced with regulated environments when developing for the banking sector. The Lebanese banking industry plays a key role in the Lebanese economy where banks continue to dominate the financial system of the country and are major providers of credit to individuals and businesses. Thus, developing software for the banking sector is of high importance and falls under

extensive security, restrictions, and regulations.

In the regulated environment, specifically in the banking sector, data security is of high importance. Thus, employees are faced with certain restrictions that might hinder their progress. First, the financial institutions work under high levels of security in regards to the data and the people entering their work premises. POBank-LebDevCo1 explained: *“They only give us fake data. Also, we would have a prior agreement on the people who are allowed to enter the banks and the access they are granted. They also restrict us to what we are allowed to observe.”* This complicates the process especially when testing. The developers are given fake data to test the software developed. Then, they are allowed only a certain number of tests with real data in the bank premises. According to DataIntegration-LebDevCo1, *“The data and personnel restrictions complicate the process even more.”* Occasionally, developers, who were not granted access at the contract signing, will need to enter the bank’s premises to solve a certain bottleneck. This would require a new authorization request which will take a couple of days; *“Such procedures will cause delays in the testing process.”*

Second, the banking sector operates under a strict and rigid hierarchy. This centralizes the power structure which delays the process. Since every time a request, requirement, or data is needed, the developers are hindered with *“lack of delegation and heavy bureaucracy”*, according to Dev5-LebDevCo1. DataIntegration-LebDevCo1 indicated: *“In the banking system, they have the hierarchy and each employee is afraid of his boss.”* Also, the business analysts are faced with definition problems. DataIntegration-LebDevCo1 gave an example:

“Maturity date for us is when the next payment is due, for the bank it is the date when the loan ends. So we face some issue with these names. We need to approach the idea between us and the client in order to reach a middle point.”

4.4.8 Approach to Non-functional requirements

Agile implementing software development companies in Lebanon aim to differentiate between functional and non-functional requirements. The functional requirements are gathered by business analysts, while the non-functional are assessed by the technical personnel. OppMan-LebDevCo1 stated: *“To apply these functional requirements, we should from the beginning of the project agree with the client on several factors.”* These factors include the technologies, data base, application server, middle wear, data integration vehicles, security requirement, stress test, quality test ect. . . Developers described cases where clients ask detailed questions on the development process. ProjMan2-LebDevCo1 described: *“Sometimes we get involved with the meetings with the clients and we start talking about technical issues.”* This helped both the clients clarify their requirements and specification and developers understand the development conditions and ask questions that would assist in identifying the both functional and non-functional requirements.

As for the non-functional requirements, technical personnel aim to identify them. It is mostly done through experience. Dev3-LebDevCo2 explained: *“We identify the non-functional requirements from our experience with previous projects.”* The non-functional requirements are dealt with using a mix between agile and traditional methods. LebDevCo1 has developed a checklist to assess non-functional requirements. ClientEng-LebDevCo1 expressed the importance of having experienced personnel when analysing non-functional requirements: *“We have a checklist. But you cannot give this checklist to anyone, they have to have experience.”*

4.4.9 Factors that Require Improvement

Through the interviews, participants highlighted factors they would like to see improved in the agile software development industry in Lebanon. Below is a table that highlights the key performance indicators (KPI) indicated by participants. For each KPI, a relative quote is chosen to further explain ad evidence.

KPI	Extracts form Interviews
Conferences on agile methodology	We should have first conferences about agile methodology. We should conduct meetings to know how to efficiently apply agile methodology in the Lebanese environment. -Dev1-LebDevCo2
Quality of Deliverables Shared Between Teams	The more the development team asks me questions, the weaker the document quality. The document that should be presented should contain all the details and all the questions that might be asked by the development team. – PO1-LebDevCo1
Time to deliver software	The most important thing is to maintain the due date. – Dev3-LebDevCo2
Number of Bugs	There is no specific number of bugs but there is the regression of the bugs to the process. There is no zero bug but their number and intensity should decrease. – PO2-LebDevCo1
Minimize Bureaucracy	We find that we are in a bureaucracy box. We need to be flexible in communicating things. We need to take a quick action to solve a problem. But they would tell you send an email. – Dev5-LebDevCo1
Minimize Command Chain	Decrease the number of people you need to go to in order to solve a certain issue. – PO1-LebDevCo2
CEO's Micro Management	The morning meeting will take more than 30 minutes because the CEO want to know every single detail. – Dev1-LebDevCo1
Task Assignment	One needs to know how to assign tasks to employees...not an easy process with the culture here. – ProjMan-LebDevCo1
Cooperation with the Client	Everything should be improved with the relationship with the client. – DevLeader-LebDevCo1
Decrease Number of Reports	We produce a large number of reports that only serve to inform other employees on our work. – PO2-LebDevCo2
Appreciate the Work Done	At the management level, they don't look at the tasks achieved by each resource, they look at the benefits and I don't feel appreciated. – Dev1-LebDevCo3
Training	We needed to be coached on Agile Software Development.

Table 4.2: Factors identified for improvement by practitioners

4.4.10 Inhibitors and Exciters of Agile Implementation

This section aims at highlighting the inhibitors and exciters in the agile implementation process in Lebanon using the TIC model. The TIC model is used to understand the exciters and inhibitors in each dimension for agile implementation [135].

The exciters and inhibitors are identified from the list of factors that influence agile implementation in the studied context. The factors identified in this chapter are analysed and grouped across the three dimensions in the TIC lens explained in section 2.6.4. Each dimension is then separated into two directions. Then in each direction, these factors are divided into inhibitors and exciters. For example, Six factors were identified across Dimension A linking the capabilities approach to institutional theory: micro management, task allocation by managers to employees, will to implement agile, external support, ability to communicate across geographical locations, and lack of agile methodology understanding. The first four factors were allocated to the direction flowing from Institutions theory to Capabilities Approach; while the last two factors were allocated to the direction from Capabilities Approach to Institutional theory. The factors are then grouped into inhibitors and exciters as shown in figure 4.3.

Virtuous and vicious circles were identified that enhance or diminish the role of agile methodology. The study identifies positive reinforcements between factors that improve the effectiveness of agile methods. An example of a virtuous circle is the exciter: inclusive and detailed sprint planning (ICT to capabilities) leading to the creation of software products (capabilities to ICT). The various communication tactics act as exciters when used along with a rich variety of communication technologies between teams across different geographical locations. Although this case is not cross-country with no time difference, yet employees cross long distances in personal cars with no public transportation available. These exciters

enable team members to overcome distance, religious, and cultural differences.

Inhibitors create vicious circles that undermine the agile software development process. For example, lack of agile methodology understanding (capabilities to institutions) leads to task allocation by managers to employees (institutions to capabilities). In addition, there are loops that hinder the correct implementation of agile methodology. For instance, the combination of agile and waterfall methodology (inhibitor from ICT to institutions) leads to the task allocation by managers to employees (inhibitor from institutions to capabilities) which in turn lead to employee resistance (inhibitor from capabilities to ICT). These closed loops obstruct agile implementation.

There was a difficulty in assigning a satisfying illustration of an exciter from institutions to ICT. This may be due to the current economic and political crisis in the country. Also, the recognition of the institution's role in support of ICT is less fortified.

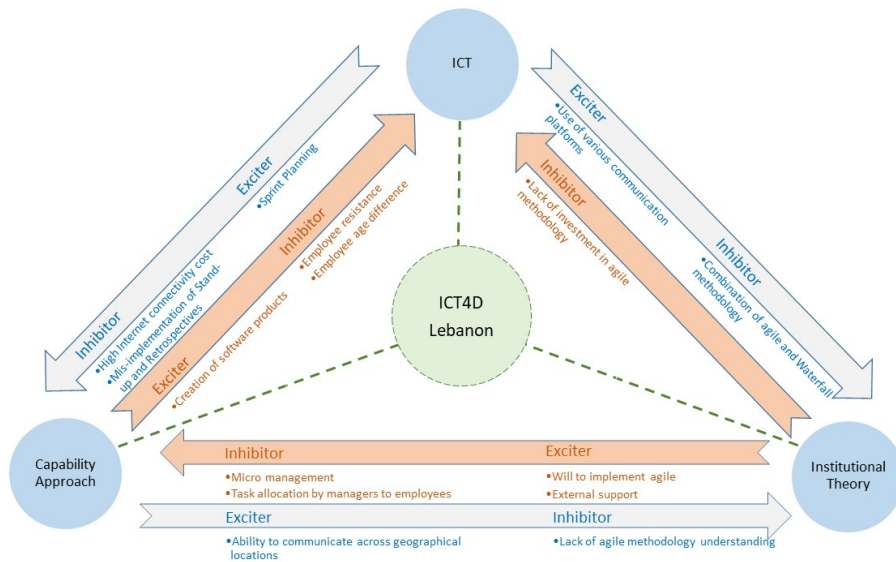


Figure 4.3: TIC lens adapted to the Lebanese context

4.4.11 Influences on Agile Implementation in Lebanon

All the above findings, the TIC model, and available literature are examined and combined to form a list of influences on agile implementation in the studied companies in Lebanon. Influences either impede or amplify agile implementation. These are then divided into external factors, characterized by an uncontrollable nature, and internal factors, controlled by the company's practitioners. Figure 4.4 lists all the influences

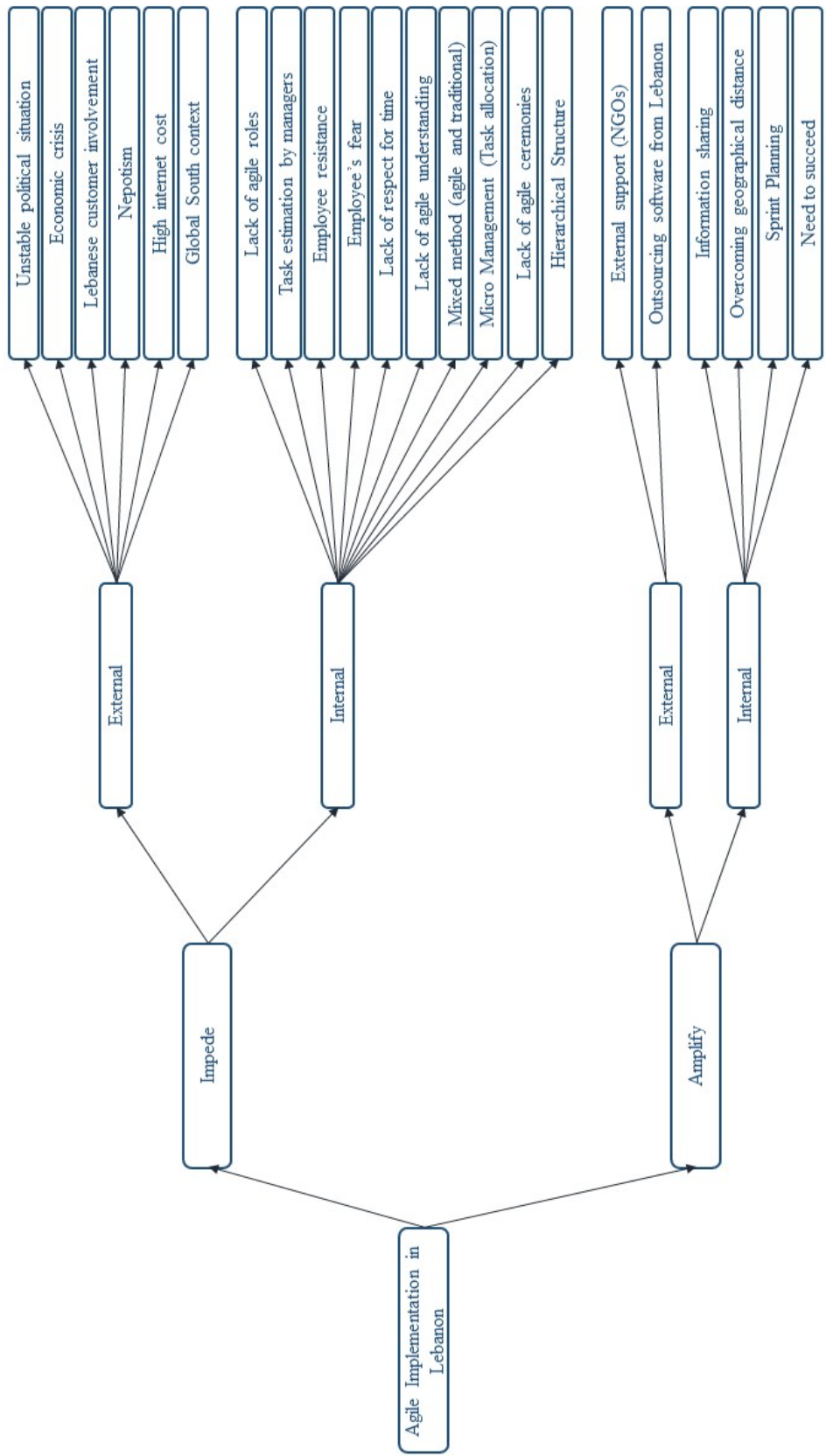


Figure 4.4: Influences on agile implementation in Lebanon

4.4.12 LebDevCo1 Process Flow

Figure 4.5 uses four swim lanes to display the main actors in the agile software development. The swim lane diagram is drawn for LebDevCo1 since this is the selected research site to conduct the intervention. This swim lane diagram is used as a representation for the three research sites. The model starts with the customer’s input on product requirements and ends with the demo to the customer. Activities that are placed on the borders of the swim lanes indicate that the activity is performed by the actors sharing the border. The diagram shows how the customer is minimally involved in the product development process. The model shows how the process lacks the input from the scrum master and lacks ceremonies, retrospectives and daily stand-ups. The model also reveals the lack of self-organizing teams especially since they are not involved in the estimation and prioritization of tasks. The diagram also shows the micro management of the CEO.

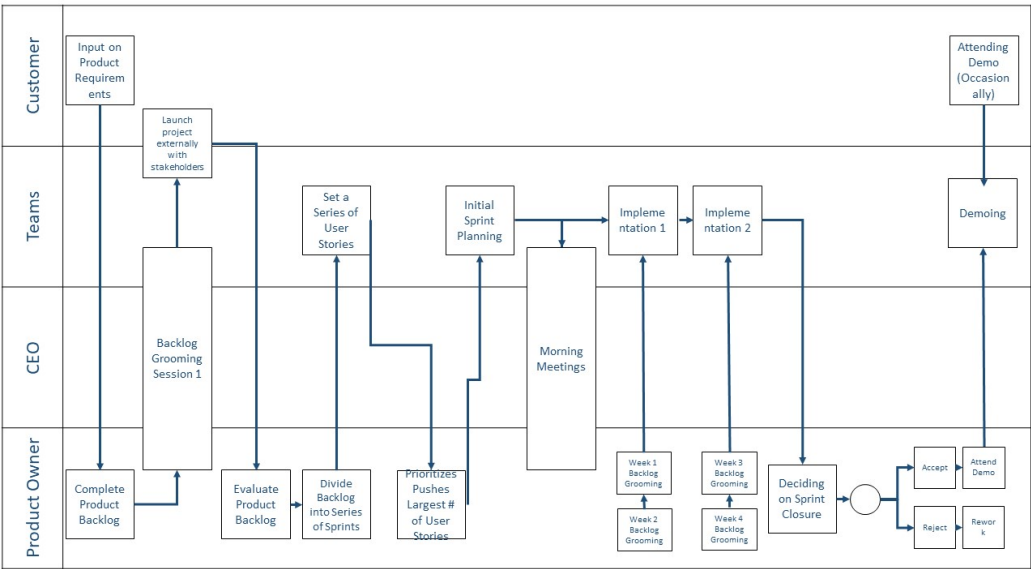


Figure 4.5: Swim lane diagram portraying the process flow for LebDevCo1

4.5 Summary

This chapter presents the findings from the two preliminary studies conducted at DevelopCo and HealthCo. For each research site, this chapter presents the process flow portrayed through the swim lane diagram, the communication at the inter-team boundary, team and customer collaboration, and the approach to non-functional retirements. In addition the first section presents the difference in agile implementation between the two geographical locations of DevelopCo. Also, the second section presents the employee training and development at HealthCo.

This chapter also presents a detailed explanation on agile implementation in the chosen SMEs in Lebanon. The chapter highlights the current status of agile implementation and reveals the challenges faced. In addition, this chapter reveals a list of the influences in Lebanon that either impede or amplify on agile implementation. These influences will be tackled in Phase 3b when creating the model.

Chapter 5

Cross Case Analysis Findings, Model Creation, Agile Tailoring to the Chosen Lebanese SME Context

5.1 Introduction

This chapter examines agile in the Lebanese chosen research site through implementing a cross case analysis (Phase 3a) and creating a model (Phase 3b). The cross case comparison was conducted between the Lebanese companies, DevelopCo, and HealthCo. It involved manually overlapping the swim lane diagrams previously presented to compare and contrast the interaction and information flows. This produced a comparison between the different agile concepts. Also, the communication means preferred by practitioners were compared along with the approach to non-functional requirements. The level of customer involvement in each site was also compared.

The cross case findings, along with the memos on agile implementation in selected research sites in Lebanon, acted as an input to create the model that will be

implemented in the chosen Lebanese SME.

This chapter also reveals the agile tailoring model fitted to the Lebanese SME context. The starting point is the swim lane diagram presented in Section 5.3.3. The creation of the model was an iterative process. This chapter describes the intervention performed in implementing agile methodology in the chosen research site, LebDevCo1. The intervention demonstrates the changes in the daily scrum meetings, retrospectives, self-organizing teams, scrum master, customer involvement, and task estimation. Finally, a post implementation evaluation and analysis is presented showing improvement in productivity levels. Quotes are used from post-implementation interviews with the members of the participating teams. This is Phase 4.

5.2 Cross-Case Analysis Findings

This section presents the findings from the cross case analysis conducted by comparing and contrasting agile implementation in the chosen Lebanese sites, HealthCo, and DevelopCo. Since the studies of HealthCo and DevelopCo were preliminary studies, not all influences identified in the in-depth Lebanese research sites were included in the cross case analysis. Table 5.1 shows the influences identified in the Lebanese research sites and compared with the findings from HealthCo and DevelopCo. The first column indicates the respective sub-section where the cross case findings are portrayed.

Section	DevelopCo	HealthCo	LebDevCo1/2/3
6.2.1	Implementing all agile roles (Section 4.2.1)	Implementing all agile roles (Section 4.3.1)	Lack of agile roles (scrum master and self-organizing teams) (Section 5.5.2)
	Implementing stand-ups, retrospectives, and sprint planning (Section 4.2.1)	Implementing stand-ups, retrospectives, and sprint planning (Section 4.3.1)	Implementing only sprint planing (Section 5.5.1)
	Pokering techniques (Section 4.2.1)	Pokering techniques (Section 4.3.1)	Task estimation by managers (Section 5.6.2)
6.2.2	Communication is a challenge (Section 4.2.3)	Communication is a challenge (Section 4.3.3)	Communication is extensively practised (Section 5.5.3)
	Diverse inter-team communication (Section 4.2.5) tools	Diverse inter-team communication tools (Section 4.3.5)	Diverse inter-team communication tools (Section 5.5.3)
	Overcoming geographical distance (Section 4.2.2)	-	Overcoming geographical distance (Section 5.5.4)
6.2.3	Agile methods (Section 4.2.6)	Mixed methods approach (agile and traditional) (Section 4.3.6)	Mixed methods approach (agile and traditional) (Section 5.3.5)
6.2.4	Team and customer collaboration (Section 5.2.4)	Team and customer collaboration (Section 5.3.4)	Poor Lebanese customer involvement (Section 5.7.1)

Table 5.1: Influences in Lebanese organizations cross compared with HealthCo and DevelopCo

5.2.1 Interaction and Information Flows

This sub section compares the process flow, focusing on one iteration, for DevelopCo, HealthCo, and LebDevCo1 (representing Lebanon) using the swim lane diagrams shown in Figures 4.1, 4.2, and 4.5 respectively. The swim lane diagrams were derived from the memos deduced from the transcribed data. An initial diagram was drawn from the memos that described the information flow. Then, constant comparison technique was used to complete the diagram through accumulating information from memos that are not directly related to information flow. Numerous differences and similarities were identified when drawing the swim lane diagrams. These revolve around related influences identified in Chapter ?? section 4.4.11 which are lack of agile roles, lack of agile ceremonies, sprint planning, task estimation by manager, and lack of agile understanding.

Table 5.2 demonstrates an inventory of the similarities and differences between the three organizations in regards to agile concepts. This table is deduced from the swim lane diagrams. The x in each cell represents the implementation of a certain concept.

Agile concepts		DevelopCo	HealthCo	LebDevCo1
Roles	Scrum master	X	X	-
	Product owner	X	X	X
	Self-organizing teams	X	X	-
Ceremonies	Customer input on product requirements	X	X	-
	Evaluate product backlog	X	X	X
	Divide backlog into series of sprints	X	X	X
	Set series of user stories	X	X	X
	Pokering user stories	X	X	-
	Sprint planning	X	X	X
	Weekly backlog grooming session	-	X	X
	Daily stand-up	X	X	-
	Demoing internally	X	X	X
	Demoing	X	X	X
	Retrospective	X	X	-
Artefacts	Program Governance	X	X	X
	Product Artefact	X	X	X
	Release Artefacts	X	X	X
	Sprint Artefacts	X	X	X
	Feature Artefacts	X	X	X

Table 5.2: Cross case comparison of agile concepts used in DevelopCo, HealthCo, and LebDevCo1

To begin with, the swim lane diagrams revealed the absence of a scrum master role and respective coordination activities conducted by the scrum master. This was

due to the lack of understanding and knowledge on agile roles.

This lack of understanding and hierarchical structure also revealed the decrease in the product owner responsibilities, represented in the added interaction and information flows in the product owner lane, with the increase of agile adaption methods. The product owner in DevelopCo conducted 4 independent activities. These activities include completing the product backlog, pushing user stories, and attending and deciding on demo. In addition, the product owner is allocated additional responsibilities in HealthCo. As shown in figure 4.2, the product owner communicates and coordinates with the Healthcare Regulatory Agencies who provides the product owner with the necessary rules and regulations to ensure compliance. The product owner in LebDevCo1 was given even more responsibilities and conducted 7 independent activities. These activities increased the product owner responsibilities thus mixing the product owner roles with a manager's role.

The lack of agile ceremonies, particularly stand-ups and retrospectives, were identified in the LebDevCo1 swim lane diagram. These caused higher interference by the upper management in daily activities such as replacing stand-up meetings with long morning meetings with the CEO. On the other hand, the sprint planning ceremony was performed. Although, team members did not have input on task allocation and estimation.

5.2.2 Inter-Team Communication

Inter-team communication was a major difference detected between the studied Lebanese companies on one side and DevelopCo and HealthCo on the other. Communication in the studied Lebanese companies is smooth and diverse communication tools are used. Face-to-face communication is also highly encouraged and used. Simultaneously, all necessary oral communication should be backed up with a written communication using the diverse communication tools available. Also, team members reveal high adaptability towards each other's preferred form of communication. In addition, LebDevCo1 was able to overcome geographical

distances to achieve collaboration. MarketingMan-LebDevCo3 explained: *We do not make an effort to communicate things, it just works great*”.

Furthermore, the usage of virtual Kanban boards was agreed upon by all participants in both DevelopCo and HealthCo, but its application was lacking in LebDevCo1. Participants in DevelopCo and HealthCo highlighted the benefits of virtual Kanban boards as means to share knowledge and information while updating user story status without disturbing the flow of work. Knowledge sharing improves the performance of team members and the success of the overall project. This in turns increases the satisfaction of team members and consequently the mutual trust. The relation between knowledge sharing and trust is reciprocal; when one increases the other increases and vice versa [111].

5.2.3 Approach on the Handling of Non-functional Requirements

Our study shows that there lies a difference in the way companies deal with non-functional requirements. HealthCo and studied sites in Lebanon use a mix of traditional and agile methods to deal with non-functional requirements. While, DevelopCo incorporated the non-functional requirements into the user stories in the agile process.

There is a major difference in the way the studied research sites handle non-functional requirements. HealthCo and Lebanese studied companies mix agile methods with traditional methods in order to handle non-functional requirements. This confuses the self organizing agile teams. For instance, in LebDevCo1, the product owner used a plan based approach to handle the non-functional requirements. This caused confusion and undermined the responsibility of the self-organizing team members. Whereas DevelopCo took the non-functional requirements and embedded them to the agile process. These requirements are turned into user stories and then agile tasks and added to the physical scrum board.

Perceiving the Option of Agile Tailoring

Agile tailoring is different among studied companies. DevelopCo performs little agile tailoring since they are aiming at becoming 100% agile. Practitioners tailor agile methods in the transition state towards achieving a pure agile development process. Contrarily, HealthCo and Lebanese studied companies apply agile along with traditional techniques such as Waterfall. Practitioners pointed out that they assume roles that are usually not present in agile software development such as product manager, line manager, project manager, sub segment head, test manager, team master...Practitioners revealed that they had to do lots of documentation in order to comply with international regulations [76], record testing and results, maintain quality, and comply with design specifications. Practitioners performed quality assurance through constant validation of products with developers and customers.

5.2.4 Customer Involvement

The importance of the customer's involvement is highlighted in the agile principles. Here rises the need to consider the customer as one of the main actors in agile software development and to allocate a lane to represent the customer. All research sites stressed on the importance of customer involvement and the understanding of the customer's requirements and embedding them into the product backlog.

The customer's involvement in the process is extensive in both HealthCo and DevelopCo. Yet, in the studied research sites in Lebanon the customer's involvement is minimum. This is true for the Lebanese customers. Restating what the ClientEng-LebDevCo1 expressed: *"The availability and commitment of their [clients in Lebanon] staff are not always as promised."* Practitioners in Lebanon explained that dealing with Western clients was easier and more collaboration was present among the different stakeholders. Dealing with Lebanese clients, on the other hand, disrupt the agile process. For instance, customer rarely attend the demo; they expect that the demo is only done upon the delivery of the final product.

5.3 Model Creation for Agile Processes in Lebanon

By carefully analysing the data collected in Lebanon and the findings from the cross case analysis, a model is created. The design-reality gap lens [21] is used to show the gap between the theoretical design and the actual reality of agile concepts' implementation. Then, a detailed analysis is done to highlight the influences that will be tackled in the model. This is done through creating a conceptual mapping that relates the influences (impediments to the agile process in the Lebanese context) to be tackled with the agile methods used. Finally, the agile model to be implemented in the chosen research site is presented.

5.3.1 Gap between Actual Implementation and Designed Agile

This section observes the difference between the actual (real) agile concepts implemented and the designed agile concepts. Table 5.3 shows the design reality gap evaluation. A detailed description of all 25 artefacts is available in Appendix A.4. This evaluation is performed according to three criteria:

1. Low: successful implementation; stakeholders are aware of the target and performance objective of agile element.
2. Medium: on the way to achieve success; stakeholders partially implement the required agile process to achieve successful criteria.
3. High: failed or no implementation; stakeholders unaware of agile element or unwilling to implement.

The main concern is to start with reducing the gaps that are identified as 'High': scrum master, daily scrum meeting, and retrospectives. Teams do not have a scrum master, who should be present to coordinate between the product owner and the software development teams. Instead, the product owner is compelled to play the role of the product owner and the project manager who uses a command and control

style of management. Also, the daily scrum meeting is not implemented and is replaced by a morning meeting with the CEO that combines all the teams and extends to an undesirably long period of time. In addition, retrospectives are not implemented at all.

Concepts	Design	Criteria	Reality
Roles	Self-Organizing Teams	Medium	Self-Organizing Teams
	Scrum Master	High	Scrum Master
	Product Owner	Low	Product Owner
Artefacts	Program Governance	Low	Program Governance
	Product Artefact	Low	Product Artefact
	Release Artefacts	Low	Release Artefacts
	Sprint Artefacts	Medium	Sprint Artefacts
	Feature Artefacts	Medium	Feature Artefacts
Ceremonies	Sprint Planning Meeting	Medium	Sprint Planning Meeting
	Daily Scrum	High	Daily Scrum
	Demo	Medium	Demo
	Retrospective	High	Retrospective

Table 5.3: Design reality gap

5.3.2 Conceptual Mapping

Influences to the agile implementation in the Lebanese context were presented in Section 4.4.11. Sixteen impediments are identified where six are external and ten are internal. The developed agile model should focus on the impediments faced when implementing agile and aim to overcome them. Conceptual mapping is used to map the influences, that impede the process, in relation to the agile methods used to overcome them. The model will focus on internal impediments since they can be controlled. Out of the ten identified internal impediments, four will be tackled.

To overcome the **lack of agile roles**, The scrum master role will be introduced. Also, the model will aim to reduce any micro-management of teams in order to become self-organizing teams.

To overcome the **lack of agile ceremonies**, morning meetings with the CEO will be replaced by stand-up meetings. In addition, retrospectives will be introduced to the process.

With the introduction of the scrum master's role, micro management will decrease. This will reduce **employees' fear** and advocate an open environment.

All of the above mentioned actions will enhance the Lebanese's **understanding of agile** software development. Workshops will be conducted to increase the understanding.

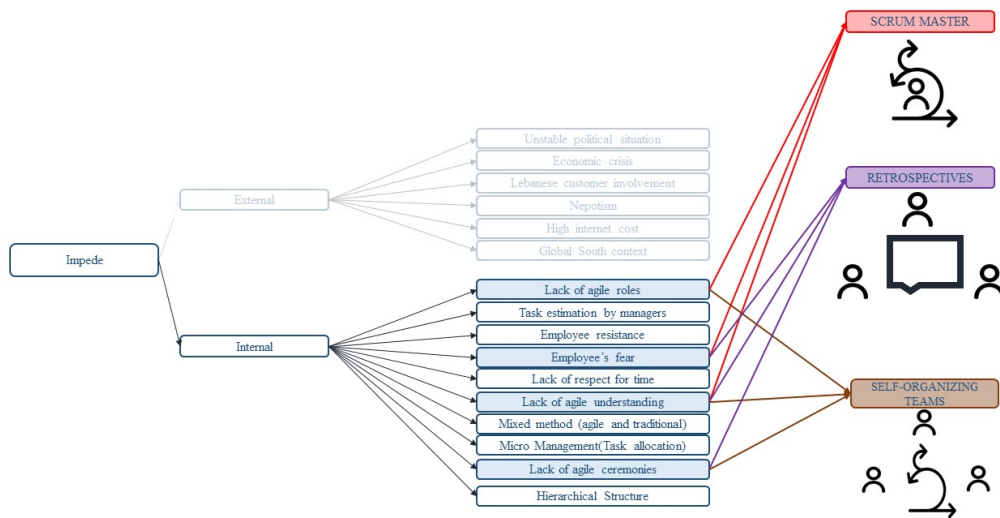


Figure 5.1: Conceptual mapping of impediments faced in relation to the agile methods used

5.3.3 Model for Lebanese SME

Figure 5.2 represents the model created to be implemented in the chosen Lebanese research site. The processes where change is planned are pointed with call-outs indicating the agile concept being implemented. These are introducing the scrum master's role, replacing existing morning meetings with daily scrum meetings, and introducing retrospectives as a closing ceremony for the iteration.

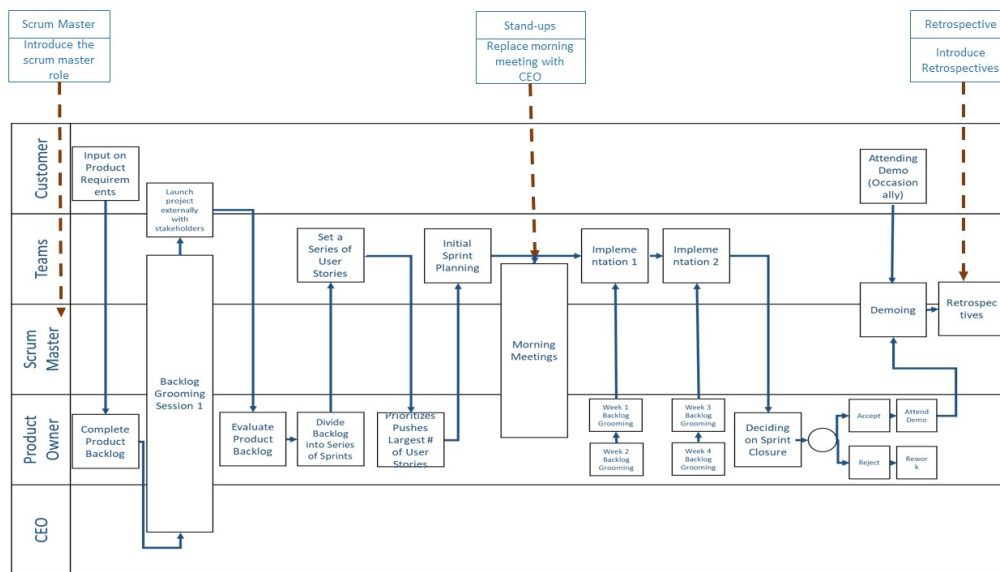


Figure 5.2: Model for agile implementation in Lebanon

5.4 Intervention Findings

This intervention was conducted in LebDevCo1. The motivation behind the intervention process is to apply and validate the model. The intervention is an iterative process starting with the baseline, presented in figure 4.5, then the pre-intervention model, presented in figure 5.2, reaching the final model that will be presented in this chapter in section 5.5. This is presented in figure 5.3.

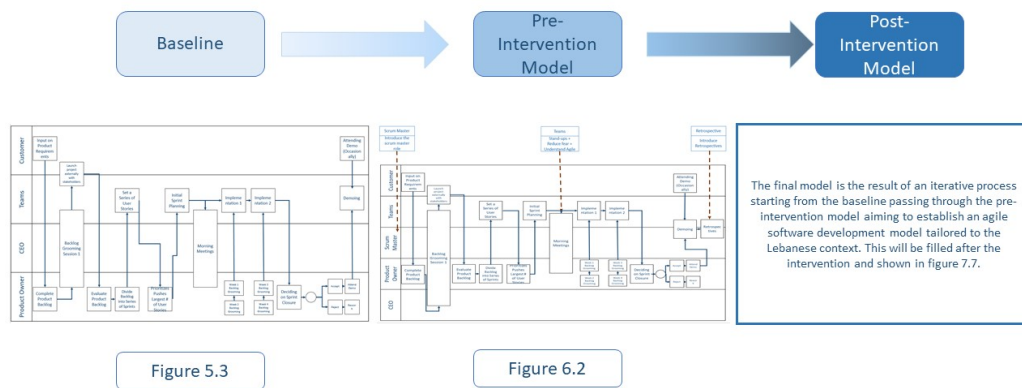


Figure 5.3: Process of final model development

Two teams were chosen to be part of the intervention labelled Team A and Team B. Team A included 6 members and the Product Owner and Team B included 7 members and the Product Owner. A series of three workshops were conducted with both teams, product owner, and managers to explain agile methodology prior to the intervention.

The following sections describe how agile methods were used to overcome the impediments identified. The iterative nature of the process allowed teams to overcome more impediments than the ones identified in the conceptual mapping, Figure 5.1, and even mitigate for impediments caused by external factors which can not be overlooked in reality.

5.4.1 Intervene to Overcome the Lack of Agile Ceremonies

The lack of agile ceremonies was overcome by introducing the daily scrum meetings first and then the retrospectives.

Applying Daily Scrum Meetings

As a simple ceremony to apply in agile methodology, daily scrum meetings, also referred to as stand-up meetings, are implemented first. For a period of 8 weeks, daily scrum meetings were recorded for both participating teams. Total time of the scrum meeting, speaker time of each participant, and time loss were tabulated. Appendix A.5 shows the first and last recordings of the stand-ups for each team. For confidentiality reasons imposed by the participating company, all recordings were deleted by the end of the same day they were recorded.

Several bottlenecks were encountered during the learning process of conducting a stand-up. Team members are not used to put in words what they are doing since the team leader or product owner is the one who speaks during the morning meetings with the CEO. Dev1-LebDevCo1 expressed: *“What am I supposed to say? It is very unnatural for me to describe what I am doing in words”*. After two trials, Dev1-LebDevCo1, DevLeader-LebDevCo1, and Dev2-LebDevCo1 wrote down on a piece of paper the answers for three questions and participated in the process; other team members were encouraged to do the same. Since participants were using papers, the Kanban concept was introduced in an 1 hour and 30 minute workshop. Team members decided to implement the physical Kanban board and use it as a guidance to conduct their stand-up meetings. The engagement of both team members increased.

Team members were sometimes interrupted by each other. Also, on certain occasions, extensive discussions on a particular obstacle took place during the stand-up; especially with participants having little respect of time limits. To confront this obstacle, 2 scrum masters were chosen for each team, after three weeks. Details on

the scrum masters' journey is discussed in section 5.4.2. After several iterations, the stand-ups were a success. Dev4-LebDevCo1 expressed: *"I am starting my day knowing what everyone is doing in details and it is taking much less time than the usual morning meetings."*

Applying Retrospectives

At the end of each iteration, team members of both teams attempted to participate in a retrospective. At the end of the first iteration and after re-explaining what a retrospective is, Team A conducted a retrospective at 3pm and Team B conducted a retrospective at 5pm. Both retrospectives failed to achieve the desired result due to the following obstacles:

- No team member wrote 3 positive and 3 negative sticky notes.
- Very few team members wrote one positive sticky note. DevLeader-LebDevCo1 said: *"Why should I write a positive sticky? The work was done, there is no need to indicate it."*
- Team members resisted this ceremony and did not recognize its benefits.
- Some team members forgot what happened. Dev8-LebDevCo1 explained: *"I forgot if anything wrong happened. We have so much on our mind."*
- Most negative sticky notes were very similar and not diverse. Team members wrote the first thing that came to their minds without thinking deeper. For example, several sticky notes revolved on interruptions during stand-ups.
- Team members were hesitant towards conducting public evaluations. In a side talk, Dev4-LebDevCo1 said: [*"What if I say something wrong and it reaches the managers?"*]
- While grouping the points, many were repetitive and only a couple tackled diverse problems faced.

After several trials to conduct a retrospectives, it was tailored to fit the Lebanese environment. Participants will write their entries anonymously and put in a jar where the scrum master will group them on the wall. The areas of improvement should include constructive criticism that will benefit the team. The retrospectives' main focus is to create a SMART action plan. A Phoenician ship was used to describe each team and three questions were labelled to the different elements of the ship as shown in Figure 5.4. The picture was printed in a large size and stuck to wall.

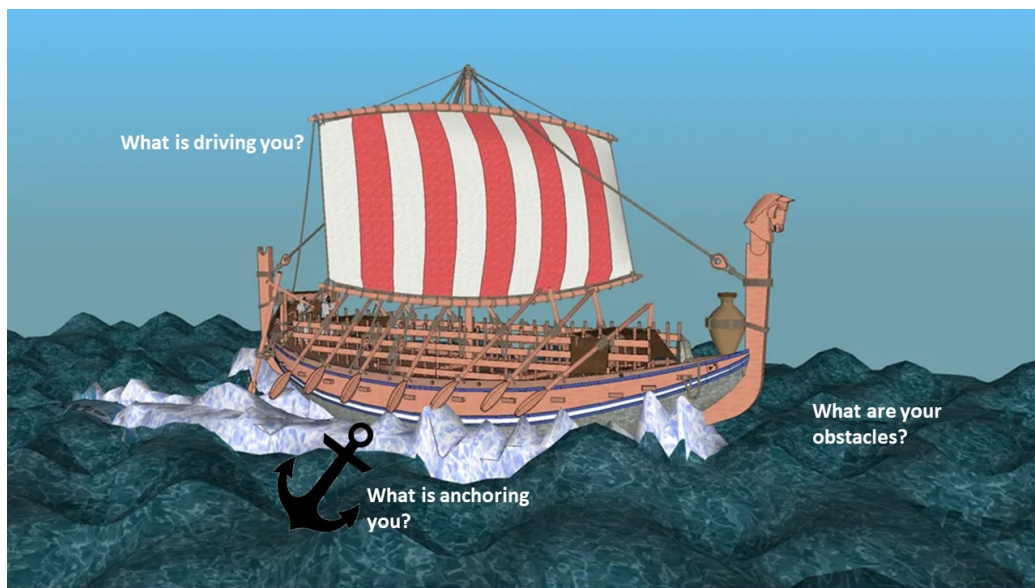


Figure 5.4: Tailored retrospective

Team members are now asked to answer the three questions anonymously described in Table 5.4. With the new format, more engagement was detected among team members. An anchor identified by several team members was customer involvement in the agile process. Dev4-LebDevCo1 claimed in Arabic: *“No matter how hard we try, customer do not collaborate with us fast.”* Thus in order to create an action plan for this, a meeting was decided on Tuesday to find a strategy to increase customer involvement; this is discussed further in Section 5.4.6. With the

improvement observed in the customer involvement after presenting the model, team members got excited and participated even more in the next retrospective.

Question	Minimum Number of answers required	Description
What is driving you?	2	Team members describe elements that contribute to their success
What is anchoring you?	1	Team members indicate the factors that are stopping them from completing their tasks.
What are your obstacles?	1	Team members explain the obstacles they faced during the sprint.

Table 5.4: Three questions in the tailored retrospective

5.4.2 Intervene to Reduce the Lack of Agile Roles

On the third week of the intervention, 2 scrum masters were chosen for each team, DevLeader-LebDevCo1 and Dev2-LebDevCo1 for Team A and Team B respectively to fill the gap. Scrum masters were introduced to their responsibilities in a workshop designed for them. A briefing on the responsibilities was also given to team members to increase collaboration. The scrum master was assigned the following responsibilities as per [17]. The scrum of scrum facilitator responsibility was assigned to the head of development.

First, to evolve as Process Anchors, I conducted with the scrum masters a series of workshops to introduce all the details of the agile process. The workshops I developed were also handed as material for the scrum masters.

Second, as Stand-up Facilitator, the scrum master should conduct the daily stand-up meetings. DevLeader-LebDevCo1 was able to conduct the stand-up smoothly right from the first trial and said: *"I feel like we are discussing things together informally in the stand-up, it is very natural."* While, Dev2-LebDevCo1 consumed some time to hand over turns between team members. Also, Dev2-LebDevCo1 expressed: *"My only concern was that we respect time, which is hard when you are not used to; also not to interrupt."*

Third, as Impediment Remover, the scrum master practiced this role after the third retrospective when they facilitated a meeting designed to think of innovative ways to enhance customer involvement. Refer to Section 5.4.6 for more details.

Forth, as Sprint Planner, the scrum master provided the support needed.

Fifth, the Scrum of Scrum Facilitator responsibility, was assigned to the head of development. This was essential as to not cause a complete organizational change and risk failure.

Sixth, as Integration Anchor, both scrum masters practice it simultaneously. A tailoring to this activity is that the scrum masters will have to report to the head of development and the product owner at the beginning and the end of such activities. This type of reporting is necessary since the organizational structure is hierarchical. A balance was found between this structure and the agile methodology.

In addition, a seventh responsibility was added. During the third iteration, the CEO expressed how he feels disconnected from both teams since they are not attending the morning meetings. He suggested attending the stand-ups. In an effort to reduce micro-management, an additional activity was added to the scrum master's responsibility: Reporter. The scrum master will report the details of the stand-up to the CEO upon his request. The scrum master acted as a reporter for an average of two times per week in the coming iterations.

5.4.3 Intervene to Overcome Task Estimation and Allocation by Managers

With the goal of increasing the self-organizing nature of teams, team members learn how to estimate and prioritize the tasks. To do so, team members learned the difference between estimating an epic and estimating a user story. Figure 5.5 explains the difference in the level of details and the techniques used. To estimate an epic, team members used the T-shirt sizing technique. This involves the labelling of epic with XS, S, M, L, XL, or XXL. It is an informal technique used to estimate larger requirements. This was used by team members in an informal way. PO2-LebDevCo1 Explained:

These estimation techniques are much more accurate than the ones we were doing. They took a bit longer but at least the result is accurate and the team members are required to commit since they estimated.”

Then, to estimate and plan the user stories, planning poker or pooking technique is used. Each estimator is given a set of cards labelled with numbers from the modified Fibonacci sequence. The product owner then reads the user story and answers any questions or clarifications requested from team members. Then each team member will pull out their chosen card but does not show until everyone is ready. If everyone has the same number then this estimation is done. If there are differences, they are discussed, especially the outliers. Finally, a re-estimation is conducted until consensus is reached. Dev8-LebDevCo1 said: *“I feel more involved in the process. I have a say and opinion on the things that I am doing.”*

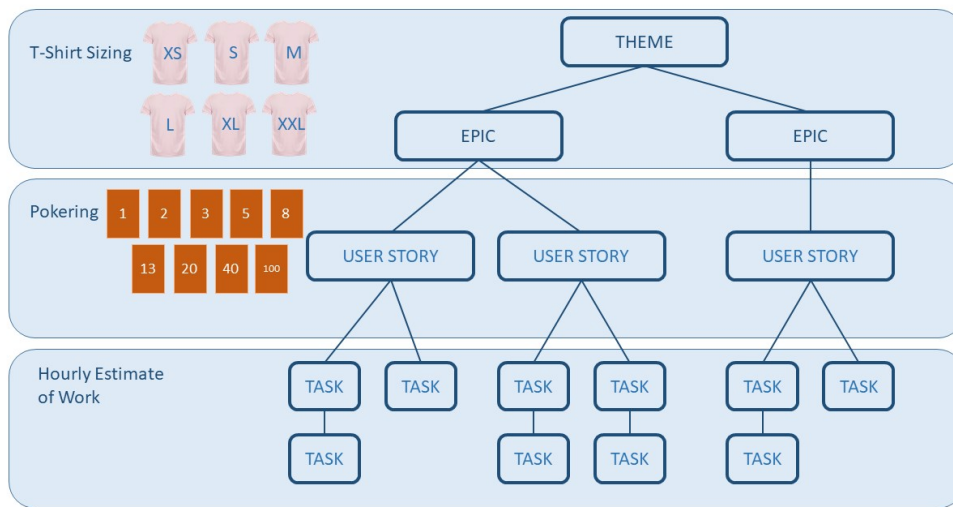


Figure 5.5: Agile estimation techniques

Team members were told to take into consideration the amount of work, complexity, risk, and uncertainty when estimating. These factors are combined together to give the estimate in order to complete product backlog item.

Mitigation of Unstable Political Situation

Due to an unstable political situation, risk and uncertainty in Lebanon should be emphasized more than other locations. First, Lebanon is facing one of the worst economic situations (for example, this limited the ability of all account holders to withdraw money which affected the salary payments). Lebanon is also facing an unsettling political situation (for example, road closures due to protests are very common).

To compensate this, a new label is added to each task. This label's main purpose is to indicate the importance of this task under uncertain crisis, political, economic, or

other. The tasks are labelled green, orange, and red by increasing level of priority. If a crisis started, team members will start working on tasks labelled red, then orange, then green. Figure 5.6 shows the risk mitigation tagging system.

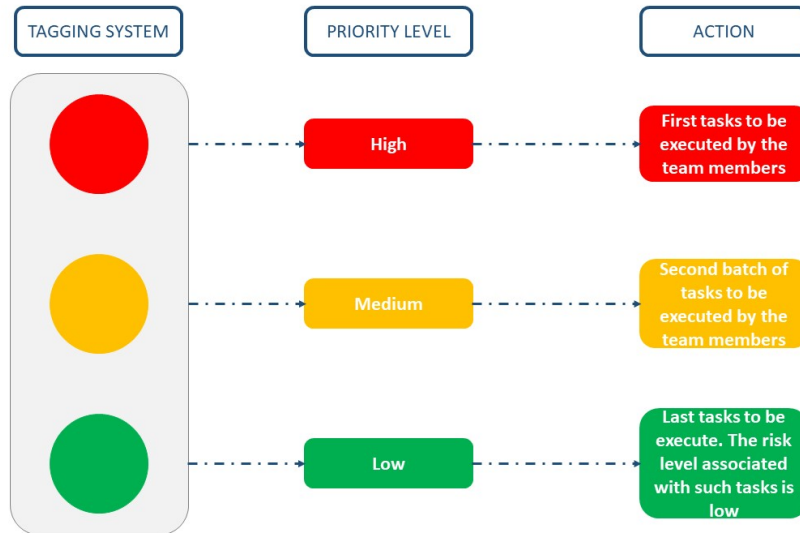


Figure 5.6: Risk mitigation tagging technique

This was put to test on August 4th, 2020. A devastating explosion occurred in Beirut which lead to the glass chartering of the Mount Lebanon offices. Employees had to work from home for 4 days under stressful unprecedented situations. Thus, employees worked on the tasks labelled red to decrease the risk of any late delivery. Consequently, team members completed tasks according to the level of priority. The work was structured and no late deliveries were detected. PO2-LebDevCo1 expressed:

“You cannot imagine the amount of glass that was broken and the chaos that occurred. The next day, we all stayed home. But we had to work and deliver. I told everyone to start working on the tasks labelled red from the technique given by you [the researcher]. Even with everything going on, we managed to deliver without being late.”

5.4.4 Intervene to Reduce Employee's Fear

Employee's fear is a factor affected by the current economic situation and the fear of losing their job. This fear can not be eliminated but through the implementation of retrospectives and the introduction of the scrum master role, this decreased. The constant fear is replaced by the trust factor among team members. Dev4-LebDevCo1 expressed the following when first implementing the retrospectives: *"What if I say something wrong and it reaches the managers?"*. During the post implementation interviews, Dev4-LebDevCo1 expressed: *"I know now that retrospectives are a safe zone. I can say what went wrong for the purpose of improving and not for it to go directly to the CEO."*

5.4.5 Intervene to Enhance Agile Understanding

Throughout the intervention, team members were in the process of learning about agile principle. There was no specific incident but rather an accumulation of learning events. Team members lack knowledge and agile coaching. Thus, a series of workshops were conducted with the chosen teams. Three workshops were conducted initially to allow team members to learn the agile principles and concepts. Team members learned that there is more to agile development than coding and testing. There are other practices some of which team members already mastered such as group programming and effective communication. As for the new practices, team members learned the different concepts. Dev3-LebDevCo1 expressed: *"To be honest, in the first interview, I did not even understand most of the agile methods you talked about. Now I am able to explain to other employees in the company."*

5.4.6 Intervene to Improve Lebanese Customer Involvement

Customer involvement has been identified as a ‘anchor’ during the retrospective as well as the interviews conducted, highlighted in Section 4.4.6. In order to enhance customer collaboration, first team members need to identify to what extend their customer collaborates with them. Thus, a scoring sheet was developed to identify the level of collaboration and group the customers into three groups. Let x be the sum of the scores for a specific customer.

1. If $11 < x < 17$ then the customer is labeled ‘Lebanese Traditional Customer’
2. If $18 < x < 28$ then the customer is labeled ‘Lebanese Connected Customer’
3. If $29 < x < 33$ then the customer is labeled ‘Lebanese Collaborative Customer’

Team members will fill in the scoring sheet by evaluating their relation with the customer according to specific evaluation criteria. If this criteria is weak, it is labeled 1. If this criteria is moderate, it is labeled 2. If this criteria is strong, it is labeled 3. Table 5.5 shows the list of evaluation criteria. Dev9-LebDevCo1 indicated: “*When we are labeling each customer, we will know what to expect.*”

Criteria	Score
Joins the initial estimation of the Product Backlog	
Supports Agile contracting	
Contributes to the creation of the product vision and backlog	
Attends the Sprint Review	
Respects team stability	
Available to answer questions	
Commit to time constraints	
Ensures an available customer representative	
Clarifies requirements when needed	
Provides feedback when needed	
Exchange mutual trust with team members	

Table 5.5: Customer collaboration evaluation criteria

Figure 5.7 places the different classification of customers according to their level of collaboration with the teams in a spectrum. The positive extreme indicates a collaborative customer, for instance the Western customers as identified by practitioners. While the negative extreme indicates a plan-based and traditional oriented customer.

After identifying the level of collaboration with the customer, the team follows the steps designed to increase customer collaboration incrementally as shown in Figure 5.8. Both teams identified their current customer as ‘Lebanese Traditional Customer’. Thus, the process starts with phase one.

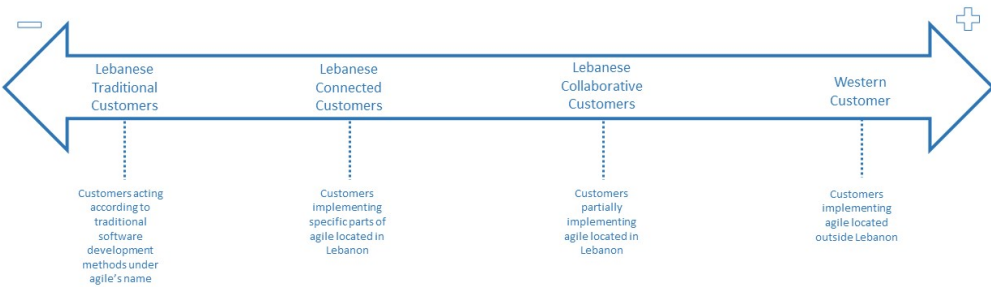


Figure 5.7: Customer collaboration spectrum

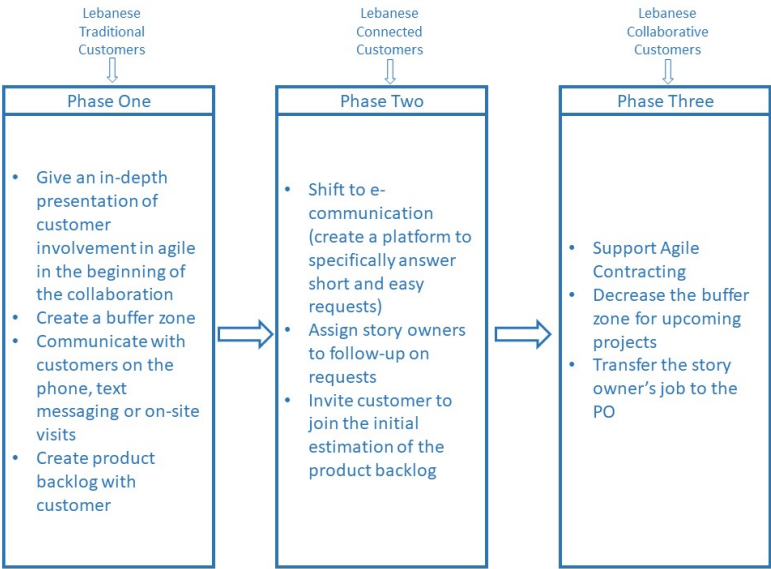


Figure 5.8: Phases followed to increase customer collaboration

5.4.7 Actions to Mitigate the Implementation of Agile in the Global South Context

Being in the Global South context, all throughout the implementation, a conflicting trade-off was explored between implementing agile as-is and tailoring of agile development methods. On one hand, agile methods originate in the Global North. Hence, agile methods represent a northern idea being propagated and adopted in the Global South. On the other hand, self-organizing teams are empowering and emancipatory. Agile is about taking power away from managers and putting more responsibility into the hands of self-organizing software development teams. DataModeler-LebDevCo1 stated:

“When we first started with agile I felt that this is something that does not fit me. But now this has changed. For example, retrospectives fit our own needs and are structured effectively”.

Agile tailoring in different cultural settings from where it originated needs to expand its corpus of basic frameworks and consider the cultural setting in which it is implemented. For instance, in order to fortify the self-organizing teams stand-ups replaced the morning meetings with the CEO. Thus, in order to keep the CEO involved but without interfering, the scrum master took the role of the reporter and informed the CEO about the updates. Dev7-LebDevCo1 stated: *“The scrum master would inform the CEO about the updates. We gained time everyday because we did not participate in the long morning meetings.”*

5.5 Agile Model Tailored to the Lebanese Context

After careful analysis of the intervention, the agile model tailored to the Lebanese context is updated. New processes and call-outs were added. The new model is presented in Figure 5.9. The colours are used as a visual aid to relate the call-outs with the respective activities. The call-out description includes a title, brief

description, and a representative picture when available. The model also includes Lebanese flags on certain activities or roles. This indicates that the activity or role was tailored to fit the Lebanese context. The model will be further discussed in Chapter 6.

The model is tailored to the context of respecting hierarchy advocated by the upper management, yet at the same time allowing for the growth of self-organizing team. For instance, this is portrayed in the daily scrum meetings where the self-organizing teams will conduct stand-ups instead of morning meetings with the CEO. Then, the scrum master's role is tailored to an additional role of Reporter, where he/she will report the subjects of the stand-up to the CEO.

The model is tailored to fit the Lebanese context. An observation is noted when the model for agile software development in Lebanon is compared with the models representing agile software development for HealthCo, Figure 4.2, and DevelopCo, Figure 4.1. Highlights from both diagrams can be observed.

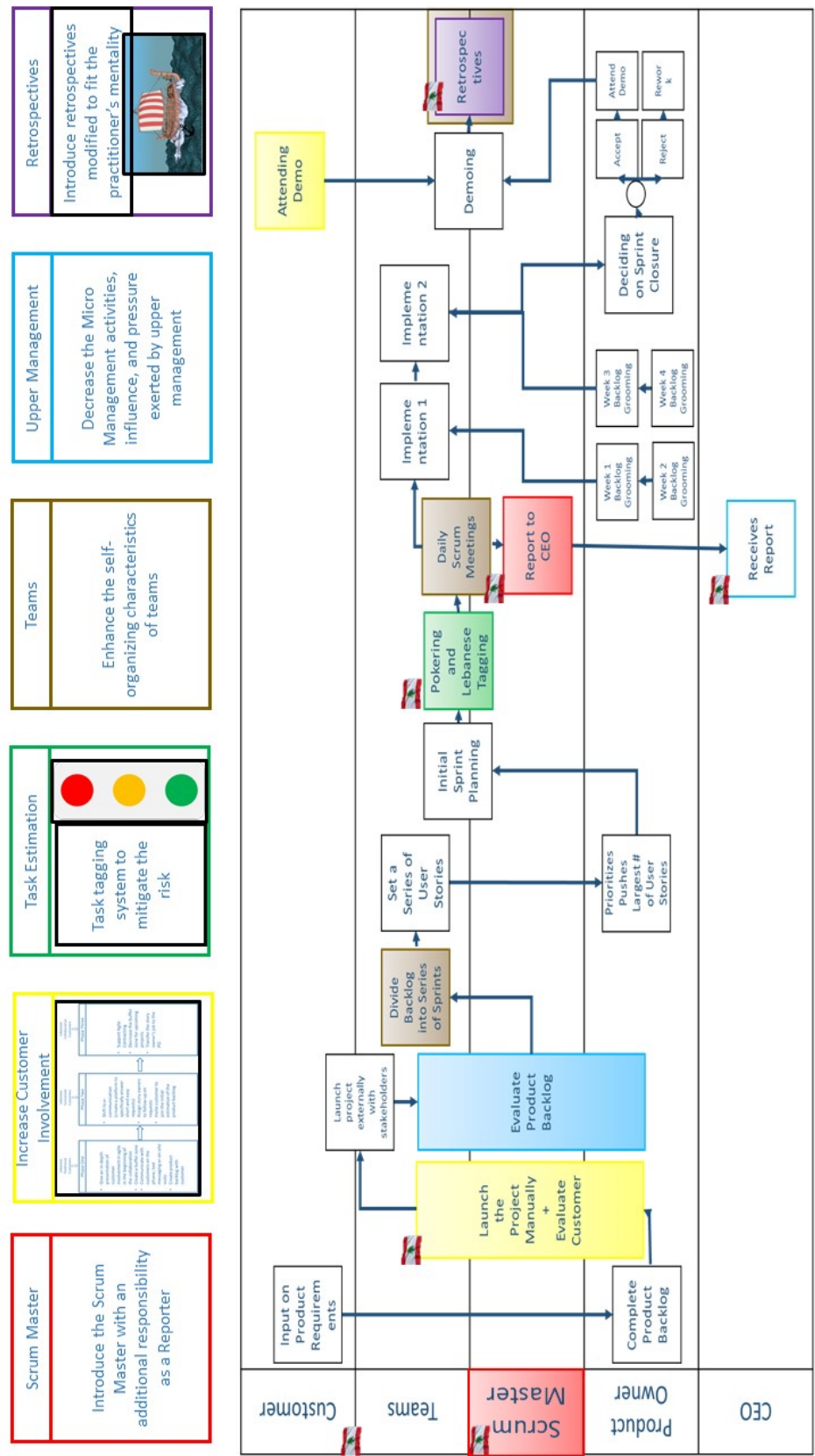


Figure 5.9: Agile model tailored to the Lebanese context

5.6 Post-Implementation Evaluation and Analysis

This section aims at evaluating and validating the change from implementing the chosen agile methods. The improvement in productivity is highlighted. Post implementation interviews are conducted with all of the team members participating. The interviews lasted between 20 to 40 minutes and were recorded, transcribed, coded and memoed according to the same procedure explained in Section 3.7. Also, a post implementation meeting was conducted. It included the members of both teams, the product owner, and three manager. Quotes from both the meeting and the interviews are used to validate the data presented. The conceptual map is also updated to introduce the actual benefits gained.

During the meeting, the design reality gap lens was revisited and new scoring was given to each element. Furthermore, the inhibitors and exciter of agile implementation, presented in Section 4.4.10, are revisited and updated.

As a result of the intervention, 7 of the 12 factors identified for improvement by practitioners were tackled (review Table 4.2). These include: minimize bureaucracy, minimize command chain, CEO's micro management, task assignment, cooperation with client, appreciate the work done, and training.

5.6.1 Conceptual Map: Post-Model Implementation

With the iterative intervention, the impediments covered increased. Internal impediments were tackled and external impediments were mitigated for. Figure 5.10 shows the conceptual mapping relating the impediments with the respective agile methods used. Colours are used to ease visualization (same colours as the ones used for the model). Dotted lines indicate risk mitigation for the external factors.

The following explains each agile method and the impediments it overcame or mitigated for:

1. Daily Scrum Meetings replaced the morning meetings with the CEO along with the introduction of the Kanban boards. The introduction of the new ceremony decreased hierarchical structure, decreased micro-management, and increased agile understanding.
2. Tailored retrospectives enhanced agile understanding, especially agile principles. It also decreased the fear of employees and helped build trust.
3. The introduction of the scrum master, with additional tailored responsibilities, reduced employee's fear and enhanced the team's self-organizing nature.
4. The lack of customer involvement was mitigated with a series of tailored actions to increase the Lebanese customer's involvement.
5. Estimation techniques decreased the hierarchical structure and micro management. These techniques also mitigated risks of unstable political situation.
6. All through the intervention, the practitioner's understanding of agile increased. The Global South context was taken into consideration when tailoring the agile methods.

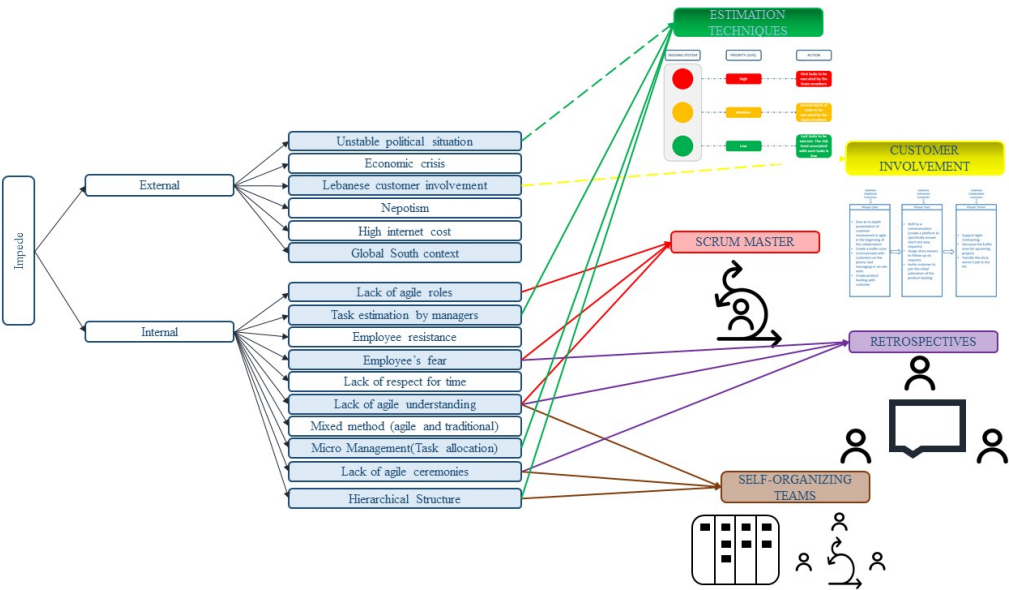


Figure 5.10: Conceptual mapping post-intervention

5.6.2 Gap between Actual Implementation and Designed Agile: Post-Model Implementation

During the meeting held with all stakeholders involved in the implementation process, the design-reality gap was revisited. The stakeholders followed the same procedure described in Section 5.3.1. The below table represents the post implementation design-reality gap model.

Concepts	Design	Criteria	Reality
Roles	Self-Organizing Teams	Low	Self-Organizing Teams
	Scrum Master	Medium	Scrum Master
	Product Owner	Low	Product Owner
Artefacts	Program Governance	Low	Program Governance
	Product Artefact	Low	Product Artefact
	Release Artefacts	Low	Release Artefacts
	Sprint Artefacts	Medium	Sprint Artefacts
	Feature Artefacts	Medium	Feature Artefacts
Ceremonies	Sprint Planning Meeting	Low	Sprint Planning Meeting
	Daily Scrum	Low	Daily Scrum
	Demo	Medium	Demo
	Retrospective	Low	Retrospective

Table 5.6: Design reality gap model post-implementation

Significant changes are observed in the design reality gap lens pre and post implementation. The blue coloured cells highlight the changes between pre-implementation shown in Table 5.3 and post-implementation shown in Table 5.6. To begin with, self-organizing teams' gap example, was labelled medium pre-implementation and then advanced to low post-implementation. Dev5-LebDevCo1 explained: *"We are now estimating tasks using T-shirt sizing technique. I feel much more involved in the process"*. PO2-LebDevCo1 said: *"As a product owner, teams members can no longer tell me I have no time. They are responsible to finish their tasks because they participated in the estimation. In this way they become more productive"*.

The scrum master's role was introduced to the process. Due to time limits, the assigned scrum masters still need more practice to apprehend all the scrum master's responsibilities. Thus, the scoring was reduced from high to medium. Dev9-

LebDevCo1 explained: *“Dev2-LebDevCo1 [name redacted] is facilitating our communication with each other during stand-ups and with PO2-LebDevCo1 [name redacted] throughout the day”*. Dev2-LebDevCo1 expressed: *“I have taken a lot of the responsibilities of the scrum master, but I still have more. I need to start facilitating the sprint planning.”*

The sprint planning meetings’ gap was reduced from medium to low. The team members are now effectively participating in the sprint planning and are no longer spectators. Although improving the sprint planning was not a direct goal, the improvement was highlighted during the meetings with the stakeholder. BuisDelHead-LebDevCo1 claimed:

“I thought our sprint planning was great. But when we started getting input from the developers, it improved a lot more. So I attended one of them and saw the T-shirt sizing technique being implemented and realized that the developers expand much better the tasks necessary to complete the product backlog.”

The daily scrum meetings were a success after several iterations. The team members highlighted extensively the benefits. Dev6-LebDevCo1 said: *“The daily scrums are much more specific and oriented towards our daily tasks.”* In addition, Dev2-LebDevCo1 explained: *“We gained time with the daily meetings. We are more productive in the meeting and the work to be done is clearer”*. Table 5.7 shows the number of hours gained after shifting from the morning meetings with the CEO to the daily scrum meetings in the last iteration. The table shows a comparison between the recordings of each team’s daily scrum and the morning meetings with the CEO. The time of the morning meetings with the CEO was taken from the Minutes of Meetings sent by email each day. In one sprint, the total number of hours gained from conducting the stand-ups rather than the morning meetings is 38 hours. In addition, Figure 5.11 shows how the teams progress in conducting daily scrum meetings.

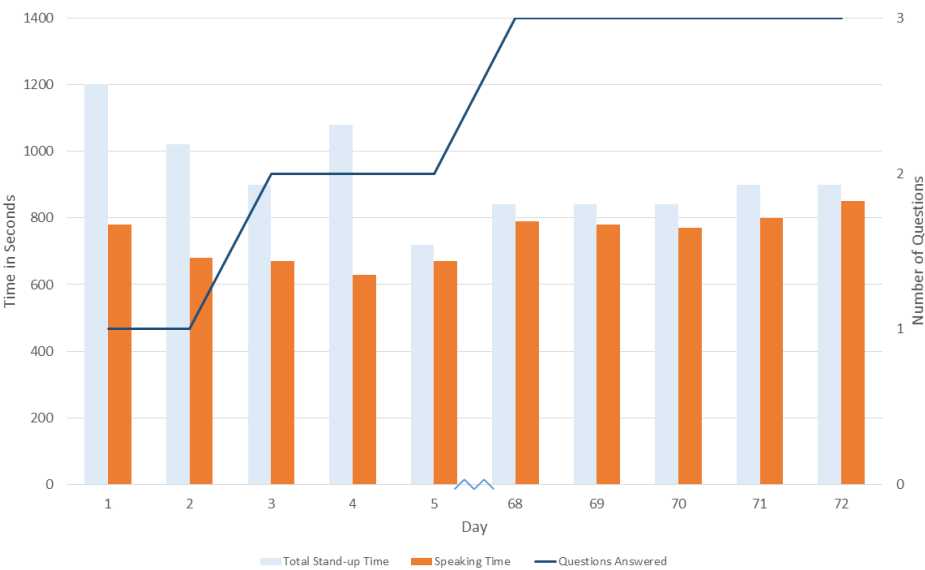


Figure 5.11: Difference observed during Daily Scrum Meetings between the first and the last week of the intervention

Day	Standup Total Time (min) Post-Intervention		Morning Meeting Total Time (min) Pre-Intervention	Time Gain (min)	
	Team A	Team B		Team A	Team B
1	13	17	32	19	15
2	13	12	35	22	23
3	13	11	17	4	6
4	14	15	32	18	17
5	13	14	23	10	9
6	13	14	38	25	24
7	15	14	36	21	22
8	15	14	31	16	17
9	13	15	40	27	25
10	13	15	23	10	8
Cumulative Time Gain (min)				172	166
Number of Team memebers in each team				6	7
Total Time Gain (hours)				18	20

Table 5.7: Time gain upon stand-up implementation

Retrospectives’ gap was reduced from high to low. Retrospectives were tailored to fit the Lebanese context. DataModeler2-LebDevCo1 explained: “At first I was against retrospectives because I though they were centred around evaluating our

own work. But then when we started putting action plans and creating solutions, like the one for customer involvement, I was more convinced.

5.6.3 Inhibitors and Exciter of Agile Implementation: Post-Model Implementation

The number of inhibitors decreased and the number of exciters increased. Figure 5.12 represents the updated TIC model after the intervention. From ICT to Capability Approach, three exciters are identified, sprint planning retrospectives and daily Scrum meetings, the two latter were previous inhibitors. From Capability Approach to ICT, nothing has changed. Although the employee resistance decreased with the involvement of employees in the sprint planning and agile estimation and allocation, yet employee resistance is not totally eliminated.

Both dimensions to and from ICT and Institutional Theory did not change. Although agile method implementation increased yet there are certain processes and activities that use a traditional software development methods.

From Institutional Theory to Capability Approach the previous inhibitors are now exciters and represented by more productive self-organizing teams. Also the will to implement agile is now replaced by implementation of agile methods. From Capability Approach to Institutional Theory, there are no more inhibitors since stakeholders were introduced to all agile principles fully and started implementing new agile methods.

A major benefit detected is the elimination of vicious circles created by inhibitors. Post implementation, not one vicious circle is reported according to the new model.

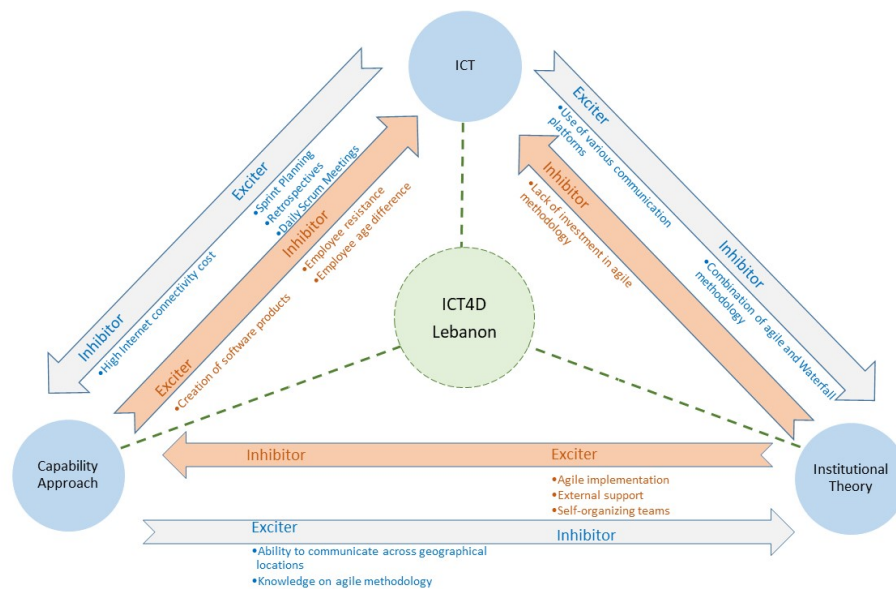


Figure 5.12: TIC model post-implementation

5.7 Summary

This chapter presents the findings from the cross-case analysis. It introduces the model to be implemented in the chosen Lebanese SME. It also explains the intervention that took place at the chosen Lebanese company. The model produced on agile implementation and tailoring to the Lebanese industry is presented along with a post implementation evaluation.

Chapter 6

Discussion

6.1 Introduction

This chapter answers the research questions and analyses, and interprets the findings in the light of related literature. It presents the contributions in relation to theory and practice as well as the limitations of this research.

6.2 Answering the Research Questions

This research aims to improve empowerment perceived by selected practitioners using agile software development methods in Lebanese small and medium-sized enterprises (SMEs). To achieve this aim, the three research questions presented in Section 1.3 are answered.

Research Question 1 How do selected practitioners from software companies describe the different approaches to agile tailoring of roles, ceremonies, and artefacts?

Careful analysis of the findings from this preliminary study reveals two different approaches when implementing agile methods [132]. The first is represented by a pure agile path adopted by DevelopCo; the second is a mixed method approach between agile and traditional methods adopted by HealthCo. Mixing agile and traditional methods is adapted to the content specific situation of companies [163]. As mentioned in Chapter 2, Ken Schwaber, the co-founder of Scrum, opposes any implementation of agile methods beyond its entire components [153]. However, this does not exclude the success of projects that implement hybrid methods, as shown in HealthCo. For example, in HealthCo, new roles are added to the original three roles to manage the non-functional requirements and the regulated environment. Numerous artefacts are also created due to the extensive documentation and reports required, as well as additional ceremonies to track the changing regulations. These roles, artefacts, and ceremonies were integrated along with traditional plan-based methods.

On the other hand, the implementation of agile methods in its entirety has shown success in DevelopCo [133] and through a large number of case studies mentioned in the literature review. The critical success factors identified in reference [46] are visible in DevelopCo, such as the inter-team communication strategy and the understanding of agile methods. DevelopCo managed non-functional requirements by incorporating all activities into the agile process and conventional roles, artefacts, and ceremonies.

DevelopCo demonstrates how any required action such as documentation, regulatory compliance, and safety critical work items can be implemented and embedded in an agile software development process. By using cross-case analysis and extensive studies of previous literature, this thesis stands with the complete implementation of agile methodology as means to develop software. Also, this study advocates the tailoring of agile methods rather than using a mixed method approach of agile and traditional methods.

By careful and systematic analysis of our data and extensive study of the previous literature, Tables 6.1 and 6.2 illustrate how any action such as documentation and

safety critical work items can be implemented in an agile process. These tables demonstrate how any activity may be completed using agile methodology. Two new artefacts are introduced (agile tailoring): Documentation Work Item and Safety Critical Work Item.

Table 6.1 shows an illustrative example of how practitioners can deal with documentation work items using agile methods. The table shows the recommendation detailing the type and description. The documentation user story mentioned may also include documentation change requests. Practitioners should be selective in their documentation practices whether it is for design documentation to help in creating future designs or imposed documentation in regulated environments.

Table 6.2 shows an illustrative example of how practitioners can deal with safety critical requirement work items using agile methodology. The table shows the recommendations detailing the type and description. There are certain regulations and non-functional requirements that are applicable across the entire development process. Such regulations may be included in the team's definition of Done which may be considered as consistent set of accepted criteria that applies to all backlog items.

#	Type	Artefact/ Ceremony	Description
1	User Story	Artefact	Product Owner creates user story for documentation purposes.
2	Definition of Done	Artefact	Product Owner creates a new definition of Done for the documentation purposes.
3	Product Backlog	Artefact	The documentation user story is added to the product backlog.
4	Product Backlog	Artefact	The Product Owner prioritizes the user stories in the backlog including any documentation user stories.
5	Product Backlog	Artefact	Identify the high level product backlog for the documentation.
6	Sprint Planning	Ceremony	Self-Organizing teams will then estimate the user stories including any documentation user stories.
7	Sprint Planning	Ceremony	Select the highest priority user story including any documentation user story for the next sprint.
8	Documentation Work Item	Artefact	Self-organizing teams will work on the documentation user story until it achieves the definition of Done.
9	Demo	Ceremony	Demo the work item (documentation) and the PO decides if the work is of acceptable quality.

Table 6.1: Illustrative example of using agile methods to deal with documentation work items

#	Type	Artefact/ Ceremony	Description
1	User Story	Artefact	Product Owner creates user stories for any safety critical requirements.
2	Definition of Done	Artefact	Product Owner creates a new definition of Done for the safety critical requirements (Since the code is subject to more stringent quality control processes)
3	Product Backlog	Artefact	The safety critical requirement user story is added to the product backlog
4	Product Backlog	Artefact	The Product Owner Prioritizes the user stories in the backlog including any safety critical requirement user stories.
5	Product Backlog	Artefact	Identifying the high level product backlog for the safety critical requirements.
6	Sprint Planning	Ceremony	Self-Organizing teams will then estimate the user stories including any safety critical requirements user story.
7	Sprint Planning	Ceremony	Select the highest priority user story including any safety critical requirement user stories for next sprint
8	Safety critical Work Item	Artefact	Self-organizing teams will work on the safety critical requirement user story until it achieves the definition of Done.
9	Demo	Ceremony	Demo the work item (safety critical requirement) and the PO decides if the work is of acceptable quality.

Table 6.2: Illustrative example of using agile methods to deal with safety critical work items

Research Question 2 How do selected practitioners describe the current implementation of agile in Lebanese SMEs, and how does this compare with the practitioners investigated in Research Question 1?

From the practitioners' perception, a list of influences that characterize the current state of agile in the studied Lebanese companies is identified. These are divided into influences that impede or amplify agile implementation. As mentioned in Chapter 4, amplifiers are influencers that support, enhance, or reinforce agile implementation; while impediments are influencers that obstruct, hinder, or prevent agile implementation. These are then sorted into external factors, characterized by an uncontrollable nature, and internal factors controlled by the company's practitioners. A total of sixteen impediments were identified that inhibit the agile implementation process and six factors that amplify the process. These influences are also portrayed as inhibitors and excitors in the TIC model, which interlinks institutional theory, the capability approach, and ICT (information and communication technology) [135].

One of the agile tailoring debates is explored when comparing DevelopCo, who applied 'pure' agile methodology, and HealthCo, who mixed agile methods with the plan based methods. The first company managed non-functional requirements as artefacts in their agile methodology, while the other company reverted to conventional plan-based software development practices of documentation, timeline estimations, and safety critical requirements. When deciding on the method used to develop software, practitioners should carefully consider the relationship between the critical success factors for the project and the method used, whether plan-based traditional method or agile method[?]. When exploring the non-functional acquirements in this study, the critical success factors were to deliver the required safety critical work items and documentation work items while adhering the evolving international regulation. Since these regulations are changing, the plan-based method characterised by linear planning falls short to achieve the desired critical success factors.

Of the sixteen impediments, six are external impediments to the agile process

implementation. Two of these are novel and specific to the Lebanese context (indicated by an asterisk in the table): economic crisis and nepotism. Table 6.3 explains the impediments and relates, when available, to existing literature.

External Impediments	Description
Effects of unstable political situation	Unstable political situation in Lebanon affected the involvement of the software development industry and obstructed daily activities on a regular basis [2].
Economic crisis*	The recent economic situation exerted high pressure on employers, worrying for their company's survival, and employees, striving to keep their job.
Lebanese customer involvement	With the lack of customer involvement comes a set of challenges in agile [31]. Lebanese clients showed little involvement in the development process.
Nepotism*	Nepotism in Lebanon exerted pressure on employers to give a job to family members requesting it, even if they do not deserve it. This hindered productivity and discouraged other employees.
High internet cost	The high internet connectivity cost and poor quality obstructed daily activities.
Global South context	Theories from the Global North cannot be copy pasted into the Global South context [102, 140].

Table 6.3: External influences that impede agile implementation in the Lebanese context

Of the sixteen impediments, ten are internal impediment to the agile process implementation. The ten internal influences exert impediments on the implementation of the agile process. Five of these are novel and specific to the Lebanese context (indicated by an asterisk in the table): extensive micro management by the CEO and

upper managers, the reasons behind the lack of agile ceremonies, task estimation done by managers in agile, employee's fear, and lack of respect to time. Table 6.4 explains the impediments and relates, when available, to existing literature.

Internal Impediments	Description
Lack of agile understanding	The lack of knowledge on agile methodology leads to incorrect implementation of agile concepts. Very few personnel possessed agile adoption and implementation knowledge in Lebanon [52].
Mixed method (agile and traditional)	Agile methods are mixed with traditional methods, such as Waterfall, to fill in the gaps. These gaps may be filled using agile methodologies.
Micro management (Task allocated by managers)*	Upper managers and CEO extremely micro managed which increases pressure on employees and limited the team's ability to become self-organizing [90]. Tasks were allocated by managers, and team members were not given the opportunity to express their points of interest. Managers limited the self-organizing team's responsibility.
The reasons behind the lack of agile ceremonies *	Two major agile ceremonies, stand-ups and retrospectives, are not implemented, thus reducing agile benefits and effectiveness. The main reason for not implementing retrospectives is fear of public self-evaluation and for not implementing stand-ups is micro management.
Lack of agile roles	Lack of scrum master limits agile spirit [17]; and lack of self-organizing teams reduces speed and agility [91].
Task estimation by managers in agile*	Task estimation is performed by managers who distance team members from the planning processes.
Employee resistance	Lebanese employees not only lacked the knowledge on agile; they resisted what they don't know.
Employee's fear*	The economic crisis increased the employees' fear of losing their job, thus hindering the implementation of agile concepts such as retrospectives and self-organizing teams.
Hierarchical Structure	The hierarchical structure limits the practitioners ability to implement the basic agile principles. It is difficult to shift from the 'command-and-control management to leadership-and-collaboration' [117]
Lack of respect to time*	Many Lebanese practitioners do not respect time limits and due dates, which made agile implementation hard.

Table 6.4: Internal influences that impede agile implementation in the Lebanese

In addition, six amplifiers were identified that enhance the agile implementation process; two are external factors and four are internal factors. Three of these are novel and specific to the Lebanese context (indicated by an asterisk in the table): information sharing, overcoming geographical distances, and the need to succeed. Table 6.5 explains the amplifiers and relates, when available, to existing literature.

Amplify Agile Implementation		
External	External support (NGOs)	Non-governmental organizations provided support for the rising software development industry [168].
	Outsourcing software to Lebanon	Countries mainly in the Gulf and MENA region were outsourcing software to Lebanon which increased the demand for in-house process improvement [96].
Internal	Information sharing*	Practitioners used diverse communication means to deliver the message and share information effectively, thus overcoming the communication challenges associated with agile [113].
	Overcoming geographical distance*	Practitioners overcame difficulties of communication between geographically distributed team members.
	Sprint Planning	Sprint planning ceremony was performed as part of the planning for each iteration.
	Need to succeed*	Lebanese companies have no other option but to succeed in order to survive the political and economic situation.

Table 6.5: Influences that amplify agile implementation in the Lebanese context

With scarcity of literature on agile software development in developing countries, especially in Africa and the MENA region, this section also reveals the practitioners' perception on agile and compares it with similar contexts.

This study presents a concept that agile software development needs to expand its corpus of basic frameworks (Global North) to study other context (Global South). Agile implementation, and in turn agile tailoring, should consider the cultural setting in which it is implemented. Understanding the practitioner's perception by placing the study in its right Global South context adds a new perspective. The implementation of agile in the Global South varies from the implementation in the Global North [134].

Similarly to Egypt [115], knowledge on agile methods is limited to cursory research. This lack of knowledge results in partial implementation of agile roles, artefacts, and ceremonies, which in turn produces challenges in agile implementation [117]. It is very difficult to find personnel who are knowledgeable on agile in Lebanon. For instance, the role of the scrum master is neglected. Also, certain ceremonies are not implemented, such as stand-ups and retrospectives. To fill the gaps, the upper management mixes agile with traditional techniques such as the Waterfall or Spiral techniques. These methods usually involve plan-based techniques which restrict agile potential in serving the customer and allowing for changes later in the development process.

This lack of knowledge leads to the ignorance of a major agile principle: 'individuals and interactions over processes and tools' [26]. The organizational structure is lead by strict hierarchy. Knowledge sharing and decision making are guided by a set of strict rules and principles passing through the upper managers, including the CEO. This limits the ability of team members to form self-organizing teams, which lies in the core of agile methodology [90]. In addition, the upper management constantly demands reports (documentation) which also hinders the process and creates new unnecessary artefacts.

A large share of software development SMEs in Lebanon are family-owned business [71]. They cannot be internationally publicly traded because of financial rules and regulations in Lebanon [120]. This allows for nepotism to manifests in the company in various forms, which demotivates the team members and limits their creativity.

With the ever unstable political situation and the economic crisis in Lebanon, employees have a fear of loosing their jobs. This affects the agile ceremonies, especially the retrospectives, since employees fear public self-evaluation. They believe this will affect their position in the company. Also, due to the strict hierarchical structure, employees may hinder knowledge sharing, when mistakes occur, fearing it may affect their job. This in turn hurdled trust building, which is essential for the agile process [164, 111].

Moreover, cultural factors are intertwined with complex issues affecting the software development teams [34]. The Lebanese culture, including the social ties, perspective of time, and the national identity, affects the implementation of agile software development [134]. For instance, stakeholders tend to be vague about specifying a certain time for task completion. Also, CEOs tend to micro manage the processes.

Research Question 3 What are the selected practitioners' perceptions of the effect of tailoring agile methods, in the Lebanese SMEs, on improved empowerment?

With the application of agile methods in the selected research sites, some agile methods were adopted as-is and others were tailored. Practitioners tailored agile methods incrementally to fit the Lebanese context, especially when impediments were faced. The tailoring of agile ceremonies and roles improved the productivity in the practitioners' teams. Challenges were confronted using the agile thinking taking into consideration the Lebanese context: Lebanese culture, CEO (upper management) mentality, employees' fears, current political situation, and economic crisis.

The practitioners' capabilities were fortified with the application of tailored agile methods. This is shown in the difference between the baseline TIC model, figure 4.3, and the post -intervention TIC model, figure 5.12. Positive effects were detected, affecting the capabilities approach in the TIC model, which in our study represents the capabilities of practitioners implementing agile. The changes flowing from the institution theory to the capabilities approach enhanced the productivity of

self-organizing teams. Also, the changes flowing from ICT to capabilities approach enhanced the communication and planning through the ceremonies conducted.

As a consequence of applying the model in Figure 5.2, it was enhanced and refined as shown in Figure 5.9. The model started from the baseline shown in Figure 4.5. Then, agile concepts were introduced to create the new model shown in Figure 5.2. Throughout the intervention, several iterations were conducted to reach the final version presented in Figure 5.9.

The highlights of this model are described below:

1. The CEO is no longer considered a main actor in the process and is now replaced by the scrum master whose role was not present earlier. (Refer to Section 5.4.2)
2. The customer evaluation is conducted in order to execute the correct action plan. Practitioners tailored this process to classify the customers according to the level of engagement. (Refer to Section 5.4.6)
3. A task tagging system is created to mitigate the risk towards the unstable political situation in Lebanon. This tagging system was tested when an explosion, third largest non-nuclear explosion in the world, hit Beirut and it was a success. (Refer to Section 5.4.3)
4. An additional responsibility is assigned to the scrum master: Reporter. (Refer to Section 5.4.2)
5. Retrospectives are tailored to fit the Lebanese context in terms of questions and ceremony structure. (Refer to Section 5.4.1)
6. Teams increased their level of productivity and self-organization.
7. The morning meetings are replaced with the Daily Scrum Meetings thus increasing the members' productivity. (Refer to Section 5.4.1)

The research around agile tailoring is predicated on the assumption that methods must be adapted to the local context. Conventionally, agile method tailoring tends to focus on factors such as business domains, technology stack and project characteristics (such as project scope or size and disposition of teams) [40]. The approach here, in contrast with conventional agile method tailoring, has been to address agile tailoring in relation specifically to the Lebanese context. So, the perspective in this research is developed to combine two aspects: the changes in Lebanese software development practices needed to implement agile, and adaptations to agile methods required to fit the Lebanese context. Agile development needs to expand its corpus of basic frameworks. Agile tailoring should consider the cultural setting in which it is implemented. With the implementation of agile methods or tailored agile methods, practitioners were able to increase their productivity.

6.3 Contribution to Theory

This thesis contributes to the studies of agile methodology using an interrelationship between the Theory of Explaining and Theory of Design and Action [82]. More specifically, the Theory of Explaining feeds into the Theory of Design and Action. The Theory of Explaining is developed to clarify things that were “poorly or imperfectly understood beforehand” [82]. This thesis presents a Theory of Explanation on the nature of agile methodology in the Lebanese industry and links the human factor and practice to technology development. The Theory of Design and Action uses contextualized general expert knowledge to develop a proposed design. This thesis presents a Theory of Design and Action through developing and validating an agile model designed to fit the Lebanese context shown in Figure 6.1. The means of representation is the swim lane diagram showing one iteration.

The results and outcomes of this research study can be seen as making a first step in filling the literature gap on agile software development in Lebanon. To the best of the researcher’s knowledge, and after an in-depth search of all recent PhD research and literature available focusing on the agile software development, there is no

study that identifies the influences. However, this research is unique, as it is the foundation of future work on the subject in developing countries such as Lebanon.

Agile tailoring is not a new concept, but it has predominantly focused on project characteristics in the past, such as large scale agile, regulated environments, and organizational structure. The Global South and Global North perspective presents a new and original framing on the tailoring concept, by placing agile methods in a wider Global South context. On one hand, agile methods represent an idea from the Global North being propagated to the Global South. On the other hand, the agile concept of self-organising teams has the potential to be empowering and reduce hierarchical management models. This approach aids in understanding phenomena of agile tailoring, where development process ceremonies are adapted to suite a specific local context.

This research produces a unique set of findings based on the mixed method research, which highlights the baseline and the challenges faced upon agile software development in Lebanon. The study not only focuses on internal company factors that affect agile implementation, but looks into the external factors.

The TIC model was used as a means to assess the practice of agile methods in the organizational context. The TIC model is used to represent the inhibitors and exciters affecting agile implementation in the studied context in the baseline state. Then, the TIC model is used again to represent the same, but after the implementation of certain agile methods tailored to the context. The difference between the two models drawn at different instances is then compared by observing the changes in the inhibitors, exciters, virtuous circles, and viscous circles.

This study uses the swim lane diagram as a means to represent the processes and information flows in agile software development. The idea of using swim lane diagrams came from previous experience and knowledge in industrial engineering. The visual display of roles, responsibilities, and sequence of processes allows the identification of bottlenecks, and in turn perform continuous improvement. This aligns with the agile concept and allows for the agile process under study to be revised after each iteration. As a result, several versions of the swim lane

diagram were created. First, the baseline representing the current state is shown in figure 4.5. Then, a proposed model is shown in figure 5.2, and presented to the practitioners in the studied organization. Then, after each iteration, a swim lane diagram was drawn to track the progress. The final model, shown in figure 5.9, represents the designed agile process containing callouts which highlight the agile methods implemented and tailored to the context.

The study also uses the swim lane diagram to develop a model tailored to the context shown in Figure 6.1. The model is tailored to the Lebanese context, taking into consideration the 22 influences of agile implementation in Lebanon. Sixteen impediments were identified out of which seven are specific to agile in Lebanon: extensive micro management by the CEO and upper managers, the reasons behind the lacking of agile ceremonies, task estimation done by managers in agile, employee's fear, lack of respect to time, economic crisis, and nepotism. In addition, six amplifiers were identified that enhance the agile implementation process with three being specific to agile in Lebanon: information sharing, overcoming geographical distances, and the need to succeed.

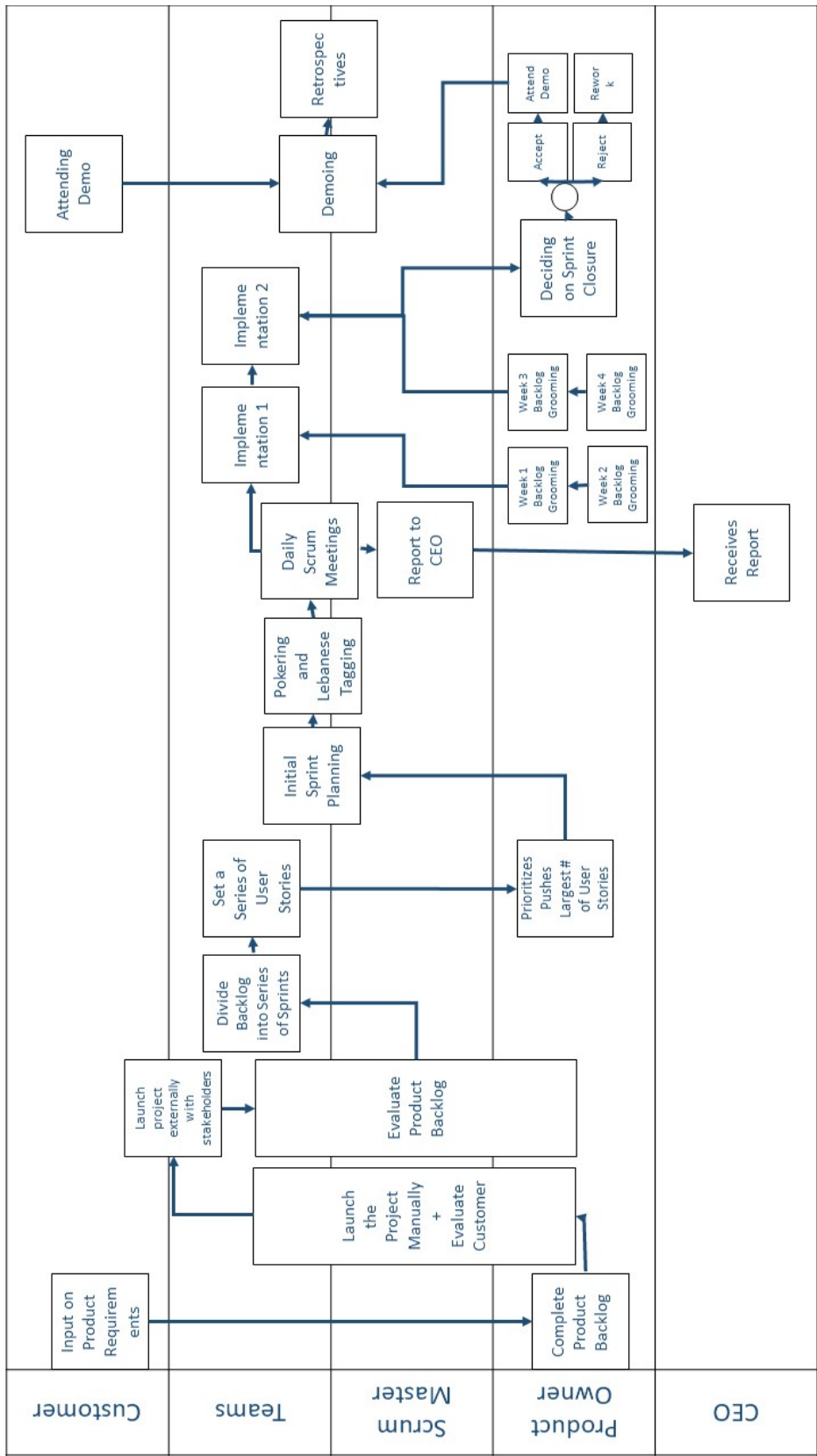


Figure 6.1: Agile model tailored to the Lebanese context

6.4 Contribution to Practice

The developed model has implications for practitioners. The following section presents the implications of this model developed based on findings, and backed-up with previous literature, for senior managers, self-organizing teams, product owners, scrum masters, agile coaches, and even customers.

For agile software development **senior managers** and company owners in Lebanon, this study produces a model that is useful to achieve an agile environment. Senior managers should decide, be aware, and be convinced that their environment needs to be agile [36]. An important aspect of this awareness is understanding agile methods, their role in creating an environment to achieve and sustain self-organizing teams, trust between team members (to decrease fear), less time wasted, agility, free flow of information, and openness [111]. Also, software development companies may consider offering relevant training to their new customers. Practitioners may begin by implementing the model shown in Figure 6.1. Based on the findings of this study, teams should start by filling in the design reality gap and draw their own swim lane diagram. Then, the goal would be to decrease the gap.

The transition of becoming a **self-organizing team** is challenging [93], especially when managers tend to micro manage and team members have fear of losing their job, as the case in Lebanon reveals. The roles and practices described in this thesis and previous literature, such as in [91, 17], help team members understand their roles and practices when becoming a self-organizing agile team. One of the characteristics of self-organizing teams is their ability to react spontaneously in response to challenges. As findings suggest, in order to achieve these characteristics, both managers and team members should work on practising daily scrum meetings, retrospectives, task estimation, and task prioritization. While some teams may easily adjust to the new environment made up of these roles and practices, others

present in a different context may struggle [134], and some may fail to make the transition. Thus, rises the need to have a scrum master to guide the transition towards a self-organizing team as well as additional responsibility to transfer daily progress to management. The description of the intervention in this thesis should help scrum masters understand what to expect in terms of challenges and how to react through the roles and practices.

This thesis aids the understanding of the **scrum master's** role in recognizing the tailored responsibilities for achieving agile software development. In the Lebanese context, the scrum master's role is to be tailored to fit the specific context, so that more responsibilities may be introduced. The micro management of managers adds responsibilities to the scrum master's responsibilities. The scrum master protects the team from any outside disruptions caused by stakeholders that will affect their work flow [17]. Also, the scrum master should always maintain a facilitator position for the team rather than a manager position [119]. Information sharing between the scrum master and the agile coach should always be present to ensure success.

The model developed and described in this thesis aids **agile coaches** to better understand agile implementation in the Lebanese context. An agile coach should evolve from the use of traditional software development project manager practices to agile methods concentrating on guiding the team to becoming a self-organizing team. The identified influences, weather they amplify or impede the process, should help agile coaches know what to expect in terms of inhibitors and excitors. An important application for the agile coach is to first understand the status-quo of the company and then iteratively introduce the agile roles, artefacts, and ceremonies and facilitate their adoption rather than forcing them on all stakeholders.

This thesis aids the understanding of the **product owner's** role. The findings can be used to develop targetted interventions aimed at stepping into the product owner role in agile, rather than the manager role of traditional methods. From our findings, the traditional responsibilities imposed by cultural understandings need to be replaced by agile behaviours. Although communication is one of the

identified amplifiers, this research corroborates previous research advocating that product owners need to enhance face-to-face communication and decrease the formal documents and emails sent under the notion of hierarchy. Product owners need to be focused on specific goals and target the stakeholders in doing the same. The product owner should also network with the scrum master on how to support the self-organizing teams [18].

Customers may be new to the agile software development methods. Organizations need to inform the customers about these methods in order to achieve the desired levels of involvement. Customers may consider carefully selecting members from within their organization as representatives to collaborate with the development teams. These representatives should be responsive and have a minimum level of knowledge on agile principles. The representative should be provided with enough time and authority to effectively collaborate with the team.

6.5 Limitations

The research limitations are discussed in this section using Lincoln and Guba's approach. This approach is characterized by a naturalistic approach [105] through evaluating the trustworthiness of the findings: credibility, transferability, dependability, and conformability.

Credibility defines the level of compatibility between the respondents' opinions and the researcher's interpretation, as well as their presentation [105]. The study conducts an in-depth insight into agile implementation in the context of developing countries. Data was first collected through semi-structured open-ended interviews. The interview guide was revised after conducting the preliminary study. Data was then collected after obtaining the organization and practitioner's consent. The interviews were unsupervised. The findings were based on the practitioners' quotes. This was informed by the Grounded Theory and cross-case analysis. However, limited detailed description of the work was provided because of the confidentiality

agreement with the research sites.

Transferability studies the applicability of the results from one context to another, while understanding the circumstances that affect the studied context [105]. Diverse research sites present the findings with stronger assurance that implications for practice may be applicable from one context to the other. This context may be limited to the developing countries (Global South context). The results may be transferable to similar contexts in the Global South in countries like Jordan, Syria, Egypt... Nevertheless, the unit of analysis in this study are the practitioners and their perception of agile implementation and not the research sites. The five research site provide good insight for an exploratory study such as this.

Dependability studies the ability to replicate the research [105]. Data was collected from multiple respondents in six different sites. However, the companies under study are spread across four geographical locations: The Netherlands, Lebanon, India, and Kenya, making replication of the precise circumstances hard to achieve. Replication assumes that the precise circumstances faced by a particular group can be reproduced exactly, which is not the case. Generalization is not the aim in this study; Grounded Theory does not claim to produce a universally acceptable theory [42].

Incorrect data is a validity threat, since the data collected from interviews may be prone to be bias [125]. However, conducting semi-structured open-ended interviews allows the researcher to let the participants provide detailed data and examples.

Conformability examines the researcher's objectivity in relation to the studied context [105]. Grounded Theory was employed (coding, constant comparison, memoing, etc). Case study notes were limited to those taken during interviews and observations conducted in LebDevCo1. Findings depend on the practitioner's perception. All data, including research and practitioner interviews' audio and transcribed data were treated anonymously to eliminate any research bias.

The study's qualitative methodological choice was derived from its aim and consequently the research questions. Qualitative research has provided the necessary

context for the study, allowed the exploration of variables, and developed themes necessary to study under-represented communities of software developers [54]. It also allowed the understanding of the practitioners' perception through the open ended interviews. These gave deep insight on the fieldwork descriptions of activities, interactions, communication, and detailed observation [126]. Whereas the quantitative methodology focuses on the number or percentage of practitioners behaving in a certain way, qualitative methodology concentrates on the practitioners' perception aiming to understand the reasons behind the behaviour.

Chapter 7

Conclusion

7.1 Summary of Thesis

In recent years, there has been a significant shift from traditional software development methods to agile methods [158]. With the increase in outsourcing of software to developing countries, such as Lebanon, the need to become more agile is increasing. Lebanon has been the primary source of outsourcing software for the MENA region [152]. The market is estimated to be worth 436.2 million US dollars in 2016, with more than 200 companies involved in the IT sector [96]. In light of recent events (economic crisis, COVID19, and political unrest), the software development industry has grown even faster, surpassing the expected estimated value of year 2020 (543.3 million) [96]. Practitioners facing challenges find little or no guidance (or research) on implementing agile that takes into consideration the Lebanese context.

Practitioners are adopting the agile methodology, yet their limited knowledge leads to improper implementation. This in turn affects the productivity of the software development teams. Low productivity influences the level of agility, customer involvement, and self-organizing teams [6]. With the growth in software development

outsourcing to Lebanon, productivity improvement becomes a necessity.

The thesis presents a literature review on agile software development. The corresponding chapter presents diverse ways of assessing agile implementation through evaluating agile concepts, challenges, tailoring, and customer involvement. The reviewed literature showed that research on agile implementation in Lebanon and its impact is under-researched, and there is a need for in-depth studies.

This research aims to improve empowerment perceived by selected practitioners using agile software development methods in Lebanese small and medium-sized enterprises (SMEs). Thus, three research questions were developed to achieve this aim: **Research Question 1** How do selected practitioners from software companies describe the different approaches to agile tailoring of roles, ceremonies, and artefacts? **Research Question 2** How do selected practitioners describe the current implementation of agile in Lebanese SMEs, and how does this compare with the practitioners investigated in Research Question 1? **Research Question 3** What are the selected practitioners' perceptions of the effect of tailoring agile methods, in the Lebanese SMEs, on improved empowerment?

The study follows a mixed method qualitative research. The research design chapter explains the methodological issues applied in this research. The rationale for adopting a qualitative approach with a pragmatic viewpoint was critically analysed. The chapter describes the research approach and introduces the four main phases and their applied methodologies. The first phase describes the data collection strategy which involves interviews and observations. The second phase involves the data analysis which starts with a within case analysis using an approach informed by Grounded Theory. The data collected involves five research sites and 49 practitioners. The third phase is divided into two parts: Part A involves a cross-case analysis comparing the findings from the all research sites; Part B involves the model creation. The last phase is the intervention study conducted at the chosen research site. It is used to analyse and validate the created model. At the end of this phase, the final model is presented with an evaluation based on post-intervention interviews. The model takes into consideration the agile tailoring to the Lebanese

SME context.

The findings from the two research sites, HealthCo and DevelopCo, are presented in the preliminary study. The different approaches taken by the two research sites are characterized by either an agile-driven approach or a mixed agile and traditional methods approach.

In addition, the findings and an in-depth analysis of agile software development in the chosen Lebanese SMEs is presented. The diverse topics that characterize agile in this study's context are described. The current status of agile implementation is presented, revealing the challenges faced. In addition, a list of the influences in Lebanon that either impede or amplify agile implementation is presented.

Then, a cross-case analysis is conducted and a model is created. The cross-case comparison was conducted between the Lebanese companies, DevelopCo, and HealthCo. The cross-case analysis involved manually overlapping the swim lane diagrams previously presented to compare and contrast the interaction and information flows. The findings reveal a comparison between the different agile concepts. Also, the communication means preferred by practitioners were compared along with the approach to non-functional requirements. The level of customer involvement in each site was also compared. Finally, after careful examination of the findings, the model for agile implementation in the Lebanese context is created.

The final phase involves the intervention at the chosen research site and the post-implementation evaluation and analysis. The final model is then refined and fitted to the Lebanese context highlighting the agile methods tailored to fit exactly the Lebanese context.

The discussion reveals the answers for the three research question. The first research question is answered by revealing the different approaches to implement agile, pure agile or agile mixed with traditional methods. Also, after careful and systematic analysis of the findings and the literature, illustrative examples are developed to show how agile may be implemented to perform safety critical and documentation work items. The answer to the second research question explains how practitioners

in the chosen SMEs describe the current state of agile and compare it to similar contexts. Twenty-two influences are also identified that impede or amplify agile implementation, out of which 10 are discovered in this study. The answer to research question 3 reveals the final model. Then, the contribution to theory and to practice is presented, along with the limitations of the study.

7.2 Reflection

In this section, I present the reflection on the main stages of research, which include identifying the aims and objectives, developing the methodology, collecting the data, and analysing the data.

The preliminary stage of the study was challenging yet exciting. The most challenging part was identifying the main aim and designing objectives and research questions that aid in achieving this aim. This process is iterative and developed continuously with the progression of the study. Developing the methodology was a process that stimulated creativity in combining different methods. For future researchers embarking on a PhD study, it is highly important to get deeply acquainted with the different methodologies before choosing their path.

Data collection was conducted under unprecedented circumstances. While collecting data from the three Lebanese research sites, Lebanon was facing an economic crisis, political turmoil, and constant road closures. Thus, it was a daily challenge to reach the research sites. In order to ensure the successful data collection, I used to park my car in the nearby town and walk to the research site to reach there on time to conduct interviews.

In addition, to ensure the authenticity of the collected data, semi-structured interviews were conducted. The same question was asked in multiple ways at different instances in the interview to ensure objectivity. Putting the participants at ease (especially considering the difficult situation everyone was facing) during the interview is essential to ensure data quality and depth.

In the process of data analysis, it is highly important to keep track of all the data through organizing it and writing titles that would summarize the context. Organization is key to make sure no data is lost (it is also a time saver). In addition, field notes provide an important source of data; it is not stressed on enough in the literature. Data collection and analysis was fortified by observations conducted at the chosen research site.

Unfortunately, on August 4th, a major explosion occurred in Beirut causing an immense blast (reaching Mount Lebanon). The explosion resulted in 204 deaths, 6500 injuries, and 300,000 people homeless. At that time I was in my write-up period. I had to pause for a period before gaining back momentum and continuing the process.

The writing process was an incremental process starting with publications. I aimed to have conference and journal publications prior to the start of the write-up phase. This method provides a head start for researchers in the write-up stage. On the other hand, researchers should know that rearranging the sections of the publication and adding the required information to each chapter is a time consuming process. Even with publications, the researcher should allocate the required time for the write-up process. It is a way to prepare the researcher to defend the thesis.

7.3 Contribution Summary

The thesis presents a tailored agile software development model that takes into consideration the Global South context, specifically Lebanon. This research begins to fill the literature gap identified in agile software development in Lebanon. The aim of this research is to improve empowerment perceived by selected practitioners using agile software development methods in Lebanese SMEs. To reach the aim of this research, the relative questions mentioned in section 1.3 were answered.

Research Question 1 How do selected practitioners from software companies describe the different approaches to agile tailoring of roles, ceremonies, and

artefacts

This question sought to investigate the different approaches to implementing agile in the investigated research sites. They are characterized by either a pure implementation of agile methods or mixing agile with traditional methods [132, 133]. The research advocates the use of pure agile or tailored agile approach rather than a mixed approach between agile and traditional methods. The research concludes that all activities may be executed using an agile approach. Even non-functional requirements, safety critical work items, and documentation may be performed using agile techniques. Tailoring may be applied to reach this aim. Two new artefacts are introduced (agile tailoring): Documentation Work Item and Safety Critical Work Item. This is accompanied by an illustrative example detailing the application methods explaining the artefacts and ceremonies in each step. The illustrative examples contain a step-by-step process, starting from the creation of the user story by the product owner to the demoing of the documentation or safety critical work item.

Research Question 2 How do selected practitioners describe the current implementation of agile in Lebanese SMEs, and how does this compare with the practitioners investigated in Research Question 1?

After the careful examination of the data collected from the three Lebanese research sites, the analysis revealed common influences that characterize agile implementation [135]. Six amplifiers were identified that enhance the agile implementation process; two are external factors and four are internal factors. Three of these are novel and specific to the Lebanese context: information sharing, overcoming geographical distances, and the need to succeed. Sixteen impediments were identified. Six are external impediments, out of which two are novel and specific to the Lebanese context: the effects of the economic crisis and nepotism on daily routines. Ten are internal impediments, out of which five are novel and specific to the Lebanese context: extensive micro management by the CEO and upper managers, the reasons behind the lack of agile ceremonies, task estimation done by managers in agile, employee's fear, and lack of respect to time. Identifying these influences is a significant step in developing the tailored agile model designed to improve the software development team's involvement.

Research Question 3 What are the selected practitioners' perceptions of the effect of tailoring agile methods in the Lebanese SMEs on improved empowerment?

After careful examination of the literature and findings of this study (pre and post intervention), a model revealing agile software implementation tailored to the Lebanese context is presented. The intervention enabled the implementation of agile methods either as they are or through tailoring these methods to fit the context. The model was validated by applying it in the Lebanese research site, which improved the practitioners' involvement in the process and in turn empowering them. It aided in enhancing agile implementation [134]. These include interventions to overcome task estimation and allocation by managers, reduce employee's fear, enhance agile understanding, enhance collaboration through introducing the previously lacking agile roles and ceremonies, actions to improve Lebanese customer involvement, and actions to mitigate the implementation of agile in the Global South context. All these improved the engagement of teams in the agile software development within the studied research site. Agile tailoring should take the cultural environment into consideration.

7.4 Future Work

Outsourcing software to Lebanon is increasing. With the devaluation of the currency, outsourcing to Lebanon is cheaper. Thus, more research is needed on the software development in Lebanon.

With the organizational structure of Lebanese companies, roles required to form and support self-organizing teams could be studied. More information on self-organizing roles would help us to establish a greater degree of accuracy on productivity improvement. In addition, strategies to increase customer collaboration in the Lebanese context could be further studied.

Moreover, a cross case analysis is currently being performed between two studies

conducted by a group of three researchers, including myself. The first is a longitudinal embedded case study where data was collected from one multinational organisation over a period of 21 months. The second is a collective responsive case study where data was collected from 6 research sites across 4 countries over a period of 5 months. This study will aid researchers to make a decision about which approach is feasible to consider for the research study.

References

- [1] a. CollabNet and a. Version One. *The 12th Annual State of Agile Report: Long Term Servey*. VersionOne, Atlanta, 2018.
- [2] Pamela Abbott. How can african countries advance their outsourcing industries: An overview of possible approaches. *The African Journal of Information Systems*, 5(1):2, 2013.
- [3] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, and Juhani Warsta. Agile software development methods: Review and analysis. 2002.
- [4] Steve Adolph, Wendy Hall, and Philippe Kruchten. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4):487–513, 2011. doi: 10.1007/s10664-010-9152-6. ID: TNspringerjour10.1007s1066401091526.
- [5] Piyanshu Aggarwal and Ritika Mool Chandani. Agile methodology influence on sdlc (software development life cycle). *Studies in Indian Place Names*, 40(50):4579–4589, 2020.
- [6] Adeel Ahmed, S Ahmad, N Ehsan, E Mirza, and SZ Sarwar. Agile software development: Impact on productivity and quality. In *2010 IEEE International Conference on Management of Innovation & Technology*, pages 287–291. IEEE, 2010.
- [7] Boluwaji A. Akinnuwesi, Faith-Michael Uzoka, Stephen O. Olabiyisi, Elijah O. Omidiora, and Paula Fiddi. An empirical analysis of end-user participation in software development projects in a developing country context.

- The Electronic Journal of Information Systems in Developing Countries*, 58 (1):1–25, 2013.
- [8] Hassan Alaaraj and Sallahuddin Hassan. Developing swot/tows strategic matrix for e-government in lebanon. *International Journal of Multidisciplinary Research and Development*, 1(7):181–186, 2014.
- [9] Hiva Alahyari, Richard Berntsson Svensson, and Tony Gorschek. A study of value in agile software development organizations, 2017.
- [10] Bahiya M. Aljallabi and Abdelhamid Mansour. Enhancement approach for non-functional requirements analysis in agile environment. In *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)*, pages 428–433. IEEE, 2015.
- [11] Phillip Armour. A little queue theory. *Communications of the ACM*, 58(1): 38–39, 2015. doi: 10.1145/2688493. ID: TNAcm2688493.
- [12] Sohail Asghar and Mahrukh Umar. Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (cots) components. *International Journal of Software Engineering*, 1 (1):32–50, 2010.
- [13] Jeffry Babb, Rashina Hoda, and Jacob Nørbjerg. Embedding reflection and learning into agile software development. *IEEE Software*, 31(4):51–57, 2014.
- [14] BankMed. Analysis of lebanon’s ict sector. Technical report, BankMed, 2014.
- [15] Michael Barrett and Eivor Oborn. Boundary object use in cross-cultural software development teams. *Human Relations*, 63(8):1199–1221, 2010.
- [16] Julian M. Bass. Agile method tailoring in distributed enterprises: Product owner teams. In *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on*, pages 154–163. IEEE, 2013.

- [17] Julian M. Bass. Scrum master activities: process tailoring in large enterprise projects. In *Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference on*, pages 6–15. IEEE, 2014.
- [18] Julian M. Bass. How product owner teams scale agile methods to large distributed enterprises. *Empirical Software Engineering*, 20(6):1525–1557, 2015.
- [19] Julian M. Bass. Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology*, 75:1–16, 2016.
- [20] Julian M. Bass and Andy Haxby. Tailoring product ownership in large-scale agile projects: Managing scale, distance, and governance. *IEEE Software*, 36(2):58–63, 2019.
- [21] Julian M Bass and Richard Heeks. Changing computing curricula in african universities: Evaluating progress and challenges via design-reality gap analysis. *The Electronic Journal of Information Systems in Developing Countries*, 48(1):1–39, 2011.
- [22] Julian M. Bass, Ian K. Allison, and Udayan Banerjee. Agile method tailoring in a cmmi level 5 organization. *Journal of International Technology and Information Management*, 22(4):5, 2013.
- [23] Julian M. Bass, Brian Nicholson, and Eswaran Subhramanian. A framework using institutional analysis and the capability approach in ict4d. *Information Technologies and International Development*, 9(1):pp. 19–35, 2013.
- [24] Kent Beck. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000.
- [25] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development, 2001. URL <http://www.agilemanifesto.org/>.

- [26] Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, et al. Manifesto for agile software development. 2001.
- [27] Dirk Beerbaum. Applying agile methodology to regulatory compliance projects in the financial industry: A case study research. *Journal of Applied Research in the Digital Economy (JADE), Special Issue*, 2, 2019.
- [28] Andrew Begel and Nachiappan Nagappan. Usage and perceptions of agile software development in an industrial context: An exploratory study. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, pages 255–264. IEEE, 2007.
- [29] Woubshet Behutiye, Pertti Karhapää, Dolors Costal, Markku Oivo, and Xavier Franch. Non-functional requirements documentation in agile software development: challenges and solution proposal. In *International Conference on Product-Focused Software Process Improvement*, pages 515–522. Springer, 2017.
- [30] Palash Bera. Does cognitive overload matter in understanding bpmn models? *Journal of Computer Information Systems*, 52(4):59–69, 2012.
- [31] Hugh Beyer, Karen Holtzblatt, and Lisa Baker. An agile customer-centered method: rapid contextual design. In *Conference on extreme programming and agile methods*, pages 50–59. Springer, 2004.
- [32] Edwin Blake and William Tucker. Socially aware software engineering for the developing world. 2006.
- [33] Alexander Boden, Gabriela Avram, Liam Bannon, and Volker Wulf. Knowledge management in distributed software development teams-does culture matter? In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, pages 18–27. IEEE, 2009.
- [34] Alexander Boden, Gabriela Avram, Liam Bannon, and Volker Wulf. Knowledge management in distributed software development teams-does culture

- matter? In *2009 Fourth IEEE International Conference on Global Software Engineering*, pages 18–27. IEEE, 2009.
- [35] Barry Boehm. A view of 20th and 21st century software engineering. In *Proceedings of the 28th international conference on Software engineering*, pages 12–29, 2006.
- [36] Barry Boehm and Richard Turner. Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, 22(5):30–39, 2005.
- [37] David N Boote and Penny Beile. Scholars before researchers: On the centrality of the dissertation literature review in research preparation. *Educational researcher*, 34(6):3–15, 2005.
- [38] Mohamed Bourimi, Thomas Barth, Joerg M Haake, Bernd Ueberschär, and Dogan Kesdogan. Affine for enforcing earlier consideration of nfrs and human factors when building socio-technical systems following agile methodologies. In *International Conference on Human-Centred Software Engineering*, pages 182–189. Springer, 2010.
- [39] Cristina Rosa Camacho, Sabrina Marczak, and Daniela S Cruzes. Agile team members perceptions on non-functional testing: influencing factors from an empirical study, 2016.
- [40] Amadeu Silveira Campanelli and Fernando Silva Parreiras. Agile methods tailoring—a systematic literature review. *Journal of Systems and Software*, 110:85–100, 2015.
- [41] Oisín Cawley, Xiaofeng Wang, and Ita Richardson. *Lean/agile software development methodologies in regulated environments—state of the art*, pages 31–36. Lean Enterprise Software and Systems. Springer, 2010.
- [42] Kathy Charmaz. *Constructing grounded theory*. sage, 2014.
- [43] Susan E Chase. Still a field in the making. *The Sage*, 2011.

- [44] Thomas Chau and Frank Maurer. *Knowledge sharing in agile software teams*, pages 173–183. Logic versus approximation. Springer, 2004.
- [45] Chantal Chelala and Lina Koleilat Ghalayini. The impact of the lack of data protection regulations on the development of e-commerce in lebanon.
- [46] Tsun Chow and Dac-Buu Cao. A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81(6):961–971, 2008.
- [47] D Jean Clandinin and Jerry Rosiek. Mapping a landscape of narrative inquiry. *Handbook of narrative inquiry: Mapping a methodology*, pages 35–75, 2007.
- [48] Alistair Cockburn. *Agile software development: the cooperative game*. Pearson Education, 2006.
- [49] Alistair Cockburn and Jim Highsmith. Agile software development, the people factor. *Computer*, 34(11):131–133, 2001.
- [50] Jean Comaroff and John L Comaroff. Theory from the south: Or, how euro-america is evolving toward africa. In *Anthropological forum*, volume 22, pages 113–131. Taylor & Francis, 2012.
- [51] Kieran Conboy and Brian Fitzgerald. Method and developer characteristics for effective agile method tailoring: A study of xp expert opinion. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 20(1):2, 2010.
- [52] Kieran Conboy, Sharon Coyle, Xiaofeng Wang, and Minna Pikkariainen. People over process: key people challenges in agile development. *IEEE Software*, 2011.
- [53] Juliet Corbin and Anselm Strauss. Strategies for qualitative data analysis. *Basics of Qualitative Research. Techniques and procedures for developing grounded theory*, 3, 2008.

- [54] John W Creswell, Ron Shope, Vicki L Plano Clark, and Denise O Green. How interpretive qualitative research extends mixed methods research. *Research in the Schools*, 13(1):1–11, 2006.
- [55] J.W. Creswell and J.D. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2017. ISBN 9781506386690. URL <https://books.google.com.lb/books?id=335ZDwAAQBAJ>.
- [56] Daniela S. Cruzes, Tore Dybå, Per Runeson, and Martin Höst. Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering*, 20(6):1634–1665, 2015.
- [57] Donatella Della Porta. 11 comparative analysis: case-oriented versus variable-oriented research. *Approaches and methodologies in the social sciences*, page 198, 2008.
- [58] Torgeir Dingsøy and Nils Brede Moe. Towards principles of large-scale agile development. In *International Conference on Agile Software Development*, pages 1–8. Springer, 2014.
- [59] Torgeir Dingsøy, Tore Dybå, Mette Gjertsen, Anette Odgaard Jacobsen, Tor-Erik Mathisen, Jan Ole Nordfjord, Kjetil Røe, and Kjetil Strand. Key lessons from tailoring agile methods for large-scale software development. *IT Professional*, 21(1):34–41, 2019.
- [60] Torgeir Dingsoyr, Davide Falessi, and Ken Power. Agile development at scale: The next frontier. *IEEE Software*, 36(2):30–38, 2019.
- [61] Torgeir Dingsøy and Nils Brede Moe. Research challenges in large-scale agile software development. *ACM SIGSOFT Software Engineering Notes*, 38(5):38–39, 2013. doi: 10.1145/2507288.2507322. ID: TNacm2507322.
- [62] Torgeir Dingsøy, Nils Moe, Tor Fægri, and Eva Seim. Exploring software development at the very large-scale: a revelatory case study and

- research agenda for agile method adaptation. *Empirical Software Engineering*, 23(1):490–520, 2018. doi: 10.1007/s10664-017-9524-2. ID: TNspringerjour10.1007/s10664-017-9524-2.
- [63] Mary Dixon-Woods, Shona Agarwal, David Jones, Bridget Young, and Alex Sutton. Synthesising qualitative and quantitative evidence: a review of possible methods. *Journal of health services research and policy*, 10(1): 45–53, 2005.
- [64] Bill Doolin. Sociomateriality and boundary objects in information systems development. *European Journal of Information Systems*, 21(5):570–586, 2012. doi: 10.1057/ejis.2012.20. ID: TNpubtec-pal/0960085x/2012/00000021/00000005/art00007.
- [65] Meghann Drury, Kieran Conboy, and Ken Power. Obstacles to decision making in agile software development teams. *The Journal of Systems and Software*, 85(6):1239–1254, 2012. doi: 10.1016/j.jss.2012.01.058. ID: TNsciversesciencedirectelsevierS0164-1212(12)00037-4.
- [66] T. Dyba and T. Dingsoyr. What do we know about agile software development? *Software, IEEE*, 26(5), 2009. doi: 10.1109/MS.2009.145. ID: TNieee10.1109/MS.2009.145.
- [67] Tore Dybå and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9): 833–859, 2008. ISSN 0950-5849.
- [68] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. *Selecting empirical methods for software engineering research*, pages 285–311. Guide to advanced empirical software engineering. Springer, 2008.
- [69] Christof Ebert. *Global software and IT: a guide to distributed development, projects, and outsourcing*. John Wiley & Sons, 2011.
- [70] Kathleen M. Eisenhardt. Building theories from case study research. *Academy of management review*, 14(4):532–550, 1989.

- [71] Josiane Fahed-Sreih and David Pistrui. Motives for entrepreneurship: The case of lebanese family businesses. *Entrepreneurship-Gender, Geographies and Social Context*, pages 82–92, 2012.
- [72] Weam M. Farid and Frank J. Mitropoulos. Normatic: A visual tool for modeling non-functional requirements in agile processes. In *2012 Proceedings of IEEE Southeastcon*, pages 1–8. IEEE, 2012.
- [73] Weam M. Farid and Frank J. Mitropoulos. Norplan: Non-functional requirements planning for agile processes. In *2013 Proceedings of IEEE Southeastcon*, pages 1–8. IEEE, 2013.
- [74] Tony Feghali, Zafar U. Ahmed, and Shireen Halawani. Determinants of lebanon’s ict export competitiveness: Evaluating a country’s readiness to export ict. *Journal of Transnational Management*, 12(3):3–23, 2007.
- [75] Brian Fitzgerald, Nancy Russo, and Tom O’Kane. An empirical study of system development method tailoring in practice. *ECIS 2000 Proceedings*, page 4, 2000.
- [76] Brian Fitzgerald, Klaas-Jan Stol, Ryan O’Sullivan, and Donal O’Brien. Scaling agile methods to regulated environments: An industry case study. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 863–872. IEEE Press, 2013.
- [77] Uri Gal, Kalle Lyytinen, and Youngjin Yoo. The dynamics of it boundary objects, information infrastructures, and organisational identities: the introduction of 3d modelling technologies into the architecture, engineering, and construction industry. *European journal of information systems*, 17(3): 290–304, 2008.
- [78] John Gerring and Lee Cojocar. Case-selection: A diversity of methods and criteria. *Sociological Methods and Research*, 45:392–423, 2016.
- [79] Shahla Ghobadi and Lars Mathiassen. Perceived barriers to effective knowledge sharing in agile software teams. *Information Systems Journal*, 26(2): 95–125, 2016. doi: 10.1111/isj.12053. ID: TNwj10.1111/isj.12053.

- [80] Barney Glasser. Theoretical sensitivity: Advances in the methodology of grounded theory. *Mill Valley, CA: University of California*, 1978.
- [81] Göran Goldkuhl. Pragmatism vs interpretivism in qualitative information systems research. *European journal of information systems*, 21(2):135–146, 2012.
- [82] Shirley Gregor. The nature of theory in information systems. *MIS quarterly*, pages 611–642, 2006.
- [83] Peggy Gregory, Leonor Barroca, Helen Sharp, Advait Deshpande, and Katie Taylor. The challenges that challenge: Engaging with agile practitioners’ concerns. *Information and Software Technology*, 77:92–104, 2016.
- [84] Michael Hammer and James Champy. *Reengineering the Corporation: Manifesto for Business Revolution*, A. Zondervan, 2009.
- [85] Tarek Ben Hassen. Knowledge and innovation in the lebanese software industry. *Cogent Social Sciences*, 4(1):1–17, 2018.
- [86] Richard Heeks. Information systems and developing countries: Failure, success, and local improvisations. *The information society*, 18(2):101–112, 2002.
- [87] Richard Heeks. Most egovernment-for-development projects fail: how can risks be reduced? 2003.
- [88] James D. Herbsleb and Audris Mockus. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29(6):481–494, 2003.
- [89] Jim Highsmith and Alistair Cockburn. Agile software development: The business of innovation. *Computer*, 34(9):120–127, 2001.
- [90] R. Hoda, J. Noble, and S. Marshall. Self-organizing roles on agile software development teams. *Software Engineering, IEEE Transactions on*, 39(3):422–444, 2013. doi: 10.1109/TSE.2012.30. ID: TNieee10.1109/TSE.2012.30.

- [91] Rashina Hoda and Latha K Murugesan. Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software*, 117:245–257, 2016.
- [92] Rashina Hoda and James Noble. Becoming agile: a grounded theory of agile transitions in practice. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 141–151. IEEE, 2017.
- [93] Rashina Hoda, James Noble, and Stuart Marshall. Using grounded theory to study the human aspects of software engineering. *Human Aspects of Software Engineering*, 5:26, 2010.
- [94] Rashina Hoda, James Noble, and Stuart Marshall. The impact of inadequate customer collaboration on self-organizing agile teams. *Information and Software Technology*, 53(5):521–534, 2011.
- [95] Rashina Hoda, James Noble, and Stuart Marshall. Developing a grounded theory to explain the practices of self-organizing agile teams. *Empirical Software Engineering*, 17(6):609–639, 2012.
- [96] IDAL. Technology sector in lebanon factbook 2018. Technical report, IDAL, 2018.
- [97] ANAND Jeyaraj, VICKI L Sauter, and M St. Validation of business process models using swimlane diagrams. *Journal of Information Technology Management*, 25(4):27–37, 2014.
- [98] M. Kassab, Maia Daneva, and O. Ormandjieva. Early quantitative assessment of non-functional requirements. *University of Twente Report*, 2007.
- [99] Samia Khan and Robert VanWynsberghe. Cultivating the under-mined: Cross-case analysis as knowledge mobilization. In *Forum qualitative Sozialforschung/forum: Qualitative social research*, volume 9, 2008.
- [100] Heinz K. Klein and Michael D. Myers. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly*, pages 67–93, 1999.

- [101] Mikko Korkala, Pekka Abrahamsson, and Pekka Kyllonen. A case study on the impact of customer communication on defects in agile software development. In *AGILE 2006 (AGILE'06)*, pages 11–pp. IEEE, 2006.
- [102] David Kreps and Julian M Bass. Southern theories in ict4d. In *International Conference on Social Implications of Computers in Developing Countries*, pages 3–13. Springer, 2019.
- [103] Kurt Lewin. Action research and minority problems. *Journal of social issues*, 2(4):34–46, 1946.
- [104] Sherlock A. Licorish and Stephen G. MacDonell. Understanding the attitudes, knowledge sharing behaviors and task performance of core developers: A longitudinal study. *Information and Software Technology*, 56(12): 1578–1596, 2014.
- [105] Yvonna S Lincoln and Egon G Guba. Naturalistic inquiry (vol. 75), 1985.
- [106] Mikael Lindvall, Vic Basili, Barry Boehm, Patricia Costa, Kathleen Dangle, Forrest Shull, Roseanne Tesoriero, Laurie Williams, and Marvin Zelkowitz. Empirical findings in agile methods. In *Conference on extreme programming and agile methods*, pages 197–207. Springer, 2002.
- [107] Mikael Lindvall, Dirk Muthig, Aldo Dagnino, Christina Wallin, Michael Stupperich, David Kiefer, John May, and Tuomo Kahkonen. Agile software development in large organizations. *Computer*, 37(12):26–34, 2004.
- [108] Mark Lycett, Chaitali Patel, Antonio Merico, Nicola Iacovelli, and Sergio de Cesare. Tailoring software development methodologies in practice: A case study. *Journal of computing and information technology*, 16(3):157–168, 2008.
- [109] P. Machamer and M. Silberstein. *The Blackwell Guide to the Philosophy of Science*. Blackwell Philosophy Guides. Wiley, 2008. ISBN 9780470756522. URL https://books.google.com.lb/books?id=Lf_uDyM6gLwC.

- [110] Robert C Martin. *Agile software development: principles, patterns, and practices*. Prentice Hall, 2002.
- [111] Orla McHugh, Kieran Conboy, and Michael Lang. Agile practices: The impact on trust in software project teams. *IEEE Software*, 29(3):71–76, 2011.
- [112] Matthew B. Miles, A. Michael Huberman, Michael A. Huberman, and Michael Huberman. *Qualitative data analysis: An expanded sourcebook*. sage, 1994.
- [113] Sunila Modi, Pamela Abbott, and Steve Counsell. Exploring communication challenges associated with agile practices in a globally distributed environment. In *Proc. of RAISE 2012-Researching Agile Development of Information SystEms Conference*, 2012.
- [114] Nils Brede Moe and Darja Šmite. *Understanding Lacking Trust in Global Software Teams: A Multi-case Study*, volume 4589. Berlin, Heidelberg: Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. doi: 10.1007/978-3-540-73460-46. ID: TNspringers978-3-540-73460-4150957Chap6.
- [115] Amr A. Mohallel and Julian M. Bass. Agile software development practices in egypt smes: a grounded theory investigation. In *International Conference on Social Implications of Computers in Developing Countries*, pages 355–365. Springer, 2019.
- [116] Michael D. Myers and David Avison. *Qualitative research in information systems: a reader*. Sage, 2002.
- [117] Sridhar Nerur, Radhakanta Mahapatra, and George Mangalaraj. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5): 72–78, 2005. doi: 10.1145/1060710.1060712. ID: TNacm1060712.
- [118] Sridhar Nerur, RadhaKanta Mahapatra, and George Mangalaraj. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5): 72–78, 2005.

- [119] John Noll, Mohammad Abdur Razzak, Julian M Bass, and Sarah Beecham. A study of the scrum master's role, 2017.
- [120] Members of Parliament. Law of private investment companies. *Official Newspaper*, 163, 2001.
- [121] Gerardo A. Okhuysen and Beth A. Bechky. 10 coordination in organizations: An integrative perspective. *Academy of Management annals*, 3(1):463–502, 2009.
- [122] Andreas Opelt, Boris Gloger, Wolfgang Pfarl, and Ralf Mittermayr. *Agile contracts: creating and managing successful projects with Scrum*. John Wiley & Sons, 2013.
- [123] Wanda J. Orlikowski and Jack J. Baroudi. Studying information technology in organizations: Research approaches and assumptions. *Information systems research*, 2(1):1–28, 1991.
- [124] Martyn A Ould. *Business processes: modelling and analysis for re-engineering and improvement*. Wiley, 1995.
- [125] Ken W Parry. Grounded theory and social process: A new direction for leadership research. *The leadership quarterly*, 9(1):85–105, 1998.
- [126] Michael Quinn Patton. Qualitative research and evaluation methods, 2002. ID: 44SALALMADS2126832310001611.
- [127] Kai Petersen and Cigdem Gencel. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, pages 81–89. IEEE, 2013.
- [128] Kai Petersen, Peter Roos, Staffan Nyström, and Per Runeson. Early identification of bottlenecks in very large scale system of systems software development. *Journal of Software: Evolution and Process*, 26(12):1150–1171, 2014. doi: 10.1002/smr.1653. ID: TNwj10.1002/smr.1653.

- [129] Minna Pikkarainen, Jukka Haikara, Outi Salo, Pekka Abrahamsson, and Jari Still. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3):303–337, 2008.
- [130] Mary Poppendieck and Michael A. Cusumano. Lean software development: A tutorial. *IEEE Software*, 29(5):26–32, 2012.
- [131] Mary (Mary B.). Poppendieck. Lean software development : an agile toolkit, 2003. ID: 44SALALMADS2166024420001611; Includes bibliographical references (p. 187-194) and index.
- [132] Scarlet Rahy and Julian Bass. Information flows at inter-team boundaries in agile information systems development. In *European, Mediterranean, and Middle Eastern Conference on Information Systems*, pages 489–502. Springer, 2018.
- [133] Scarlet Rahy and Julian Bass. Overcoming team boundaries in agile software development. *Journal of International Technology and Information Management*, 2019.
- [134] Scarlet Rahy and Julian Bass. A post-colonial analysis of agile software development methods in ict4d. In *IFIP WG9. 4 European Conference on the Social Implications of Computers in Developing Countries*. Springer, 2020.
- [135] Scarlet Rahy and Julian Bass. Implementation of agile methodology in developing countries: case study in lebanon. In *IFIP WG9. 4 European Conference on the Social Implications of Computers in Developing Countries*. Springer, 2020.
- [136] Balasubramaniam Ramesh, Lan Cao, and Richard Baskerville. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 20(5):449–480, 2010.
- [137] Felipe Barbosa Araújo Ramos, Antonio Alexandre Moura Costa, Mirko Perkusich, Hyggo O Almeida, and Angelo Perkusich. A non-functional requirements recommendation system for scrum-based projects. In *SEKE*, pages 149–148, 2018.

- [138] A. Rao and M. Gopichand. Four layered approach to non-functional requirements analysis. *International Journal of Computer Science Issues (IJCSI)*, 8(6):371–379, 2011. ID: TNproquest913581792.
- [139] Rod Rasmussen, Tim Hughes, JR Jenks, and John Skach. Adopting agile in an fda regulated environment. In *Agile Conference, 2009. AGILE'09.*, pages 151–155. IEEE, 2009.
- [140] MN Ravishankar, Shan L Pan, and Michael D Myers. Information technology offshoring in india: a postcolonial perspective. *European Journal of Information Systems*, 22(4):387–402, 2013.
- [141] Mark S Reed. *The research impact handbook*. Fast Track Impact, 2016.
- [142] Zelalem Regassa, Julian M. Bass, and Dida Midekso. Agile methods in ethiopia: an empirical study. In *International Conference on Social Implications of Computers in Developing Countries*, pages 367–378. Springer, 2017.
- [143] Linda Rising and Norman S Janoff. The scrum software development process for small teams. *IEEE software*, 17(4):26–32, 2000.
- [144] Daniel Robey, Richard Welke, and Daniel Turk. Traditional, iterative, and component-based development: A social analysis of software development paradigms. *Information Technology and Management*, 2(1):53–70, 2001.
- [145] Colin Robson. *Real world research*, 2016. ISSN 9781119083412.
- [146] Diane Rover, Curtis Ullerich, Ryan Scheel, Julie Wegter, and Cameron Whipple. Advantages of agile methodologies for software and product development in a capstone design project. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–9. IEEE, 2014.
- [147] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131, 2009.

- [148] Per Runeson, Martin Host, Austen Rainer, and Bjorn Regnell. *Case study research in software engineering: Guidelines and examples*. John Wiley and Sons, 2012.
- [149] Nayan B Ruparelia. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3):8–13, 2010.
- [150] Viviane Santos, Alfredo Goldman, and Cleidson de Souza. Fostering effective inter-team knowledge sharing in agile software development. *Empirical Software Engineering*, 20(4):1006–1051, 2015. doi: 10.1007/s10664-014-9307-y. ID: TNspringerjour10.1007/s10664-014-9307-y.
- [151] Mark Saunders, Philip Lewis, and Adrian Thornhill. Research methods for business students, 2003.
- [152] Klaus Schwab. The Global Competitiveness Report. Technical report, World Economic Forum, 2018. URL <http://www3.weforum.org/docs/GCR2018/05FullReport/TheGlobalCompetitivenessReport2018.pdf>.
- [153] Ken Schwaber and Mike Beedle. *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River, 2002.
- [154] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, 1999.
- [155] Mohammad Shameem, Rakesh Ranjan Kumar, Chiranjeev Kumar, Bibhas Chandra, and Arif Ali Khan. Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process. *Journal of Software: Evolution and Process*, 30(11):e1979, 2018.
- [156] Helen Sharp, Yvonne Dittrich, and Cleidson RB De Souza. The role of ethnographic studies in empirical software engineering. *IEEE Transactions on Software Engineering*, 42(8):786–804, 2016.

- [157] Darja Šmite, Nils Brede Moe, and Pär J Ågerfalk. *Agility across time and space: Implementing agile methods in global software projects*. Springer Science & Business Media, 2010.
- [158] Adam Solinski and Kai Petersen. Prioritizing agile benefits and limitations in relation to practice usage. *Software quality journal*, 24(2):447–482, 2016.
- [159] Viktoria Stray, Nils Brede Moe, and Rashina Hoda. Autonomous agile teams: challenges and future directions for research. In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, pages 1–5, 2018.
- [160] Lucy A Suchman. *Plans and situated actions: The problem of human-machine communication*. Cambridge university press, 1987.
- [161] Abbas Tarhini, Manal Yunis, and Abdul-Nasser El-Kassar. Innovative sustainable methodology for managing in-house software development in smes. *Benchmarking: An International Journal*, 25(3):1085–1103, 2018.
- [162] Philip S Taylor, Des Greer, Paul Sage, Gerry Coleman, Kevin McDaid, and Frank Keenan. Do agile gsd experience reports help the practitioner? In *Proceedings of the 2006 international workshop on Global software development for the practitioner*, pages 87–93, 2006.
- [163] Georgios Theocharis, Marco Kuhrmann, Jürgen Münch, and Philipp Diebold. Is water-scrum-fall reality? on the use of agile and traditional development practices. In *International Conference on Product-Focused Software Process Improvement*, pages 149–166. Springer, 2015.
- [164] Erik H. Trainer and David F. Redmiles. Bridging the gap between awareness and trust in globally distributed software teams. *The Journal of Systems and Software*, 144:328–341, 2018. doi: 10.1016/j.jss.2018.06.028. ID: TNsciversesciencedirectelsevierS0164-1212(18)30125-0.
- [165] UNDP. Mind the gap a labour needs assessment for lebanon. Technical report, UNDP, 2016.

- [166] Juhani Vaivio. Interviews – learning the craft of qualitative research interviewing, 2012. ID: TNtayfranc10.1080/09638180.2012.675165.
- [167] WEF. Executive Opinion Survey. Technical report, World Economic Forum, 2018. URL http://reports.weforum.org/pdf/gci-2017-2018-scorecard/WEF_GCI_2017_2018_Scorecard_EOSQ130.pdf.
- [168] WIPO. Study on the economic contribution of the software industry in Lebanon. Technical report, WIPO, 2015.
- [169] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*, volume 9783642290442. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012 edition, 2012. ISBN 9783642290435.
- [170] Robert K. Yin. *Applications of case study research*. Sage, 2011.
- [171] Youngjin Yoo. The dynamics of it boundary objects, information infrastructures, and organisational identities: the introduction of 3d modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems*, 17(3):290–304, 2008. doi: 10.1057/ejis.2008.13. ID: TNpubtec-pal/0960085x/2008/00000017/00000003/art00008.
- [172] Fadi A. Zaraket, Majd Olleik, and Ali A. Yassine. Skill-based framework for optimal software project selection and resource allocation. *European Journal of Operational Research*, 234(1):308–318, 2014.
- [173] Darja Šmite, Nils Brede Moe, Aivars Šablis, and Claes Wohlin. Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, 86:71–86, 2017. doi: 10.1016/j.infsof.2017.01.003. ID: TNsciversesciencedirectelsevierS0950-5849(17)30043-5.

Appendix A

Appendices

A.1 Ethical Approval Form



University of
Salford
MANCHESTER

Research, Innovation and Academic
Engagement Ethical Approval Panel

Doctoral & Research Support
Research and Knowledge Exchange,
Room S27, Maxwell Building
University of Salford
Manchester
M5 4WT

T +44(0)161 295 5278

www.salford.ac.uk/

23 April 2019

Scarlet Rahy

Dear Scarlet,

RE: ETHICS APPLICATION STR1819-30 – Tailoring Agile Software Development Methods – Case Study Lebanon

Based on the information you provided, I am pleased to inform you that your application STR1819-30 has been approved.

If there are any changes to the project and/ or its methodology, please inform the Panel as soon as possible by contacting S&T-ResearchEthics@salford.ac.uk

Yours sincerely,

A handwritten signature in black ink, appearing to read 'Dr Prasad'.

Dr Devi Prasad Tumula
Deputy Chair of the Science & Technology Research Ethics Panel

A.2 Interview Guide

I want to ask you about your experience in agile software development. This research involves interviews with different members from the institution you belong to with different roles and responsibilities. I want to learn more about your views on agile software development. Particularly, I am interested in knowing if, and how, tailoring of agile software development methods may enhance the process of software development. I want to ask you the following questions and tape record your answers. I will keep your responses completely confidential and nothing will be shared with the company. I plan to publish interview extracts but I will make sure your names remain anonymous. Can I switch on the recorder?

Your Current Project

How many projects are you working on currently? (If more than one project, ask about each project separately.)

What is the title of your current project?

What is your main role in the project?

What is the overall project structure? And does it differ from one project to another?

Details on your Current Project(s):

In your project, may you specify the following?

- What are the main processes?
- Who are the main actors?
- What is the chronological order of processes performed?
- How is the written communication (formal) transferred from one actor to another? And how frequent?

- How is the informal communication transferred from one actor to another? And how frequent?
- Are there any different kinds of ways to communicate other than the ones previously mentioned?

Agile Implementation

How to you send and receive information?

How do you communicate with your colleagues in the team?

How do you communicate with the people outside the team and at what frequency?

How do you measure the size of the software (poking) to be developed?

How do you estimate the timeline needed to finish a software development? What are the factors that you take into consideration?

What agile software development disciplines aid in estimating the project schedule?

What are the factors that negatively affect your delivery with respect to time?

How do you list the risks for the project? And at what intervals of time?

How often is the customer's feedback portrayed back to you? Does this affect your performance in the next software development project?

How do dependencies between teams affect your flow of work?

Do you apply 100 percent agile methods or do you tailor these methods according to your needs? if yes, how?

About the interviewee

Again, I will keep your responses completely confidential and nothing will be shared with the company. I plan to publish interview extracts but I will make sure your name remain anonymous. I need the following information just to keep record of the data collected.

Name and age:

Organization and Location:

Position in organization:

Role in agile applications and techniques:

May you tell me about your qualifications and work experience (in agile):

A.3 Communication Tools used by Practitioners

Practitioner	Quote	Instant Message	Emails	Video Conference	Kanban Board
Director2	“The most is talking, then messaging, then passing user stories through Trello boards is less frequent but they have bigger impact.”	x			x
HR	“A lot passes through email. We use Trello boards and slack as internal chat. I think these are the three main communication forms. And of course face to face.”	x	x		x
Designer	“We start first with the briefing at the sprint planning that is made by user stories that are put on Trello. . . . Most of the communication is just speaking together or on Slack. . . we use email if the person is not in the office.”	x	x		x
Developer1	“We may communicate through Slack or Trello. But we are in an open office environment, so we are able to talk to each other.”	x			x
Developer2	“We do a lot of communication via Slack. Also we use Trello to track user stories when we develop them in the meeting.”	x			x
Technical Lead	“We use Trello and this is where we set all the user stories for the teams. We use slack. And of course communicate face-to-face daily.”	x			x
Director1	“We use Slack for written chatting, Google hangout for video conferencing, and the digital scrum board Trello for communicating the stories. The developers that are physically together in the office they just talk.”	x		x	x
PR	“We can chat on slack if it is internally. . . I can simply talk to the people. . . I put my stories on Trello.”	x			x
Sales	“I use Slack. . . Rarely via email but it is not a very big office so I just walk to them and ask them a question.”	x	x		

Table A.1: Communication Tools used by Practitioners

A.4 Artefact List at LebDevCo1

Artefact Group	Artefact	Criteria
Program Governance	Reference Architecture	Low
	Architecture standards	Low (when requested by client)
	Risk Assessment	Low
	Contract	Low
	Test Plan	Low
Product Artefact	Product backlog	Low
	Product Binaries	High
	product architecture implementation,	low
	user acceptance tests	low
Release Artefacts	Release plan	Low
	Integration tests	low
	Regression tests	Low
	Release binaries	High
Sprint Artefacts	Burn Down Chart	High
	Status Board	High
	Sprint Backlog	Low
	Sprint binaries	High
	User story estimates	Medium
Feature Artefacts	User stories	Low
	Test Criteria	Medium
	Unit tests	Low
	Issues	Low
	Detailed Design	Low
	Source Code	Medium
	Feature binaries	High

A.5 Daily Scrum Meeting Time Records

Team A	Speaking time	Answer Q1	Answer Q2	Answer Q3
VA1	60	Y	N	N
VA2	70	Y	N	N
VA3	Preferred not to speak the first time	N	N	N
VA4	50	Y	N	N
VA5	190	Y	Y	N
VA6	60	Y	N	N
PO	120	Y	Y	Y
Total Time	17 min (550 sec speaking time. a lot of time was spent to encourage team members to speak)			

Table A.2: Daily Scrum Meeting Time Records for Team A; Day: 1

Team B	Speaking time	Answer Q1	Answer Q2	Answer Q3
VB1	50	Y	N	N
VB2	80	Y	Y	N
VB3	130	Y	Y	N
VB4	60	Y	N	N
VB5	80	Y	N	N
VB6	110	Y	Y	N
VB7	150	Y	N	N
PO	120	Y	Y	Y
Total Time	20 min (780 sec speaking time. a lot of time was spent to encourage team members to speak)			

Table A.3: Daily Scrum Meeting Time Records for Team B; Day: 1

Team A	Speaking time	Answer Q1	Answer Q2	Answer Q3
VA1	80	Y	Y	Y
VA2	110	Y	Y	Y
VA3	100	Y	Y	Y
VA4	90	Y	Y	Y
VA5	110	Y	Y	Y
VA6	110	Y	Y	Y
PO	120	Y	Y	Y
Total Time	13 min (720 sec speaking time.)			

Table A.4: Daily Scrum Meeting Time Records for Team A; Day: 40

Team B	Speaking time	Answer Q1	Answer Q2	Answer Q3
VB1	110	Y	Y	Y
VB2	110	Y	Y	Y
VB3	100	Y	Y	Y
VB4	90	Y	Y	Y
VB5	110	Y	Y	Y
VB6	100	Y	Y	Y
VB7	90	Y	Y	Y
PO	120	Y	Y	Y
Total Time	15 min (830 sec speaking time.)			

Table A.5: Daily Scrum Meeting Time Records for Team B; Day: 40