Requirements engineering in market-facing projects: a case study

Author: Martin Wesselink

Requirements engineering in market-facing projects: a case study

Date
October 3, 2011

Author
Martin Wesselink
Business Information Technology,
School of Management and Governance
Student number 0196614
E-mail m.g.wesselink@student.utwente.nl

Graduation committee
Dr.ir. Hans Moonen
Department of Information Systems and Change Management
University of Twente, School of Management and Governance
E-mail hans.moonen@utwente.nl

Dr. ir. Maya Daneva
Department of Information Systems
University of Twente, School of Computer Science
E-mail m.daneva@utwente.nl

Geoffrey Davitt
Department Chubb Insurance Group, IT
E-mail gdavitt@chubb.com

George White
Department Chubb Insurance Group, IT
E-mail ghwhite@chubb.com

Executive summary

Chubb wants to improve their business processes by automating parts of their communication channels between them and their brokers. Therefore, Chubb is developing market-facing systems that address these business processes. However, the current approach that Chubb is using for developing systems is mainly suitable for developing systems based on existing systems. Due to that market-facing systems are systems that are not based on existing systems; the current approach is not optimal for the development of market-facing systems. Therefore, Chubb asked us to do a research on gathering, negotiating, validating and prioritizing requirements of market-facing projects from a multi-stakeholder perspective. What brings us to the central question of this research, namely:

How to gather, negotiate, validate, and prioritize requirements of market-facing projects from a multi-stakeholder perspective?

In order to answer this question, this research is divided into five sub questions. Namely;
1. What methods/approaches for similar projects and companies are suggested in scientific literature?
2. How does Chubb currently gather, negotiate, validate and prioritize the requirements of market-facing projects?
3. What can be improved in the way Chubb gathers, negotiates, validates and prioritizes requirements reviewing the current practice and comparing against scientific literature?
4. What would be a better way for Chubb to carry out requirements engineering for market-facing projects?
5. How to evaluate the proposed framework/methods?

Literature study
The first sub question is answered based on a scientific literature study. The papers found and used in this study are selected based on certain inclusion and exclusion criteria. The overall requirement engineering approaches found in this literature study are mainly based on plan-based and agile strategies. Plan-based strategies consist of an overall implementation plan and agile strategies consist of small iterations where stakeholders have to work intensively with each other to achieve the highest satisfaction. The techniques found in the literature study to gather requirements are; grouping diverse stakeholders, interviewing experts, the use of case scenarios, the use of default templates and documenting requirements and decisions. To enhance the negotiation process, the literature suggest to keep consistency by the use of an ontology, to negotiate iteratively to solve conflicting demands, to consider the emotional factors to gain a better understanding of stakeholders and their requirements and to use a collaboration system to document the requirements and improve the accessibility of it. For the validation of requirements, the literature suggests to review viewpoints by multiple and different stakeholders, to use cognitive profiles in order to gain a better understanding of the requirements by the stakeholders and to use automated validation mechanisms in case of vast amounts of requirements to speed up the process. Further, mechanisms that are suggested by the literature to prioritize requirements are cost and benefit mechanisms and cumulative voting. In the former, stakeholders collaborative determine what the cost is of implementing each potential requirement and determine how much value the requirement will gain. In the latter, each stakeholder is given an amount of 100 points that he or she has to use to vote on his or her most important requirements. The requirements that have the most votes are marked as important and requirements that have the least votes are marked as not necessary or will be dropped out of the project.

Case study
The second sub question is answered through an exploratory case study that is conducted within Chubb. In this case study, experts were interviewed, available documentation was analyzed and direct observations were done to collect data. Findings from this case study suggested that the overall requirements engineering within market-facing projects in Chubb are done based on an iterative approach where in prototypes are used to negotiate, validate and prioritize requirements and to gather additional requirements. In this process a business specification document is created where in all the requirements, specifications and decisions are logged. The stakeholders in market-facing projects are chosen among three aspects, namely; their expertise, their financial sponsorship and their delivery responsibilities. Techniques that are used to gather requirements are; market-research, workshops, case scenarios and face-to-face meetings. In the negotiation process, Chubb is using visualization tools, face-to-face meetings and a collaboration system to increase the understanding of the

requirements by the stakeholders and to facilitate discussions. If there are conflicting requirements, the requirements engineer reflects these concerns to the scope of the project to determine if there are inline with it. Besides this, the requirement engineer validates the concerns by making everyone aware of the conflict and by retrieving their opinions about it. Further on, the requirement engineer also estimates how much time and money it will cost to realize these conflicting requirements. Based on these aspects, the requirement engineer decides whether to drop or to add the conflicting requirements to the development of the system. The prioritization of the requirements in market-facing projects in Chubb is mainly done through two techniques, namely reflecting the requirements back to the scope of the project and by determining the return of investment on the requirements. When a requirement delivers a high return on investment and is in scope of the project, the probability that it will gain a high priority is most likely. When a requirement delivers a low return on investment and is not inline with the scope of the project, the probability that it will gain a low priority is most likely.

Recommendations

The third question is answered based on the findings of previous research questions. Elements from the literature and the current approach are analyzed to improve the current approach. The overall approach that is currently used can be improved by using the approach proposed by Sen and Hemachandra in [1]. With this approach, Chubb could reduce the chance that requirements may not be recognized or recognized too late. For the gathering of requirements, Chubb could use the cognitive tool proposed in [2] to gain a better understanding of the issue by the business experts. In this way, the business experts would find it easier to express their thoughts on the new system. Further, case scenarios can be used to capture the purpose of the system and the reason why a given design will meet that purpose. Finally, templates can be used in the gathering process to speed up the requirements validations and prioritization process. To enhance the negotiation process, ontology can be used to reduce the chance of misinterpretations of the requirements by the stakeholders and a collaboration system could be used to improve the cost effectiveness, user satisfaction and the final outcomes of the project. The validation process could be improved by using the model suggested by Ahmad in [3]. This model addresses the problem of ambiguous and conflicting requirements by negotiating and validating the requirements in iterations that consists of pre-defined steps. In the validation process, Chubb could use the cognitive method of Carod and Cechich in [2], previously mentioned. With this method, they can improve the sign off technique that is currently used by presenting the requirements in a way that the stakeholders feel most comfortable with. Through this, the stakeholders will better understand what he or she needs to sign off on. Finally, the prioritization process in market-facing project could be improved by using the model of Recheva et al. [4] and the cumulative voting method proposed by Leffingwell and Widrig in [5]. The model of Recheva et al. in [4] could be used as guideline to make sure that the stakeholders consider all important aspects in the prioritization process and the cumulative voting method proposed by Leffingwell and Widrig in [5] could be used to reduce the gap of human subjectivity in the prioritization process.

A framework is made based on these recommendations to answer the fourth research question (see page 59); “What would be a better way for Chubb to carry out requirements engineering for market-facing projects?” This framework consists of small iterations where each iteration consists of a pre-sprint, sprint and post-sprint process. In the pre-sprint process, requirements are elicited through the following techniques;

(i) Market research
(ii) Stakeholder selection based on expertise of the product, sponsors and deliverables
(iii) Interviewing based on cognitive profiles
(iv) Existing system analysis
(v) Case scenarios

In the sprint process, the requirements elicited in the pre-sprint process are negotiated, validated and prioritized. In this process, ontology should be used to reduce misunderstandings and notes should be stored in a collaboration system to understand later on why decisions are made. To solve conflicting goals or requirements, the spiral model of in [3] should be used to iteratively solve those conflicts. Further on, the goals and requirements should be validated through multiple sources and by the sign off technique. To prioritize requirements, the aspects mentioned in [4] should be addressed. In the post-sprint process, the requirements and goals should be documented in the business specification document.

Validation

The last research question: “How to evaluate the proposed framework?” is through some articles found in scientific literature. Winbladh mentioned in [6] 6 guidelines on where a requirements engineering framework needs to consists of. These guidelines are used to evaluate the proposed framework and from this it turned out that the proposed framework covers all guidelines to some extend. Further on, the metrics of Costello and Lui in [7] could be used to measure the impact of the framework in practice. However, these metrics measure only quantitative data. To fully evaluate the framework, also qualitative data need to be considered. In order to retrieve qualitative data, interviews need to be conducted with stakeholders that are familiar with the existing approach and the proposed framework in this thesis.

The implementation of the new framework needs to be done after or simultaneously with a project that is using the existing approach where in the metrics of Costello and Lui [7] are used. In this way, data from both projects can be compared and evaluated.

Limitations and implications for further research

- Due to the fact that this research is only conducted within Chubb, it has its limitations in generalizing the results for scientific field and other practices. To generalize the findings from this study for the field of science and other practices, this study also needs to be conducted in other companies that are developing market-facing systems with multiple stakeholders.
- The researcher did an extensive literature study and found articles based on predefined inclusion and exclusion criteria to reduce the chance of missing important literature. However, during this study the researcher found a vast amount of literature and due to time limitations it could be that there is more literature that could be relevant for this research. Therefore, more research needs to be done to find elements that can contribute to the requirements engineering process in market-facing projects.
- Some interviewees (experts) mentioned that developers do not always understand the business. However, other sources in this study did not indicate this. To find out if developers understand the business, more research needs to be done.
- According to the business experts, the selection of stakeholders is done among three aspects (sponsorship, expertise and deliverables). Nevertheless, other sources in the case study only showed that these aspects are present among the stakeholders but do not provide evidence that the selection of stakeholders is really done among these aspects. More research is required to find out, how stakeholders are selected in projects. Knowing how stakeholders are selected in project could contribute in understanding the requirements among the stakeholders.
- The new framework is based on the framework proposed by Sen & Hemachandra in [1]. However, in their paper they mentioned that their framework is only tested on the correctness and not yet on the completeness of requirements. Therefore, more research is required to test the new framework on the completeness of the requirements.
- The cognitive tool suggested by Carod and Chechic in [2] is only tested in a controlled environment. In order to adapt this tool, it should also be tested in practice. Therefore more research on this is required.

Preface

Seven months ago I started this research project in Chubb Insurance Group Australia to complete my study Business and IT at the University of Twente. I was very interested in doing my research aboard to improve my personal skills and to face the challenge to step out of my comfort zone by going alone. Chubb Insurance Group Australia gave me the opportunity to do a research project in their office in Sydney. This research project is about the requirements engineering in market-facing technology projects that involves multiple stakeholders and is described in this thesis. In this project, a literature study is done to find the best elements in the field of science that can be used for market-facing projects. An exploratory study is done to understand the current situation of how the requirements engineering is done in market-facing projects. Recommendations are drawn and an improved framework is proposed for the requirements engineering based on the findings from both studies. Finally, suggestions are made to evaluate the proposed framework.

During this research project I learnt a lot about the insurance business in the way how they conduct business and their terminology. Besides this, I also gained knowledge from practice and the field of science on requirements engineering, project management, conducting case studies and writing this thesis.

However, without the help of other people I would probably not be able to accomplish this thesis. Therefore I would like to thank everyone who supported me in conducting this research. My special thanks go out to my supervisors at Chubb and at the University of Twente; Geoffrey Davitt, my supervisor at Chubb, who was always willing to help me and gave me the opportunity to do my research project within Chubb Australia. George White, my second supervisor at Chubb, for giving me advice when I was in doubt and his experience as a good project manager. Hans Moonen, my supervisor at the University of Twente, who is always willing to help me, and for his clear feedback and in arranging my internship in Chubb. Maya Deneva, my second supervisor at the University of Twente, for pointing out the research directions when I was in doubt and for giving me clear, structured and straightforward feedback.

Further on, I would like to thank all the interviewees for helping me by participating in my research, all my friends and the all the people that I met in Australia. All of them really helped me in motivating me to finish this thesis. Last but not least, I would like to I thank my family who always supported me in my decisions and are always willing to help me.

Martin Wesselink
Enschede, 03-10-2011

Contents

1. Introduction

Businesses are continuously trying to optimize their business processes in order to gain more benefits. One of the ways to optimize their processes is to automate them by IT. According to [8], the IT quality influences the organisation. The higher the quality of systems, information and service, the better the business will perform. Therefore, businesses are constantly improving existing systems and developing new systems. In order to improve existing systems or to develop new systems, requirements need to be gathered, negotiated and validated.

Currently, the Chubb insurance group company is transubstantiating this and want to improve some of their business processes by automating parts of their communication channels between them and their brokers. Therefore, they are developing a market-facing technology (MFT) system called iClose that will automate a range of business processes and will provide a complete solution to address all aspects of the broker-insurer relationship. At the moment of writing, Chubb realized a beta version of iClose and this beta version needs to be modified against business requirements that need to be gathered and validated. However, due to the fact that this system involves multiple stakeholders from different organizations, Chubb want to know how the requirements can be gathered, validated and negotiated most properly. Therefore, the main goal of this research is to determine how Chubb can improve the gathering, validation and negotiation of the requirements in a complex multi-stakeholder system project.

1.1 Chubb insurance company

Chubb insurance company is an insurance company that was formed in 1967 and is listed in the New York Stock Exchange since 1984. Originally they are an American company who were specialists in marine underwriting business in the seaport district of New York City. Nowadays their head office is located in Warren, New Jersey, United States and it is one of the largest publicly traded insurance organisations in the world.

Chubb provides insurance in all kinds of areas to business and personal customers all over the world. Currently, Chubb has more than 10,000 employees and over 100 branches in North America, Europe, South America and the Pacific Rim.

The organisation is divided into two areas, namely; production departments and service departments. The production departments consist of the following departments;
Chubb Commercial Insurance, Chubb specialty Insurance, Accident and health and Personal lines. The service departments consist of; Administration, Accounts, Claims, Operations Services Division, Human resources, Information Technology, Marketing, Loss Control Services.

In Australia, Chubb is located in Sydney, Melbourne, Perth and Brisbane and offers commercial insurance products, accident and health insurance products, construction and commercial surety bonds. One of the commercial insurance products is the ForeFront package. This package consists of multiple policies and will be digitalized through the iClose system.

1.2 Market-facing Technology and the iClose project

According to the dictionary of marketing, market-facing is the point of contact between the supplier and their customers [9]. A market-facing enterprise, is an organisation that is sensitive to the needs of its markets and customers [9] and therefore it adapts itself to these needs [10]. In terms of the project, market-facing points are between Chubb, her brokers and customers. Chubb as a market-facing enterprise using the iClose project, want to adapt to the needs of the Small and Medium Enterprises by;

- Addressing the gaps of the current manual approach to Small and Medium Enterprises. Currently, quotes and policy for small and medium enterprises are manually produced, which takes too much time and effort. One of the objectives of the iClose system is to digitalize these processes in order to reduce the time and effort to quote and bind. In this way, Chubb can offer Small and Medium Enterprises more cost effective and less time consuming quotes and policies.
- Increasing current and new distribution channels.
  With iClose system, brokers will be able to request quotes and policies digitally and as mentioned before, through the digitalization it will reduce time and costs in the current distribution channels.
- Providing free underwriting and support resources.
  The iClose system will automatically quote, rate and bind policies. This means that underwriters do not directly have to be involved in the process and therefore brokers have the ability to quote, rate and bind whenever they want.
- Improving distribution management information.
  More data will be captured in the iClose system than in the manual process. Through this, the management would be able to gain more statistical information about the quote, rate and bind processes of Small and Medium Enterprises.

By means of these strategic objectives, and as already mentioned, Chubb wants to digitalize a range of business processes by the iClose project to provide a complete solution that addresses all aspects of the broker-insurer relationship. iClose will contain a series of modules and interfaces that can be implemented individually or in any combination to fill the gaps in the current market-facing situation. It will contain a flexible architecture to expand it with other modules as the need arises. The current addressed functionality is represented in Figure 1.

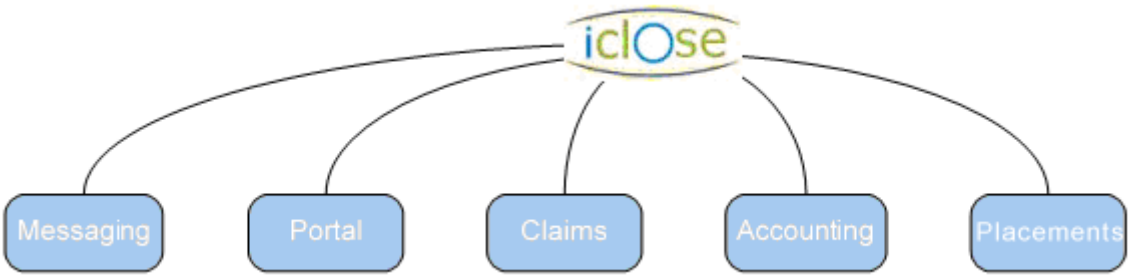


Figure 1: Functionality in the iClose system

iClose Messaging

iClose Messaging is a facility that transfers data between parties in a reliable and secure environment. All the modules use iClose Messaging as the security and transport layer. Features that are included in the iClose Messaging system are;

- Support for four different security profiles
- Synchronous and asynchronous data exchange
- Support attachments
- Message receipt acknowledgement

Parties that use iClose require an iClose Messaging Server to communicate with each other.

iClose Webportal
The iClose Webportal is a complete framework for developing, rating and deploying e-commerce insurance products. It empowers insurers, giving the ability to design, configure and deploy new revised insurance products quickly and cost effectively using the following utilities:

- iDesign - used to build online insurance projects
- eRate – a rating engine. For example to calculate premiums,
- iWrite – used to create documentation that will be returned to the broker,
- iRefer – supports referral for transactions that cannot be rated.

iClose Claims
iClose Claims automates the process of claim handling in the broker/client relationship. With iClose Claims, claims are initially registered through facilities in the broking system and synchronized with the insurer's claim system. This reduces duplication, ensures data in both systems is up to date, provides brokers with a more comprehensive claims profile for their clients and facilitates claims analysis and reporting. iClose claims can be implemented independently and linked to other processing facilities to provide an end to end solution.

iClose Accounting
The accounting functions between the broker and insurers will be automated due to the iClose accounting module. The first accounting process that will be automated is the settlement (payment and allocating money to the outstanding premium) process, where iClose accounting automatically produces settlements files and forwards this to the Underwriter's[*] system. When a broker pays the underwriter by Electronic Funds Transfer, it triggers the upload of the settlement file into the underwriter's system and matches the policies with the payments.

iClose Placements
The iClose Placements module is used to simplify the quote and bind phases within the process of creating / renewing policies (attached in Appendix A), where a quote is the commercial offer and the bind is the agreement before the policy has been officially issued. This module provides the broker with the ability to initiate a quote request, to submit the required risk details to multiple insurers and to monitor and manage quote responses. The module provides insurers with multiple options for responding to the quote request and to convert the quotes to policies. The process of iClose placements is depicted in the Figure 2.

Figure 2: The process of iClose placements.

Overall iClose process
The overall process of iClose product is depicted below in Figure 3. In this diagram the broker access the system and request a quote of an insurance package. The insurers provide the broker a quote and based on the best quote the broker request the chosen insurer to bind the quote. The insurer then bind the quote (policy) and create an invoice for the broker. All steps in the diagram are marked with a number and each number is explained below;

1) Broker accesses the iClose placements via broking system.
2) Broker initiates a quote request within iClose Placements.
3) The broker chooses the insurers to request a quotation.
4) Quote submitted to the insurer via iClose messaging. Insurer receives email advising quotation request received.

[*] A professional that evaluates the risk of insuring a particular person or asset and uses that information to set premium pricing of insurance policies.

5) Insurer accesses quotation request (multiple channels).
6) Negotiations between broker and insurer.
7) Insurer offers terms via iClose Placements, Broker receives email advising terms have been provided.
8) Broker accesses iClose to advise whether cover is required or quotation lost. A request to bind cover is then sent to Insurer and Insurer receives email advising cover request.
9) Insurer then confirms cover bound via iClose Placements, Broker receives email advising cover bound, then uses broker system to upgrade quote to a policy with terms and policy coverage summary.

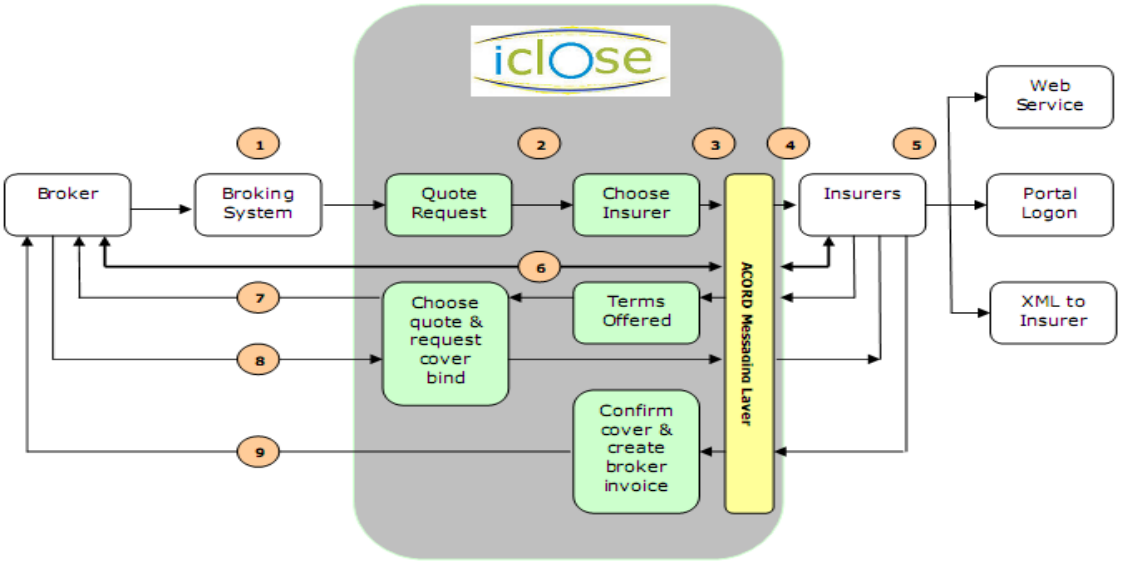


Figure 3: Overall process of the iClose system.

However, as mentioned before in the introduction, the iClose project is still under development and the first product that will be digitalized through the iClose project is the ForeFront insurance package. The reason that this package is chosen is because it has the lowest revenue when underwriting it manually and requires the least effort in digitalizing it because of its standard liability products.

1.3 Forefront Portfolio package

The forefront portfolio package is an insurance package that is designed for private owned companies that cope with many of the most dangerous threats to their financial well-being. For example the risk in the relationships dealing with;

- Employees
- Vendors
- Investors
- Customers
- Competitors
- Government Agencies
- Suppliers
- Creditors

The forefront Portfolio contains eight coverage sections that formulate a comprehensive insurance solution designed to be flexible, able to be tailored to the customers' needs. These eight coverage sections are described in Table 1.

Table 1: Insurance coverage sections of Forefront package

Forefront Portfolio Coverage Section:	**The Risk:**
Employment practice Liability Insurance	Employees and former employees can sue a firm, its board members and its officers for discrimination, harassment and other illegal employment practices.
Director and Officers Liability Insurance	Investors, customers, clients, government regulators, and Insurance competitors can sue a firm's board members and officers over their actions or decisions.
Trustees Liability Insurance	Retirees, former employees, and employees can sue the firm and its plan fiduciaries for alleged mismanagement in administering benefits.
Crime Insurance	A trusted employee can embezzle funds, steal inventory, or commit fraud over a long period of time.
Miscellaneous Professional Liability Insurance	Customers can sue a firm for alleged errors and omissions committed during the delivery of a professional service.
Statutory Liability	A regulatory/Government body may impose a fine upon the company for breach of their statutory requirements
Kidnap/Ransom and Extortion Insurance	An employee can be kidnapped while travelling overseas, or a criminal can attempt extortion against the firm by threatening its employees or products
Internet Liability Insurance	The company can be sued for copyright infringement or defamation over content it posts on its website.

1.4 Problem description

As already mentioned in Section 1.0 and 1.2, Chubb is developing a system that is called iClose. With this system they want to digitalize a range of business processes and provide a complete solution that addresses all aspects of the broker-insurer relationship. The first product that Chubb wants to digitalize is the Forefront insurance packages. In order to do that, this package has to be analyzed and the demands and restrictions from the insurance and brokers have to be understood completely. However, in previous situations, most of the system developments by Chubb were based on old existing (legacy) systems that never considered market-facing technologies. The traditional method that is used in these cases is the System Development Life Cycle (SDLC) where the requirements are gathered from existing systems through a pre-defined framework. After the requirements engineering, the new system was developed, tested and delivered. Due to that, the iClose system is a completely new system that involves a new way of doing business (market-facing), the current SDLC requirements engineering framework does not fit. In this project, there are no existing systems that can be analyzed for the complete development. Requirements have to be gathered, negotiated, validated and prioritized through multiple stakeholders which is a complex process because all stakeholders have different perspectives, requirements and priorities. Therefore, Chubb want to have a requirements engineering framework that can be used in market-facing projects like iClose.

According to [11], Requirements Engineering (RE) is the process of discovering the degree to which the software system meets the purpose for which it was intended. This is done by identifying stakeholders and their needs, documenting these needs in a form that is suitable for analysis, communication and implementation. However, this process encounters a number of difficulties. There are multiple distributed stakeholders that have their own views and goals. These views and goals may vary and conflict with other stakeholders. Stakeholders can have problems making these goals explicit which can constrain their satisfaction due to multiple factors outside their control [11]. Over the last decades several researchers have developed various approaches in order to address this gap, such as Goal-Oriented requirement engineering [12-15], Agile based visualization techniques [16, 17], Agent-Oriented requirements engineering [18] and Rapid Application Design [19]. According to [20] most of these techniques run into trouble when practitioners get lost in the complexity of the methodologies, or analysts lose sight of the goals by becoming fixated on generating deliverables, or the techniques become unmanageable when the number of stakeholders is large. Therefore, currently and over the last several decades researchers have been developing techniques to negotiate and validate the requirements among the stakeholders. In [21], Balfagih & Hassan present a quality model

that classifies non-functional characteristics based on multiple stakeholders' requirements. Arnold et al., propose in [22] a framework that supports the modelling and automated validation of requirements based on a model driven approach. However, this framework is only applicable in the .NET platform by garbage-collected programming language such as C#. There are many other techniques available in academic literature among the validation and negotiation of requirements. In order to create a requirements engineering framework for Chubb that considers market-facing technology and addresses the issue to gather, negotiate, validate and prioritize requirements among multiple stakeholders, the following question needs to be answered;

How to gather, negotiate, validate and prioritize the requirements of market-facing projects from a complex multi-stakeholder perspective?

To answer this main research question, I formulated several sub questions. These sub questions are;

1. What methods/approaches for similar projects and companies are suggested in scientific literature?

2. How does Chubb currently gather, negotiate, validate and prioritize the requirements of market-facing projects?

3. What can be improved in the way Chubb gathers, negotiates, validates and prioritizes requirements reviewing the current practice and comparing against scientific literature?

4. What would be a better way for Chubb to carry out requirements engineering for market-facing projects?

5. How to evaluate the proposed framework/methods?

First, these sub questions need to be answered in order to answer the main question. A literature review needs to be done to find the best elements of how to gather, validate and negotiate the requirements for market-facing systems. After this, the current situation within Chubb about how requirements engineering is done in market-facing projects need to be investigated through an exploratory case study. Based on the findings of this case study and the literature study, recommendations and a framework will be created on the gathering, negotiation, prioritization and validation of requirements that can be used to analyze other insurance packages for the iClose project and other market-facing projects. Finally, limitations will be discussed and suggestions will be made to test and evaluate the new framework.

1.4.1 Practical significance in general

According to Evens [23], Mastering End-to-End Business Processes and Customer Engagement are in the top 10 concerns of Chief Information Officers (CIO). These concerns all involve requirements engineering. Mastering End-to-End business processes is about understanding and gathering the right requirements that are needed to enhance these End-to-End business processes. Customer Engagement also effects requirement engineering, by listening to customers, system and Business Analysts will understand what needs to be changed to get optimal customer satisfaction. According to a survey conducted by Duvall in 2009 [24] the following concerns are all within the Top 10 concerns of CIO's;

(i) IT and business alignment
(ii) Business agility and speed to market
(iii) Business process re-engineering
(iv) IT reliability and efficiency.

All these concerns are affected by requirements engineering. Improving the process of requirements engineering results in a improved Business and IT alignment, business agility and speed to market, Business process re-engineering and in an improved IT reliability and efficiency. Also Luftman and Kempaiah stated in a survey conducted [25] in 2007 that one of the Top 5 concerns is to improve IT quality which is also effected by requirements engineering. Therefore, from a practical point of view it can be concluded that proper requirements engineering is very crucial. Gartner recognized this and

published a recent article [26] about Wiki systems that can be used as a collaboration medium to gather, negotiate and validate the requirements of multiple stakeholders. Besides this, Gartner also [27] mentioned that Project Managers should collaborate on requirements with stakeholders, including executive sponsors, to determine what the primary business driver of the project is (schedule, budget or functionality). Based on these recent publications it can be concluded that they are still looking for improvements on requirements engineering in the field of practise.

1.4.2 Practical significance for Chubb

A requirement engineering framework special for market-facing projects will reduce time, effort and complexity in the requirement engineering process. By reducing the complexity, Chubb would be able to identify issues in the requirement engineering process earlier and better. In the end, all these aspects can decrease the costs in de overall development of market-facing projects. Besides this, having a formalized approach of requirements engineering in market-facing projects also enriches the insight of the market-facing project. Through that the stakeholders get acquainted with the standardized approach, they will be able to understand the status of the project.

1.4.3 Theoretical significance

As can be observed from the introduction of this section, the gathering, validation and negotiation of requirements is also very popular in the field of research. In previous studies lots of methods, frameworks and approaches are described [2, 3, 22, 28-30]. Some of these papers mention that aspects such as cognitive profile mapping [2, 28, 29], tools like Unified Modelling Language (UML) and Object Constraint Language (OCL) [22], decision logging and systematic negotiations [13, 27] will contribute to the requirements engineering. However, as [31] already mentioned, empirical testing is needed to validate these methods, frameworks and approaches. This research can contribute to the discipline of Requirements Engineering by initiating existing models in practice to provide empirical evidence.

1.5 Structure and approach

This research is divided into four parts which are used incrementally. The first chapter is the research approach. This part consists of an orientation study of the research topic to gain insights on the issues that need to be addressed, an overview of the company and project, a problem description, the research objectives, the research questions and research methodology are formulated and described. In the second chapter, the first sub question of the research is addressed by an in-dept scientific literature research to gain best elements mentioned in the field of academics. In the third chapter of this thesis is explained how an exploratory case study is conducted within Chubb and findings from this study are described to answer the second sub question about how Chubb is currently gather, negotiate, validate and prioritize the requirements of market-facing projects. In the fourth chapter;

(i) Recommendations are given based on the findings from the previous studies to answer the research question: “What can be improved in the way Chubb gathers, negotiates, validates and prioritizes requirements reviewing the current practice and comparing against scientific literature?”
(ii) A new framework for requirement engineering in market-facing projects is presented drawn from the recommendations and previous studies in order to answer the research questions: “What would be a better way for Chubb to carry out requirements engineering for market-facing projects?”
(iii) Suggestions are given to evaluate the new framework to answer the last research question: “How to evaluate the proposed framework/methods?”

In the final chapter, conclusions are drawn and limitations and implications for further research are given based on the findings of this research. A schematic structure of the chapters is depicted in Figure 4.

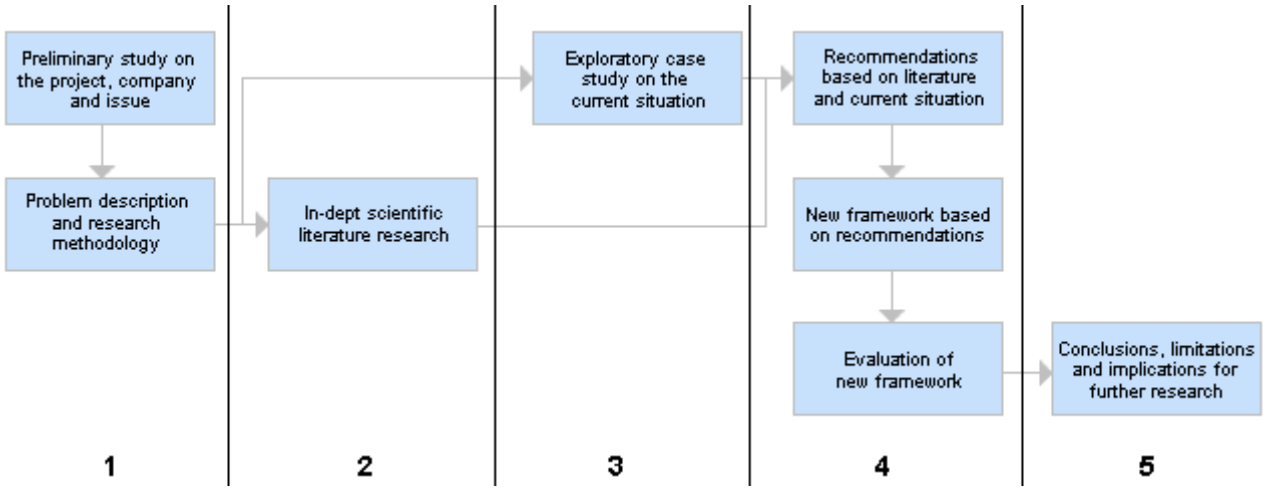


Figure 4: Schematic framework of research structure.

2. Literature study

In this chapter, the elements found in academic literature are described. This chapter is divided into multiple sections, starting with a first section where the literature search methodology is described. Based on that, in the second section, the requirements engineering techniques found in the literature are described. The following three sections consider the four aspects of the main research question, namely; how to gather requirements in a multi stakeholder project, the negotiation with stakeholders about the requirements in a multi stakeholder project and the validation and prioritization of requirements in a multi stakeholder project. The last section presents the conclusion of our findings and provides an answer to the first research question.

2.1 Literature search methodology

The central research question that is addressed by this literature research is the first research question mentioned in the previous chapter, namely: "What methods/approaches for similar projects and companies are suggested in scientific literature?" In order to answer this question, sub questions are formulated. Answering these sub questions will finally answer the concerned research question of this chapter with the focus on the central research question on this thesis. These sub questions are;
- What methods/approaches are used in requirements engineering for similar projects?
- What techniques/methods are used to gather requirements from multiple stakeholders?
- What techniques/methods are used to negotiate requirements among the stakeholders?
- What techniques/methods are used to validate and prioritize the requirements from multiple stakeholders?

All these questions will be addressed through a systematic literature review according to the guidelines of [32]. Relevant papers of the first question will be identified by requirement engineering approach/methods. For the second question, relevant papers will be identified by requirements gathering techniques and methods. For the third question, relevant papers will be identified by requirements negotiation techniques and methods. Relevant papers for the forth question will be identified by requirements prioritization and validation techniques and methods. In the conclusion, the central research question of this chapter will be answered based on the findings from the sub questions mentioned above.

2.1.1 Search strategy

The intention of a systematic literature review is to ensure that a relative consensus of relevant literature is collected [32]. To do this, multiple scientific search citation databases are used that consult multiple scientific libraries and databases. The search databases that were chosen are Scopus and Web of Science. For searching through these databases, the following search strings were used;

For the first question the search strings are:
- (i) E-insurance
- (ii) insurance system requirements
- (iii) requirements multi-stakeholders
- (iv) customer facing internet system requirements
- (v) market-facing requirements
- (vi) requirements engineering insurance.

For the second question the search strings are;
- (i) requirements gathering
- (ii) requirements multi-stakeholders
- (iii) negotiation in the requirements elicitation and analysis process.

The search strings for the third question are;
- (i) negotiate requirements stakeholders
- (ii) requirements negotiation
- (iii) negotiation in the requirements elicitation and analysis process.

For the final sub question the search strings are used;

(i) validating requirements stakeholders
(ii) requirements validation prioritization
(iii) requirements prioritization.

The composition of these search strings are based on a small manual study on requirements engineering. From this manual study, search strings were elicited and iteratively used to get the composition that provided the most relevant results. During this process, varieties of combinations of these strings were used. The reason that Scopus and Web of Science were chosen, is because Scopus addresses multiple other well know databases, like IEEE Xplore, ACM Digital library and Science Direct, and Web of Science is chosen due to the fact that it coverage's the most Top 10 IS journals and Top 25 IS journals [33]. Further on, the search included scientific conference proceedings, journals, magazines and workshop proceedings and was conducted by searching in the title, abstract and keywords. Based on the findings of this search, two other search strategies were conducted, namely a back- and forward search. In the backward search, citations of previously found papers are reviewed, which resulted in new relevant papers. These found papers were screened on their title, abstract and keywords with the same inclusion and exclusion criteria used in the previous search. In the forward search articles were identified that cited previously found papers. These articles were also screen on their title, abstract and keywords with the inclusion and exclusion criteria that was used in the previous search. Further on, for both search strategies the search engine Scopus and Web of Science were used because these engines cover the most conference and journal papers.

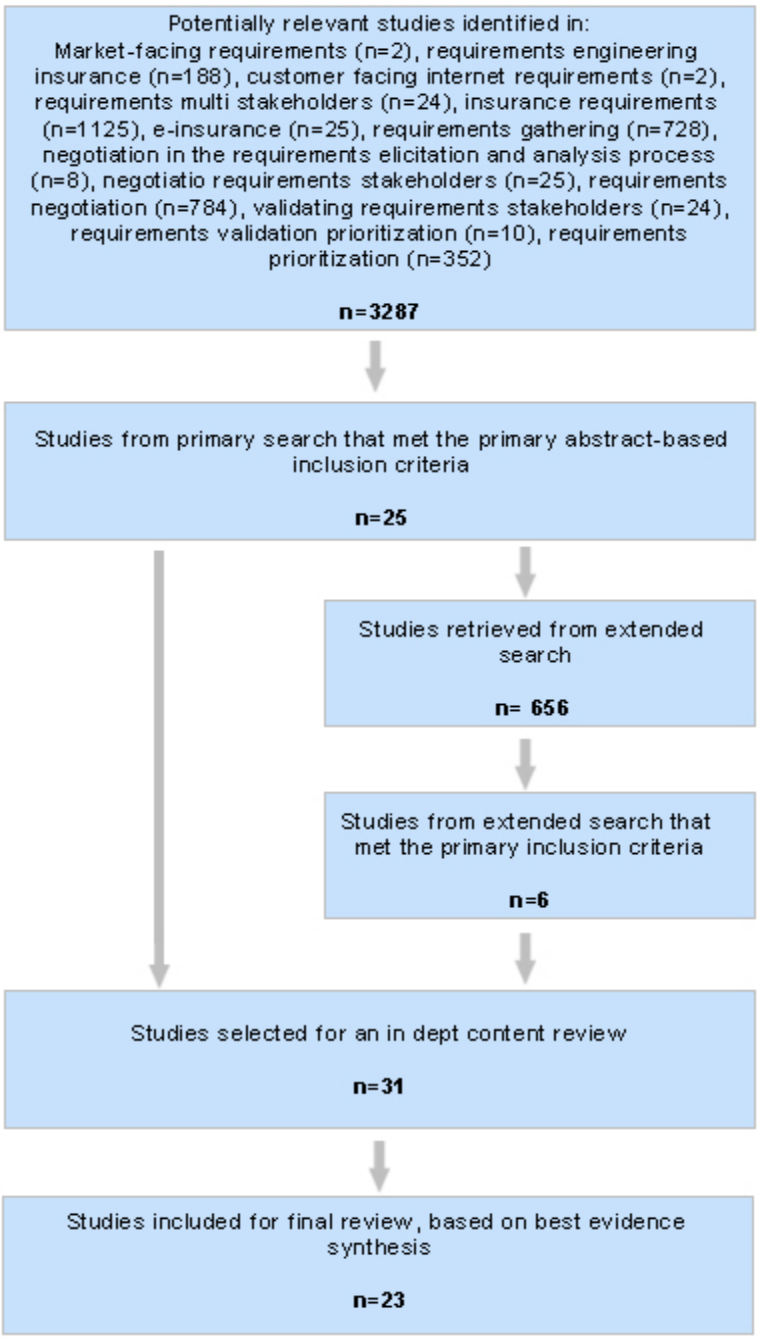


Figure 5: Selection of scientific studies

2.1.2 Selection of the studies

Figure 5 depicts the selection of the papers based on the search strategy mentioned in the previous section.

The search was performed between March 3 and March 10, 2011 and resulted in 3287 potentially relevant studies, 1234 from the search engine Scopus and 1524 from the search engine Web of Science. These studies were screened on the title and abstract-based inclusion criteria.

From this screening, there were 25 studies found that were relevant. These 25 studies were further investigated by doing a forward and backward search on the citations in the studies. During this

search another 6 studies were found that met the primary inclusion criteria. After that, these 31 articles were reviewed more in dept on the content based on the inclusion and exclusion criteria. This resulted in a total of 23 relevant studies. These studies were selected for a final review and analysis of this literature study. These studies are;

- From Scopus:
  [11] [20] [22] [2] [28] [3] [34] [35] [36] [37] [38] [34] [35]

- From Web of Science:
  [29] [39] [40] [41] [42] [43] [44]

- Extended search:
  [45] [4] [46]

During the final review, concepts that are consistent will be identified in the found studies and compared with each other. Besides this, studies were checked if they are duplicated in the found studies or if they used the same data.

2.1.3 Inclusion and exclusion criteria

As mentioned before the found papers were analyzed based on inclusion and exclusion criteria. The inclusion criteria are the criteria that the papers should meet and the exclusion criteria are the criteria that the papers should not meet. If the paper met one of the exclusion criteria, it will be excluded from the research. If the paper met one of the inclusion criteria and did not met one the exclusion criteria it will be included in the research. The inclusion criteria of this literature study are;

I1 – The paper discusses a RE approach for multiple stakeholders.

I2 – The paper discusses aspects of requirement negotiation, validation and prioritization.

I3 – The paper mentioned aspects on requirements gathering methods and techniques.

I4 – The paper discusses RE aspects and approaches for market-facing and insurance projects.

I5 – The paper is original, no duplications in the found literature and use unique data sets.

The exclusion criteria of the literature study are;
E1 – The paper discusses market-facing or insurance project aspects that are not specifically applicable
to RE.

E2 – The paper discusses RE that is not applicable in market-facing or multi-stakeholder projects.

E3 – If papers have the same results. One of them will be excluded.

E4 – The paper only reviews other approaches.

In the last review, 8 papers were excluded. Some of them did not meet the inclusion criteria and others meet the exclusion criteria. The papers that were excluded from the study are listed in Appendix B. Two papers did not meet the inclusion criteria and met the exclusion criteria E1. Four papers were excluded because they meet the exclusion criteria E2. There were two duplicated papers, where one was excluded due to exclusion criteria E3. One paper gave an overview of known methods in scientific literature and was excluded due to E4.

2.1.4 Concept matrix

A literature review should be concept-centric. Therefore, concepts determine the organizing framework of a review [32]. As for the literature study of this thesis, concepts were identified in the last review of the found papers, where the full text was retrieved and analyzed. These concepts are summarized in the concept matrix presented in Table 3. The concepts are organized by the gathering of requirements, the negotiation of requirements and the validation and prioritization of requirements. In the table each concept is given a number and described in Table 2.

Table 2: Description of concepts

Concept description	
1 = Requirements approaches	8 = Iterative
2 = Diverse stakeholders & Pairing staff	9 = Emotions
3 = Interviewing	10 = Viewpoint stakeholders
4 = Logging	11 = Cognitive selection
5 = Scenarios & Stakeholder positions	12 = Automated validation mechanisms
6 = Collaboration system & Logging	13 = Cost Benefit agile prioritization
7 = Consistency	14 = Cumulative Voting

Table 3: Concept matrix

Articles	Concepts													
		Gathering				Negotiation				Validation & prioritization				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Nuseibeh and Easterbrook [11]					x					x			x	
Fruhling et al. [20]	x	x			x									
Arnold et al. [22]												x		
Carod and Cechich [2]			x								x			
Kof [28]	x		x									x		
Ahmad [3]	x		x		x			x		x				
Cafer and Misra [29]					x			x			x			
Nejati and Chechik [34]							x	x						
Liu et al. [35]	x									x				
Peng et al. [36]	x			x		x						x		
Laurent et al. [45]		x					x					x		
Gaur et al. [37]					x									
Jiang and Yang [38]							x				x			
Molina [39]	x						x				x			
Wu et al. [40]	x					x		x						
Port and Bui [41]	x	x						x		x				
Kelly [42]	x			x	x			x						
Gorschek and Wolin [43]				x				x						
Racheva et al. [4]													x	
Colomo-Palacios et al. [44]									x					
Chatzipetrou et al. [47]														x
Sen and Hemachandran [35]	x													
Ramos et al. [46]									x					

Concept 1 addresses the approaches in requirements engineering. Concepts 2 to 5 are mainly used in the gathering process of requirements. The concepts 6 till 9 are categorized in the requirements negotiations process and the concepts 10 to 14 are categorized in the requirements validation and prioritization process. Although these concepts are divided into categories, they are overlapping to a certain extent. For example iteration is not only considered during the requirements negotiation, but also during the requirements gathering, validation and prioritization. To begin with, the first concept, requirements engineering approaches, is explained and described in the next section.

2.2 Requirements engineering approaches

Many different approaches are proposed in the found literature. However, most of these approaches have similarities with each other. The approaches that differs the most from each other are explained in this section. Starting first with the goal oriented approach. In the goal oriented approach, the focus is on the goals of the stakeholders. According to [48] a goal in requirements engineering is; “an objective that the system under consideration should achieve”. In general, in the goal oriented approach, the first step is to identify the goals of the stakeholders where different stakeholders have different goals which may conflict with each other. There are several approaches proposed to analyse

which goals are important and to determine what the main goal of the system is and how to solve goal conflicts [28] [39] [35].

Some other approaches are based on collaboration engineering [36] [20] [40]. According to [49] collaboration engineering is an approach to design re-usable collaboration processes and technologies that are meant to produce predictable success among practitioners of recurring mission-critical collaborative tasks [49]. According to [20], the purpose of collaboration engineering is to quickly and repeatedly create an environment that promotes creativity and communication between participants and consists of the processes described in Table 4.

Table 4: Processes in collaboration engineering [20]

Pattern	Description
Generate	Move from having fewer concepts to having more concepts.
Clarify	Move from less to more shared meaning for the concepts under consideration.
Reduce	Move from have many concepts to a focus on fewer concepts deemed worthy of further attention.
Organize	Move from less to more understanding of the utility of priority of concepts toward goal attainment.
Build consensus	Move from having more disagreement to having less disagreement on course of action.

Relating this to requirements engineering, the first process “generate” is to collaboratively generate as many requirements as possible by the stakeholders (gather requirements). The second and third process “clarify” & “reduce” relates to the negotiation of requirements where stakeholders have to discuss which requirements are important and which are not. The final processes “organize and build consensus” relate to the validation and prioritization of the requirements where stakeholders have to validate and prioritize the requirements in order to get a shared understanding of the system. Studies that consider Wiki systems as a collaboration tool to gather, negotiate, validate and prioritize requirements use this kind of approach.

The approaches described above are based on plan based strategies. These strategies have an overall implementation plan. However, in reality, it seems that requirements can change over time which can change the whole implementation plan. Therefore another strategy is developed; this strategy is called Agile where approaches use small iterations where stakeholders have to work intensively with each other to review each other to achieve the highest satisfaction. Found studies that consider this strategy are [41] [42] [3]. However, in this strategy there is a chance that important requirements may go unrecognized or recognized too late and get not implemented [41]. For example, Sen and Hemachandran tried to address this aspect and proposed in [35] an agile technique that extracts goals from stakeholders. Where they decompose high level goals into lower level or sub goals. They validated their methodology through a case study and found that the correctness of the elicited goals was obtained by the approval of the stakeholders, but the issue of completeness could not be verified.

Reflecting this back to the first sub question, “What methods/approaches are used in requirements engineering for similar projects?” some approaches that are used are goal oriented approaches, collaboration engineering and several other agile approaches. However, all these approaches have their pros and cons. Therefore the next sections describe the elements found in the literature that are important to take into consideration, starting with the gathering of requirements.

2.3 The gathering of requirements in multi stakeholder projects

In the found literature, multiple elements can be used to improve the requirements gathering process. These elements are addressed below, grounded by the related literature.

2.3.1 Diverse group of stakeholders and pairing

As already mentioned in the introduction, a lot of attention has been paid to examine the requirements. As for the requirements gathering process, [20] [41] mentioned that a group of diverse stakeholders is critical. Fruhling et al., evaluated in [20] the collaboration engineering process that would facilitate system requirements validation and elicit new requirements from multiple stakeholders who had different needs. They conducted two sessions where they applied several collaboration techniques. In the second session, Fruhling et al. made more diverse groups of stakeholders by assigning fixed seats to diminish influencing group dynamics such as ranks, personality and organizational background. Each group of stakeholders had technical staff and functional users. As a result, this second session generated twice as many requirements per participant than the first session. Also Port and Bui mentioned, in [41], that the group of stakeholders must be diverse. They suggest that each group of stakeholders must include at least one customer representative and one development manager. In which the customer representative can be seen as the functional user and the development manager as technical staff. However, Laurent et al. criticise in [45] that the requirements generated by joint requirements gathering can lead to increasingly large volumes of stakeholders' requests. This can lead to a total project failure because the amount of requests cannot be realized in the available amount of time.

2.3.2 Interviewing

Besides the diverse group of stakeholders, another technique that can be used to gather proper requirements is interviewing [2] [28] [3]. Ahmad proposes a new negotiation spiral model in [3] that also considers requirements elicitation. According to Ahmad [3], requirements elicitation is the process to extract and identify the requirements systematically from a combination of human stakeholders. She mentioned that one of the techniques that can be used to identify requirements is interviewing. Also Carod and Cechich mentioned in [2] that interviewing should be used for requirements elicitation, especially in new domains. In [2] they did a case study where they map stakeholders to cognitive profiles. There findings showed that the elicitation techniques, like interviewing, should be carefully selected according to the cognitive skill of the stakeholder to improve the requirements engineering process. However, these are only results from a controlled experiment, which have their limitation, as they cannot be generalized to every situation. In [28], Kof suggests that it is necessary to know the rationale that lead to particular requirements in order to solve contradictions in the requirements. By rationale, he means the goal of a particular requirement that is suggested by a stakeholder. In order to identify goals from stakeholders, he mentioned that two key questions could be applied, namely: why and how [28]. An answer to a how-question gives a possible refinement of the goal. Moreover, an answer to a why-question identifies its superior goals. During interviewing, these techniques can be used to identify the goals of the requirements and stakeholders properly.

2.3.3 Logging

Another aspect that needs to be considered is logging. By the use of logging, stakeholders will better understand the requirements. In [43], Gorschek and Wohlin developed a model for market-driven requirements engineering that allows the placement of requirements on different levels and supports abstraction or a break down of the requirements to make them comparable with each other. One of the initial steps in this model, Gorschek and Wohlin mentioned [43] to create an overview of the requirements to the extent of being understood by the Product Manager. This overview of the requirements should at least per requirement contain; a description, the reason, benefit and rationale, the restrictions and risks and a title. By having this overview, the requirement engineer will be able to compare and validate these requirements. Also Peng et al., addresses in [36] the aspect of logging in the sense of a Wiki system. They recognize that in projects with a large scale of stakeholders, the focus of attention becomes a problem. Therefore, Peng et al. suggest in [36] the use of a Wiki system, which has the ability to offer distributed requirements elicitation and documentation. In a Wiki system, the behaviour of the stakeholders among the requirements can be monitored. Based on that, the requirements can be validated. I.e. every stakeholder can browse through the existing requirements. The longer a requirement version the more times it is browsed, indicates it is more stable [36]. However, Kelly proposes in [42] a framework that can be used to improve the facilitation of requirements elicitation. As part of this framework, she proposes the use of user stories documented on story cards, where stakeholders have to document their story from an user perspective. These stories are then used to identify a high-level plan of the project and to provoke an in dept discussion

between developers and stakeholders in order to draw up proper requirements for the system under development.

2.3.4 Scenarios and templates

The user stories technique that Kelly mentioned in [42] can also been seen as a scenario technique, where every story card is a different scenario. The use of different scenarios can contribute to the requirements gathering process where the same aspect will be treated from multiple viewpoints, which will rise in more requirements. Nuseibeh and Easterbrook mentioned, in [11], that the only way to capture the system's purpose and the reason about whether a given design will meet this purpose is by describing the environment and expressing what the new system must achieve in that environment. Nuseibeh and Easterbrook suggest in [11] to do this by modelling stakeholders' goals and by using scenarios that illustrate how goals can be achieved. Gaur et al., propose in [37] a Multi-Person Decision-Making Model to negotiate and integrate the requirements of various stakeholders. In this model, they use user stories as a scenario technique to acquire requirements from the users. In order to retrieve these user stories, they made a prescribed template as follows; "As a <user type>, I want to <goal> So that <reason>". They same kind of technique is also used in [20] where Fruhling et al. used a template of how a requirement should be stated. This helped them to speed up the process of categorizing and validating the collected requirements. Another paper where templates are used to collect the information from users is [29]. Cafer and Misra proposed in [29] a cognitive requirement specification model that is based on the cognitive classification of customers. For the collection of the requirements, they proposed two different types of models based on the cognitive capabilities of the users. Each model contains a template that has to be used to gather the requirements from the users. Also Ahmad addresses the concept of scenarios in her paper [3]. Besides interviewing, she also mentioned the use of use-cases to gather requirements. A use case is a description of potential series of iterations between the system and its environment [50] that can be seen as a description of several scenarios in which the system interacts.

2.3.5 Conclusion

In summary, according to the found literature, the aspects and techniques that need to be considered to gather the requirements properly are; the use of a diverse group of stakeholders where each group at least should have one person from the development staff and one person that is actually going to use the system. Stakeholders should be interviewed in order to get a complete set of requirements. Questions should be made based on cognitive profiles, including why and how questions to refine goals and identify superior goals. The elicited requirements should be logged in a central system in order to compare and validate the requirements among the stakeholders. Scenarios should be used to gather multiple viewpoints from stakeholders and the environment, which improves the requirements elicitation process. Templates should be used to speed up the categorization and validation process.

2.4 The negotiation of requirements in multi stakeholder project

The negotiation process in requirements engineering has been extensively researched. Many papers have been published on this topic. However, due to the limitations of time, only those papers that are potentially interesting for the situation of Chubb are addressed in this research. Elements found in these papers are described below.

2.4.1 Collaboration system and logging

As already mentioned in the section above about the gathering of requirements in a multi stakeholder projects, Peng et al. suggest in [36] the use of a Wiki system to document and validate requirements. However, they suggest that the Wiki system is also sufficient for requirements negotiation. By the use of a Wiki system, stakeholders can modify the requirements by giving a mandatory motivation. Based on that, other stakeholders can mark the current requirements version according to their own preference that will indicate the degree of satisfaction of that specific requirement [36]. Once the requirement reaches the required degree of satisfaction, it will become mature and suitable for further development. Due to this technique, distributed stakeholders can negotiate with each other and all the discussions are recorded which could be convenient to know later on why certain decisions are made. Besides Peng et al. also Wu et al, suggested the use of Wiki system in [40]. In [40] Wu et al. tested 32

real-client, graduate level team projects using a wiki-based requirements negotiation support system called WikiWinWin. This system is based on the WinWin equilibrium theory of [51]. The WinWin equilibrium theory links win, conditions, issues options and agreements. It establishes a win-win equilibrium where all win conditions are covered by agreements and all issues are resolved by options [40]. The results of the test showed that; 1) better project outcomes were correlated with the frequency of use. The more the Wiki system was used, the better the requirements results. 2) The users found the Wiki system beneficial to the requirements engineering process. 3) The tool showed improvements on cost-effectiveness. However, the usability for non-experts is still challenging and it is not sure whether the frequency of use is the real cause of the improved results. It could be that these stakeholders had more expertise in general.

2.4.2 Consistency

Many software projects fail due to incomplete or incorrect requirements. In order to get the right requirements, information needs to be gathered from stakeholders. This information then needs to be interpreted, analyzed and validated. However, due to the fact that most customers could hardly articulate their requirements it can easily be misunderstood by other stakeholders [38]. Therefore, Jiang and Yang propose a model in [38] to elicit performance requirements from customers for financial information systems based on ontology. By using an ontology model they want to bridge the cap of inconsistency. The ontology describes the requirements in a formal and a unified language to avoid misunderstanding between different stakeholders [38]. Luarent et al., recognize this in [45] and stated in their conclusion that consistently using a project glossary throughout the software development lifecycle can also significantly improve performance. In [45] they propose an approach for automating a significant part of the requirements prioritization process due to a non-functional requirement classifier. This classifier is a data-mining tool that classifies a broad variety of non function requirements types related to attributes such as security, performance or usability. In [39], Molina et al. propose a measurable requirements metamodel that offers support to the elicitation of measurable requirements. This measurable requirements metamodel provides users with the abstract syntax for the specification of measurable requirements. By using an abstract syntax, Molina et al., addresses the gap of inconsistency due to abstraction that describes the way how requirements should be formulated. This would benefit the negotiations of the requirements among the stakeholders. Nejati and Chechik present in their work [34] a formal framework for merging and conflict resolution in the negotiation process. They stated that different stakeholders use different requirement models. To solve this conflict, these models need to be merged. This merging can only be done when these models are consistent. According to Nejati and Chechik stakeholders have to negotiate in order to solve these inconsistencies. Once these inconsistencies are resolved, the stakeholders can use the framework presented in [34] to merge their requirement models and to align their requirements with each other.

2.4.3 Iterative

According to many studies [34] [3] [29] [30] [42] [41] [43] [40] the requirements negotiation process needs to be done iteratively in order to get the best requirements for the system under development. As mentioned before, in [3] Ahmad introduces a spiral model that consists of elements that support the requirements elicitation and analysis process. In the analysis process of this model, negotiations amongst the stakeholders have to occur in order to get an overall agreement on the requirements. According to Ahmad this is an iterative process and therefore her model allows renegotiation. She describes that the requirements negotiation process exists of; 1) Identify conflicts. 2) Develop alternatives solutions. 3) Elaborate solutions. 4) Judgement and trade-off. 5) Evaluate and analyse agreement. If stakeholders still have disagreements after “evaluate and analyse agreement”, then this has to go into another spiral beginning with the first step: Identify conflicts [3], see Figure 6. Wu et al., state in [40] that; “Stakeholder collaboration is the key success factor for requirements negotiation. During the negotiation the stakeholders have to collaboratively and incrementally found out what has to be build.” The process of collaboratively and incrementally negotiating the requirements is an iterative process where stakeholders have to share knowledge, understand each other and to resolve conflicts of interests [40]. These activities are similar to the negotiation activities described in [3]. As mentioned before in Section 6.1, in [42], Kelly proposes a framework to improve the facilitation of agile requirements elicitation. According to her, multiple diverse stakeholders have to continuously negotiate with each other to clarify requirements. This remains a difficult aspect in the requirements engineering and her framework tries to address this issue. The results of her study indicate that writing user stories

achieved in an agile (iterative and incremental development) manner gives positive results on exploring and classifying requirements [42].

Figure 6: Requirements negotiation process explained in [3]

2.4.4 Emotions

An element in the requirements engineering that did not have much attention is the emotional factor of humans that are involved during the process [44]. The requirements engineering process is knowledge intensive and because of that it also is human intensive. Therefore, one of the key elements that play an important role during the negotiation and acceptance activities of the requirements is the emotional factor. According to Ramos and Barry [46], considering the emotional factor in requirements engineering will contribute to the transformation process that involves the use of a new system to users and in defining the requirements in ways that are beneficial for the stakeholders. Colomo-Palacios et al. [44] proposed in their paper a tool for analyzing and evaluating emotions in order to create a better understanding of the requirements and for the categorization of the requirements. The tool that they propose is the Affect Grid psychological tool that is created by Russel et al. [52]. During the requirements engineering process, stakeholders have to express the emotion that requirements raise for them. After each iteration, the requirements and emotions change and will be registered. The collected data is then encoded together with the requirements to ensure the pursued emotional traceability. Based on that, the requirement engineer will be able to know the stakeholder's emotions that involve understanding the stability and reliability of the definition of the requirements. By knowing this, it helps the requirements engineer or manager to create an environment that is capable of combating the effects of bad emotions.

2.4.5 Conclusion

In summary, the elements found on the negotiation of requirements in a multi stakeholder's project are; 1) the use of a collaboration system where all the decisions made are logged. This will help later on to analyze and understand why decisions are made, which reduces time and effort. 2) During the negotiations, definitions and requirements need to be consistent. This avoids misunderstanding between stakeholders and improves the performance of the negotiation process. 3) Iterative negotiations about the requirements, where stakeholders have to participate to get an overall agreement among all stakeholders. 4) Emotional factors should be considered to understand the stability and reliability of the definitions of the requirements.

2.5 The validation and prioritization of requirements in a multi stakeholder's project

In this section, the elements of validation and prioritization of requirements found in the literature are presented and discussed. The addressed papers are selected on the relevance of this study. The first element that is addressed is the viewpoint of the stakeholders.

2.5.1 Viewpoint of stakeholders (source validation)

Stakeholders have their own perception on the software development process. However, most of these perceptions have similar elements and relate to some extend to each other. This means that most stakeholders often have similar requirements but formulated it in different ways with different priorities. Therefore these requirements need to be validated and prioritized. Liu et al. presented and tested in [35] a Correlation-Based Priority Assessment framework, which prioritizes the requirements gathered from multiple stakeholders by incorporating their overlapping perspective on relationships of requirements. They found that by using this framework, software process requirements with a longer and stronger impact on other requirements from multiple perspectives receive higher priorities [35]. Also [3] takes the viewpoints of stakeholders into consideration in her spiral model. In the elaboration step, stakeholders have to discuss the solution alternatives where everyone can put in their perspective to promote a better understanding among all the stakeholders. Due to the discussion, the stakeholders are automatically addressing the next step: to validate the requirement by evaluating and analyzing them [3]. Nuseibeh and Easterbrook present in [11] an overview of the field of software systems requirements engineering. In their work they suggest that only modelling information flows and system states of the environment and expressing what the new system must achieve is not enough. According to Nuseibeh and Easterbrook the viewpoints in the sense of goals and scenarios from stakeholders that illustrate how goals can be achieved have to be taken into consideration [11]. In [41], Port and Bui mixed two primary strategies for requirements prioritization. Their purpose is to get the best from both strategies without having the drawback. The strategies that they mixed are plan-based strategy and agile strategy. The new strategy contains seven process steps. The second process step in their strategy is: Generate a candidate list of ‘base requirements’ from the current collection of stories, where stories are the viewpoints of the stakeholders. Then in the third step a cost and value ranking technique will be used to asses the collected requirements based on the stories of the stakeholders.

2.5.2 Cognitive selection

Another aspect that can effect the validation and prioritization of requirements are the cognitive skills of the stakeholders. Stakeholders have different skills and have different preferences to analyze requirements. According to Felder and Silverman, these preferences have the following dimensions [53] [54]; sensing/intuitive, visual/verbal, active/reflective, sequential/global. As already mentioned in Section 6.2, Carod and Cechich did a case study in [2] where they map stakeholders to cognitive profiles. In this case study, they tested visual and non-visual specifications based on the cognitive profile of the stakeholders. They found that 81.8% of the respondents with strong visual preferences agreed on feeling more comfortable with visual specifications and 36.4% of the respondents with non-visual preferences felt more comfortable with visual specifications. Based on that, they concluded that stakeholders perform better when their cognitive profiles are aligned with the notation of the software requirements specification [2]. As mentioned before stakeholders vary in their technical knowledge and therefore it may not be suitable to apply one general technique to every stakeholder. Therefore, [29] proposed a cognitive requirement classification model where they classify stakeholders as professional or amateur. Both classes have their own model for validating requirements. However, this model is only suitable in small projects because it is not going too much into details and lacks thorough analysis. The performance requirements model proposed in [38] is also based on a cognitive approach. This model uses different performance metrics for different stakeholder’s roles to validate performance requirements. According to [39], in goal-oriented requirements engineering, the use of visual models is more promoted than textual descriptions about the system. This probably implicates that stakeholders find it easier to understand and to analyze the goals of the system within a graphical presentation. For example in a goal tree, stakeholders can directly see what the main goal (root) is and what the sub goals are to achieve the main goal. This technique can contribute to the validation and prioritization process of requirements among multiple stakeholders.

2.5.3 Automated validation mechanisms

Large projects with many stakeholders often have many potentially conflicting requests and requirements. In these projects, the requirements from stakeholders can rise up to over a thousand requirements. Analyzing these requirements by hand can lead to missed deadlines, disorganized development efforts and late discovery of architectural requirements. Therefore, several researchers tried to automate the process of requirements prioritization. Laurent et al., proposed an approach for automating a significant part of the requirements prioritization process in [45]. The method they

propose uses a probabilistic traceability model combined with a standard hierarchical clustering algorithm to cluster incoming stakeholder requests into hierarchical feature sets. They did a case study to test this model and their findings were that due to the probabilistic nature, the model did not perfectly detect and categorize all the requirements. However, this model could still be applied for low level and exhausting categorization tasks. This allows stakeholders to work at a higher level of abstraction, which reduces their workloads in prioritizing requirements. In [36], Peng et al. proposes a new requirements maturity concept. This concept is applied in a Wiki system where multiple stakeholders collaborate in the requirements negotiation process. In this Wiki system, the requirements maturity concept determines, based on the degree of modification, satisfaction of the stakeholders and the duration, if the requirement is become mature or not. Therefore stakeholders can only pay attention to those mature requirements and analyze them. This could reduce a significant part of their workload as requirement analysts [36]. As mentioned before in the introduction, in [22], Arnold et al. created a validation framework that supports modelling and automated validation of a set of functional and non-functional requirements. This framework is based on the idea of expressing requirements in terms of responsibilities and scenarios and organizing these responsibilities and scenarios in contracts that can be transformed to components of an actual system. Their findings were that this framework offers a novel solution for automated requirements validation. However, this is very specific to .NET platforms and languages such as C# [22]. In [28], Kof uses natural language processing techniques to automatically identify goals in dialogs of stakeholders. This technique contributes to the Win-Win negotiation approach to resolve conflicts among stakeholders in such a way that the goal of every stakeholder is satisfied. This technique helps to identify conflicts early in the development process.

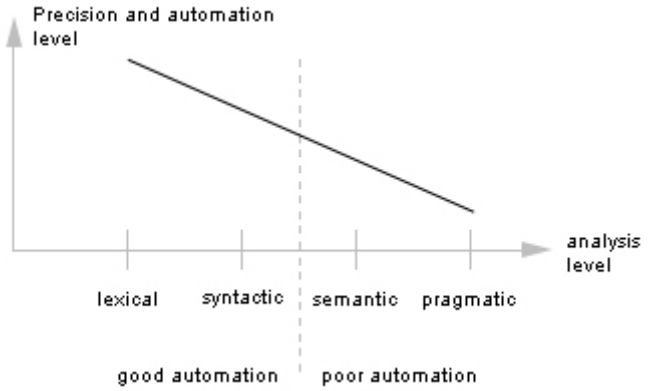


Figure 7: Levels of analysis on goal identification

However, he stated that there are different levels of analysis on goal identification. These levels are; lexical, syntactic, semantic and pragmatic. The best results on goal identification can only be achieved in the first two levels [28]. The precision of the different analysis levels is depicted in the Figure 7.

2.5.4 Cost and benefit prioritization

In a cost and benefit prioritization stakeholders collaboratively determine for each potential requirement what the cost of implementing it would be and how much value the requirement will contribute to the system [11]. As mentioned earlier in this section, Nuseibeh and Easterbrook proposed in their paper [11] a requirements prioritization framework that can be used for agile and plan-based driven software development. Their proposed framework is based on a cost and benefit prediction model and uses two approaches. The first approach is mainly an agile approach and adds a bit of plan based approach by including cost assessment and Pareto ordering in the requirements prioritization. The second approach takes primarily a plan based approach but adds frequent reprioritization (iteratively). They tested their new prioritization approaches through simulations and found that the new approaches produced better results than pure agile or plan-based approaches. However, they only tested their new approach on a limited amount of variations in agile and plan-based software development. Since there are many variations in agile and plan based software development in practise, their results do not represent all these variations. Racheva et al. [4] did a multiple case study on agile requirements prioritization methods and yielded a conceptual model for understanding the prioritization process. The result of the case study suggested that there are five aspects that clients consider in making decisions on requirement priorities. These aspects are;

Business Value, Effort Estimation/Size Measurement, Learning Experience, Input from the developers and External Change. Reflecting this to the cost benefit prioritization approach, the Business Value can be seen as the benefit and Effort Estimation/Size measurement as the cost. However, as the findings of their case study suggested, Learning Experience, Input from the developers and External Change also impact the decision on prioritization and should be taken into consideration in the requirements prioritization process.

2.5.5 Cumulative Voting

Another prioritization mechanism is the Cumulative Voting method. This method is a simple and straightforward voting schema where each stakeholder is given a constant amount of 100 points that the individual stakeholder can use for voting on his or her most important requirements. In this way the amount that the stakeholder assigned to the requirement represents the relative preferences (priorities) he or she prefers among the requirements [47]. However, when multiple stakeholders are involved in the project, this method faces some challenges because not all stakeholders have the same knowledge and influence on the project. Therefore Chatzipetrou et al. proposed a framework in [47] for analysing data obtained from Cumulative Voting prioritization studies. This framework studies the correlation structure of the data, grouping of variables and respondents and also detects the outliers, which contributes to the process of cumulative voting from a mutli-stakeholder perspective.

2.5.6 Conclusion

In summary, the important elements found in the literature to validate and prioritize the requirements in a multi stakeholder's project are; viewpoint of stakeholders, cognitive selection, automated validation mechanisms, cost and benefit prioritization and cumulative voting. Due to the viewpoints of the stakeholders, the requirement analyst will be able to better understand the importance of the value of the requirements. Based on the viewpoints, the requirement analyst can determine which requirements are most important and which are less important. Secondly, stakeholders perform better in validating requirements when their cognitive profiles are aligned with the notation of the requirements specification. Therefore, cognitive selection is an important concept in the validation process of requirements. Using automated validation mechanisms can reduce requirement validation workload in large projects where many stakeholders are involved that together can have thousands of requirements. Automated validation mechanisms automatically categorize requirements, which makes it easier for requirement engineers to do the validation and prioritization. Cost and benefit prioritization techniques allow stakeholders to collaboratively determine for each potential requirement what the cost and benefit of implementing it will be and how much the requirement will contribute to the system. Finally, Cumulative Voting is a simple method consisting of a voting schema where each stakeholder is given a constant amount that he or she can use for voting on the most important requirements. In this way the most important requirements among the stakeholders can be discovered and prioritized.

2.6 Conclusion – Answer to Research Question 1

The found elements from this study can contribute to the thesis by creating an understanding of the issues that occur during the requirements engineering process. The concepts address issues that occur during the requirements engineering process. The found literature in this study proposed approaches, models and frameworks that address these issues and answers the first research question; “What methods/approaches for similar projects and companies are suggested in scientific literature?”. In summary, the overall requirement engineering approaches that are found in the literature are; plan based approaches (i.e. goal oriented and collaboration engineering approaches) and agile approaches that consists of small iterations to address changing requirements. The techniques found in the literature to gather requirements are; grouping diverse stakeholders, interviewing, documenting, and the use of case scenarios and standard templates. For negotiation, the found literature suggest; to use collaboration systems, to document decisions, to be consistent for example by the use of a ontology, to do negotiations iteratively and to consider the emotional factor of the stakeholders. Techniques that can be considered to validate requirements are; reviewing different viewpoints from different stakeholders, making use of cognitive tools to gain an better understanding and to use automated validations mechanisms when there is a vast amount of requirements. Finally, the techniques found in the literature that can be used for prioritizing the requirements are; prioritizing based on costs & benefits and by the to do cumulative voting among the stakeholders.

After understanding the current situation and the issues that occur within Chubb, the found techniques, approaches and methods in this chapter can be used to improve the current process of requirements engineering in market-facing projects within Chubb. Based on the found approaches, mentioned in Section 2.2, and the current situation at Chubb, the most suitable approach can be gained (goal oriented, collaboration engineer and plan-based or agile). Gathering techniques found in this literature study can be used to optimize Chubb's requirements gathering process. During the requirements negotiations, techniques for negotiation found from the literature can be used to improve the negotiations and the understanding of the requirements among the stakeholders. The found validation techniques can be used to optimize the correctness of the requirements and prioritization techniques can contribute to the prioritization process by determining the importance of the requirements in a systematic way.

However, this literature study is not exhaustive meaning that not all the elements discussed in the scientific literature are addressed. Due to time limitations and the vast amount of available literature on requirements engineering, it was not possible to address all elements. Only the elements that were important, according to the thoughts of the researcher, are addressed. Further research can be conducted to gain more elements that can be used during the requirements engineering. The next chapter describes a case study within Chubb in order to gain an understanding of the current situation of the requirements engineering process.

3. Case study – The current situation

In the first section of this chapter, the setup of the case study research is described. Findings from the case study research are described in the second section. In the final section of this chapter, the research question 2 is answered based on the findings from the previous section.

3.1 Case study research methodology

As mentioned before, this study uses a case study as one of its primary research method. The reason for this is that this study tries to understand and improve the phenomena in practice, this requires a qualitative research to understand all variables. Action research would have been an applicable research as well, however this costs more time and would not be suitable within the time available for this thesis.

The research question that needs to be answered in this case study is: “How does Chubb currently gather, negotiate, validate and prioritize the requirements of the iClose system?” which is presented in Section 1.4. Through the fact that this question is a “how” question, this case study will investigate a process.
In this question, the process that needs to be explored is the gathering, negotiation, prioritization and validation of the requirements currently at Chubb in market-facing technology projects. Through that there is little known about the current requirements engineering in market-facing projects within Chubb, the most appropriate design for this research is an exploratory case study [55]. The purpose of this study is to collect data and to make knowledge explicit about concepts that can be improved in the requirements engineering process of market-facing projects at Chubb. During this exploratory case study, the following units of observations were observed.

- Stakeholders
    - Role
    - Behaviour
    - Meeting attendance
    - Expertise
- Meetings
- Iterations
- Available Documentation

3.1.1 Case study research questions

In order to answer the research question:” How does Chubb currently gather, negotiate, validate and prioritize the requirements of the iClose system?”, sub questions are formulated and presented in Table 5. Data that is collected through these questions is both qualitative and quantitative. The nature of the data is specified for each question. Qualitative data is collected from interviews with stakeholders and quantitative data is collected by direct observations due to counting the iterations, meetings, requirements and attendance of the stakeholders.

Overall requirement engineering process		
SRQ 2.1	Does the project have iterations in terms of requirements engineering?	Quantitative / Qualitative
SRQ 2.2	Are stakeholders using a standard methodology or templates during the requirements engineering?	Quantitative / Qualitative
SRQ 2.3	What templates of methodologies are stakeholders using during the requirements engineering	Qualitative

Requirements gathering		
SRQ 2.4	How are requirements gathered?	Qualitative
SRQ 2.5	How many requirements are collected?	Quantitative

Requirements negotiation		
SRQ 2.6	How many stakeholders are involved during the requirements engineering?	Quantitative
SRQ 2.7	How many conflicting demands occurred during the requirements engineering?	Quantitative
SRQ 2.8	How are these conflicting demands addressed?	Qualitative
SRQ 2.9	How are requirements communicated to stakeholders?	Qualitative
SRQ 2.10	Do the developer and the other stakeholders understand each other?	Qualitative/ Quantitative
Requirements validation		
SRQ 2.11	How are requirements tested/validated?	Qualitative
Requirements prioritization		
SRQ 2.12	How are requirements prioritized?	Qualitative

Table 5: Sub questions of research question 2

3.1.2 Case selection and generalization

For this investigation, two cases are selected. The first case is the investigation of the requirements engineering of the iClose project which is explained in Section 1.2. The second case is the investigation of the requirements engineering in a similar project that involves multiple stakeholders and addresses market-facing technology in order to be able to generalize the findings and limitations of this study. This project is called the ARCH project.

At the moment, Chubb Asia Pacific is using Kaizen tools to create, rate and issue insurance policies. These kaizen tools are Word and Excel templates that calculate the premium and generate the wording of the policies. In the ARCH project, these tools will be replaced by a system that is called Avantage. This software package is a custom made package for Chubb and developed in Chubb Canada. With Avantage, Chubb is trying to improve the process of creating, rating and issuing policies in order to save resources and in the future to outsource some parts of this process due to market-facing technology. Therefore, during the development of this system, the market-facing technology aspect has to be taken into consideration and multiple stakeholders have to be involved to improve the process. However, in the ARCH case there are limitations, these are;

- The market-facing technology is very limited and therefore is not the main focus of the project, which is in contrast to the iClose project where the main focus is on market-facing technology.
- The ARCH project is based on existing systems that will be reengineered and improved, therefore the requirements engineering deviates from the requirements engineering approach in the iClose project.

Due to these limitations the main focus will be on the iClose project and the ARCH project will be used to ground the findings in the iClose project.

The generalization of the results will be limited to Chubb since the particular cases only occur internally in Chubb. The effort of this study is mainly focussed on particularization [56] and specification [57] because this research is an in-dept examination of the current requirements engineering situation. Further, due to this examination, concepts can be developed that can be implemented and used to do analyses for future research on requirements engineering within Chubb.

3.1.3 Data collection

The data of the cases will be gathered through multiple sources. These sources are; interviews, available documents and direct observations. A detailed description of each source is described below, starting with interviews.

Interviews

Interviews with users in different functional areas and management levels were held to determine how requirements gathering, negotiation, prioritization and validation is done at Chubb in both projects. These interviews were held at the sites and by telephone during the period February through August 2011. Findings from these interviews are mainly qualitative. A standard interview template is used to

guide the interview; this template is attached in Appendix C and explained here. The first couple of questions are used to identify the interviewees' role and position in the company and the project. The section 'definitions of requirements` is used to extract the overall methodology of how the interviewee is doing requirements engineering. The third section of the interview template `gathering requirements` is used to understand the way of how the interviewee is gathering requirements. The fourth section of the template is `requirements negotiation` and is used to understand how the interviewee solves conflicts and negotiates among the stakeholders in the projects. The fifth section of the interview template is 'prioritization and validation of requirements` and is used to gain knowledge about how the interviewee prioritizes and validates the requirements. The last two sections of the interview template, namely 'general requirements questions' and `case scenarios` are used to identify and understand the issues that the interviewee addresses during the requirement engineering process in market-facing technology projects and in multi stakeholder projects. As evidence and with permission of the interviewee and the company, these interviews are recorded on tape and transcribed for analysis purposes.

Available documents
Documents that are available within Chubb are analyzed to validate the findings from the interviews and to find other concepts of the current requirements engineering methodology. The documents that are analyzed are;

- Plan of approach documents, to gain knowledge on how the requirements engineering is done.
- Meeting notes, to gather information about the negotiation among the stakeholders.
- Other documents, it could be that other documents are used during the requirements engineering, which are not included in the plan of approach. Therefore, during the interview sessions, stakeholders will be asked which document are available and used in market-facing technology projects and multi stakeholders projects.

However, the available documents are analyzed critically since these documents can deviate from how the requirements engineering process is done in the real world at Chubb.

Direct observations
Direct observations are done to observe how the requirements engineering process is currently done in the iClose project. However, no observation were be done in the ARCH project because the observer did not had the permission for this. Findings from the observations are documented through a template that is used during the observations and addresses the following aspects;

- The overall requirements engineering process.
  Are they using a plan-based approach or an agile approach?

- The negotiation among the stakeholders.
  How are conflicting requests of requirements from stakeholders solved?
  How does the requirement engineer communicates the requirements to the stakeholders?
  What techniques does the requirement engineer use to gain an understanding of the requirements by the stakeholders?

- The prioritization process.
  Analyzing the methodology that is used to prioritize the requirements

- The gathering of the requirements.
  Which techniques does the requirement engineer use to gather the requirements?

- Number of iterations
- Total number of stakeholders
- Number of stakeholders in meetings
- Number of meetings
- Number of addressed requirement concerns per meeting
- Total number of addressed requirement concerns by stakeholders

Total number of iterations, total number of meetings, number of addressed issues per meeting and total number of requirements are collected to indicate the complexity. The total number of stakeholders, number of stakeholders in meeting, number of addressed issues per meeting are collected to indicate the understanding of the stakeholders. The template used for the direct

observation is attached in Appendix D & E. Further on, during the observations the following aspects are considered;

(i) A requirement concern can have multiple requirements
(ii) Iteration is the amount of loops per requirement concern during the observation
(iii) Total iteration is the amount of meetings about the requirements where the current prototype is discussed during the observation period.

3.1.4 Data analysis

As mentioned in the Section 3.1.2, there are two types of data collected, namely qualitative data and qualitative data. These different types of data have different types of analyses. The analysis of the quantitative data, which is collected through direct observations and available documentation, is done by descriptive statistics e.g., frequency, mean, mode, standard deviation. The results of this analysis are used to support the findings from the qualitative data.

In the case of the qualitative date two analysis approaches were used, namely a within-case analysis and a cross-case analysis. In the within-case analysis approach, the concepts from one source are compared to another source in the same case. Differences and similarities between the sources are described based on this analysis. In the cross-case analysis, concepts found in one case are compared to the concepts found in the other case. Differences and similarities of the concepts between the cases are described based in this analysis. For these analyses, the transcripts that are produced from the interviews and direct observations, and available documentation are used as a data source. These transcripts are analyzed by the use of coding techniques due to a specialized qualitative data analysis software package that is called NVivo. This software package helps to classify, sort and arrange information, to examine relationships in the data and combine analysis. From these analyses, themes are produced, described and explained.

3.1.5 Internal validation

As already mentioned in the section above, there are multiple data sources selected for this case study that are used for analysis. The reason for this is to internally validate the findings of the case study. By having this, multiple sources will eliminate the chance of alternative causes of the results. As mentioned before, these sources are; interviews, available documents and direct observation. Interviews will be recorded, available documents will be copied and transcripts will be made of the direct observations. With this data, other researchers will have the possibility to reproduce the findings found in this research. However, depending on the sensitivity of the data, it could be that interviewees may choose to stay anonymous and parts of the available documents and produced transcripts could be concealed.

3.1.6 External validation

The external validation of the case study is about the generalization of the results. This means that the findings from this case study should also hold in different organizations that are in the same situation. Therefore, this case study should actually be done in two different organizations but due to the fact that this research is only dedicated to Chubb; it would not be possible to investigate another organization. However, two case studies are chosen within Chubb to generalize the research to a certain extent. One case study investigates the requirement engineering in the iClose project that is a multi stakeholder market-facing technology project and the other investigates the requirements engineering in another similar project called ARCH that also involve multiple stakeholders and market-facing technology. As mentioned before, the findings from these cases are compared by a cross-case analysis to find similarities. For example, to gather external valid information, finding X in project (or case) A must also be true in project (or case) B and the other way around.

3.2 Findings from the current situation

As mentioned before in the previous section, interviews are conducted, available documentation is analyzed and direct observations are done in order to retrieve data on how Chubb currently engineers the requirements in market-facing project. The experts that were interviewed are given in Table 6.

Table 6: Interviewed experts

iClose project		
Role	**Reference**	
Project Manager	ICPM1	Appendix F1
CFO	CFO1	Appendix F2
IT Manager (interview iClose)	ITM1	Appendix F3

ARCH project		
Role	**Reference**	
Project Manager	ARPM1	Appendix F4
Business Analyst	ARBA1	Appendix F5
IT Manager (interview ARCH)	ITM2	Appendix F6

Market-facing projects in general		
Role	**Reference**	
Business Partner Service Manager	BPSM1	Appendix F7

3.2.1 Overall process of requirements engineering in MFT projects

During the research period, the overall requirement engineering process is analyzed in order to define which major methodology is used in market-facing projects that involve multiple stakeholders. For this, several questions are defined in Section 3.1.2 that are used as a guideline to gain an understanding of this process. Starting with the first question;

SRQ 2.1: “Does the project have iterations in terms of requirements engineering?”

In the iClose project;
The Project Manager in the iClose project stated in the interview that the requirements engineering is more a back and forward process to elaborate the business requirements. By this, the back and forward process can be seen as iterations, in which the requirements iteratively are elaborated and once it is mature enough it will be signed off;

"It is more of a back and forward process where someone from the business will elaborate the business requirements. The Business Analysts will write it down put it into a document and than back and forward, so it will get refined. Until it is ready to be signed off. [ICPM1]"

During the direct observation, all meetings were addressing requirements concerns and most of these concerns were addressed in multiple meetings (Appendix G). This implicates that the requirements concerns are going through multiple iterations formulate the final requirements.

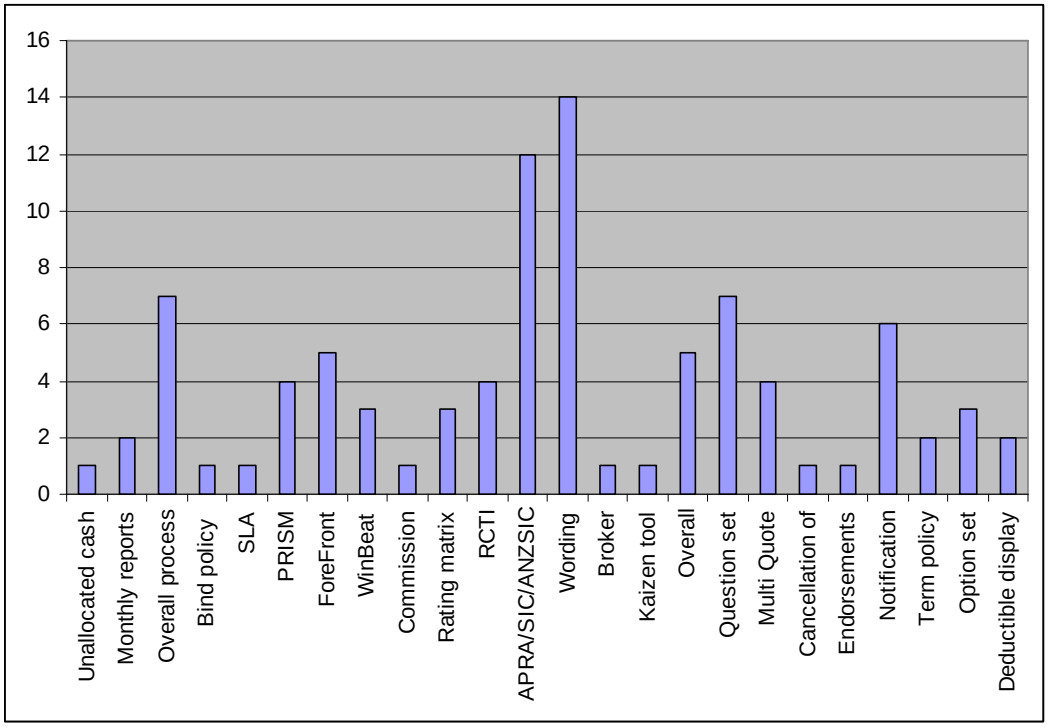


Figure 8: Number of iterations per requirement concern

As depicted in Figure 8, in the observation period, 24 concerns were addressed where 8 concerns had the lowest amount of 1 iteration, 2 concerns had the highest amount of 14 and 12 iterations and the average of amount of iterations per concern is 3,21. Also in the notes that were created after the meetings stated requirement concerns multiple times, which implicated that these are multiple times addressed in different meetings. I.e. in the notes of 24-05-2011 and in the notes of 22-06-2011 (Appendix H) where the occupations code problem is addressed, which means that this is going through multiple iterations in order to generate the final requirements from the concern.

The ARCH project

Two of the four participants that were interviewed from the ARCH project explicitly stated that iterations are used during the development of the project. In the interview with the IT Manager [ITM1] about the ARCH project, he stated that;

"What we are planning to do, and we have done this twice, is to demonstrate it in an iterative approach of the development lifecycle. So basically we develop something of the system and we showed it to make sure that they agree with the changes that we made. [ITM2]"

Further on, the Project Manager of the ARCH project suggested that they are using the agile approach for some parts of the project;

"You are familiar with the kaizen tools that we have here? Yes. Ok, this is the agile portion; the Canadian team have taken the kaizen tools and reverse engineered them into Advantage. [ARPM1]"

This is in line with what the IT Manager says about the iterations because the agile approach consists of multiple iterations. Besides this, there is also a document that consists of a registration of attendance of the participants during the iterations (Appendix I). In this document, there are two iterations where the attendances of the participants are recorded and a third iteration is made that will be used to record the attendance of the participants for the next iteration. This document is an excel document with the following tabs: 14-04-2011 (Prototype Review), 15-04-2011 (INT Demo) and 18-04-2011 (INT Demo). Each tab represents an iteration and consists of a table of attendance, which grounds the statements made by the Project Manager and IT managers.

Findings from question SRQ 2.2 and SRQ 2.3: "Are stakeholders using a standard methodology or templates during the requirements engineering?" and "What templates of methodologies are stakeholders using during the requirements engineering?"

The iClose project
According to the IT manager and the found documentation on the intranet of Chubb (Appendix J) the general methodology that is used for IT projects within Chubb is SDLC;

"*Basically we are following the standard SDLC (software development life cycle). This is a set of phases where we need to go through including testing and releasing, migration testing*" [ITM1].

However, as the CFO and the IT Manager suggested, that the iClose project is different from traditional projects in terms of requirements engineering and development;

"*The iClose project is probably a little bit different from how it normally go about gathering requirements for a IT project*" [CFO1]

"*If you look at the methodology used for the iClose product, it is actually quite a different methodology to what we used for other projects*" [ITM1].

Several sources suggested that the approach used in the iClose project is more a prototyping approach, in which they monitor, elaborate and validate the development of the project on a periodically bases, for example every week. During these iterations the business see how the new system is going to work and based on that they give new requirements and validate the current requirements, the Project Manager of iClose stated that;

"*The other thing is it may minimize the risk is because we are doing prototyping. We don't have people that go a way and build something for 6 months and then come back. We have to see something every week. Because of that we know that some things aren't working and we can change the approach*" [ICPM1].

Findings from the direct observations show that there are several (6 observed) meetings in which the prototype of the iClose system is discussed among multiple stakeholders (Appendix K). During these meetings, stakeholders validated the current system, requirements concerns and suggested new requirements concerns. These findings are in line with the findings from the available documents. For example the outstanding issues from the meeting of 16-02-11 and 22-02-11show that issues of 16-02-11 are discussed and a new requirement concern "Document generation and email to Broker and CICA" is added in the meeting of 22-02-11 (Appendix L).
In the interview with the Business Process Sevice Manager, who was involved in another market-facing project in another country, he stated that this other project also used prototyping as the development and requirements engineering method;

"*That is called PeopleSure, and we are rolling it out in Australia at the moment. In terms of the requirements for that project, it was done in Agile, so more in a prototype type affair*" [BPSM1].

The documentation that is used in that project is a Business specification documented in which the requirements are documented. He said that this document is generally used in these kinds of projects;

"*Yeah, there generally is a kind of a business spec (business specification document)*" [BPSM1].

In terms of the iClose project, this statement is confirmed from several sources. The CFO stated that there is a business specification document that is used to document the requirements. With this documentation, the developer and other stakeholders will be able to understand the requirements that are formulated by the business;

"*There has been a business requirement document writing which have been come out of the meetings with the business. It more a guide or a reference point then a formal document is used by developers to program*" [CFO1].

From the available documentation analysis, this document is found and attached in Appendix M.

Further on, during documentation analysis, another template was found. This template is used in meetings to document and monitor the requirement concerns and consists of a table with the following columns; no, priority, requirement concern, comments from Chubb, comments from the developer, complexity and effort. Examples of the template are presented in Appendix L. Besides this, it seems to be that not all the available documentation and templates within Chubb for software development and requirements formulation are used, according to the CFO of Chubb Australia;

"*We are not using, or following all the documentation that is out there in Chubb. But certainly the important documentation*" [CFO1]

According to the Project Manager this is because the project does not consider very critical processes and because of time limitations it would not be realistic to use all the available documents and templates;

"*I would not be surprised if that is a very detailed framework. We use some of that, but how much we use is definable. Because the other thing is; it depends on the project. If you spending 20 million dollars, you make sure that the requirements are right. If you spent 20 million dollars and the system isn't used. You will get into trouble. On the flip side; if you are spending 20 000 dollars, you do something fast. You think it is going to work; it might work of it might not work, but you got a greater leap up there because you aren't losing that much money. For the iClose project we are spending a couple of 100 000 dollars maybe more, but we have a greater latitude to accept risk*" [ICPM1]

The ARCH project

In the ARCH project, the requirements engineering approach is also mainly based on prototyping. The Business Analyst and the Project Manager of the ARCH project and the CFO mentioned that the major approach for requirements engineering and software development used in the ARCH project is done by prototyping;

"*This is the agile portion; the Canadian team have taken the kaizen tools and reverse engineered them into Advantage*" [ARPM1]

"*They build the product requirement into the prototype and then they discuss the prototype*" [CFO1]

"*When prototype is build we give a presentation. That is how we give the business the chance to show them how the prototype is working. Then they try to use the prototype and they suggest; if it is working the way they want or if they want to change something or if they want to remove something, that kind of stuff*" [ARBA1].

As the Business Analyst already suggested, once a prototype is ready the stakeholders will sit down and validate the requirements and negotiate what they want to change, add or remove. However, some parts of the development and requirements engineering in the ARCH project is also done due to the "old" methodology. The Project Manager stated that if there is functionality required that is currently not supported by the old existing system then the Business Analyst will use this methodology to gather requirements from the experts;

"*But there are certain requirements that are needed in Advantage that kaizen do not support. For those, we have a Business Analyst, taken the old methodology.*" [ARPM1]

However, the Business Analyst clarified that when there are some parts of the requirements not clear, this will be gathered from multiple sources instead of extracting it from the system;

"*You still need to gather the requirements because it is not clear enough or just not right*" [ARBA1]

In the ARCH project, there are several templates used for requirements engineering. The Project Manager mentioned that there are many temples available within Chubb, but, as similar to the iClose project, only the templates that are critical are used;

"*We have standard templates. That are templates that Chubb is using for gathering requirements. Although it is very broad, it kind of makes it what you need to make it*" [ARPM1].

During the documentation analysis, several templates were found. These templates are attached in Appendix N. The first template in Appendix N is used to formulate the business specification document (which is also found in the iClose project). Due to this template the stakeholders, requirements, goals and scope, the testing and implementation are defined and explained. The second template, in Appendix N, is used to formulate the processes that are impacted by the system under development. Once this template is filled out, it explains the impact and the workflow of the system under development.

Conclusions from questions SRQ 2.1, SRQ 2.2 and SRQ 2.3
Based on the findings from the first question, it can be concluded that most of the requirements concerns are addressed multiple times in order to draw the final requirements. From this, it can be concluded that iterations do occur during the requirements engineering and the project development of market-facing projects that involve multiple stakeholders. However, some requirements concerns are addressed only once. This can be through that these requirement concerns are very clear and obvious to everyone and therefore the stakeholders agreed on the concern directly in the first iteration.
Further on, the methodology that is used for market-facing projects that involve multiple stakeholders is generally based on prototyping. This approach consists of periodically meetings wherein the current software prototype of the system is discussed and validated. In this approach the development and requirements engineering are done simultaneously, which has the benefit that requirements can be validated in the next prototype meeting.
Besides this, there seems to be one general template that is used in several kind of market-facing projects. This template is the business specification template (Appendix N) wherein the requirements are documented and elaborated according to the feedback of the stakeholders.

3.2.2 Current situation of requirements gathering in MFT projects

Several aspects are analysed to determine how the requirement gathering is done at Chubb in the iClose and the ARCH project. These aspects involve how stakeholders are selected in the requirements engineering process, how they gather requirements, the techniques and methods they use and how they document their requirements. Based on the findings of these aspects, the questions SRQ 2.4 and SRQ 2.5 are answered.

Selection of stakeholders
Requirements are gathered from several sources. One of these sources is the stakeholders who have certain expertise in which they can contribute to the gathering, validation, prioritization and negotiation process of the requirements. Therefore, the process of how the stakeholders are involved in the project is analyzed. Findings from several sources in the iClose and ARCH project suggest that the stakeholders are selected based on the following aspects;

<u>Expertise of the product</u>
Several sources suggested that the people who have the knowledge of how the product works should be involved during the project as stakeholders [CFO1, ARBA1, BPSM1]. This could be; the end users as well as managers that understands how the product works or should work. For example in the ARCH project the managers are chosen to be the stakeholders because they know how the product works or should work. If they do not know it, they will go to their experts or end-users and asked how it works or should work;

"*Yes but we don't get back to the persons, I get back to the SBU managers. And they need to answer the questions. If they don't know, they can go back to their team and ask the questions*" [ARBA1].

<u>Sponsorship</u>
Another aspect that is mentioned by several sources in the selection of the stakeholders is; sponsorship [ICPM1, BPSM1]. The existence of a project is because there are people who finance the project to achieve a certain goal. These people are chosen as stakeholders in the project because they can translate their goal into requirements. For example as the Project Manager of the iClose project suggested that one of the sponsor stakeholders could be the end customer who pays for the project;

"*There are people they need to be involved because they got to deliver stuff and sign off and the end customer, because they pay for it*" [ICPM1].

Deliverables
A third aspect in the stakeholder selection is suggested by the IT Manager, namely; There several stakeholders chosen by the fact that they have to deliver the final product [ITM1]. Often these people are the developers. These people have expertise in the feasibility of the system under development. They are able to determine what is possible, what not and how much time it will cost.

During the direct observations in the iClose project, the involvement of the stakeholders was registered in order to validate the suggestions by the interviewees. If these factors are considered during the stakeholder's selection then at least one stakeholder of each factor should be involved in the project. From the 24 meetings that were observed, the involvement of the different types of stakeholders is (Appendix O);

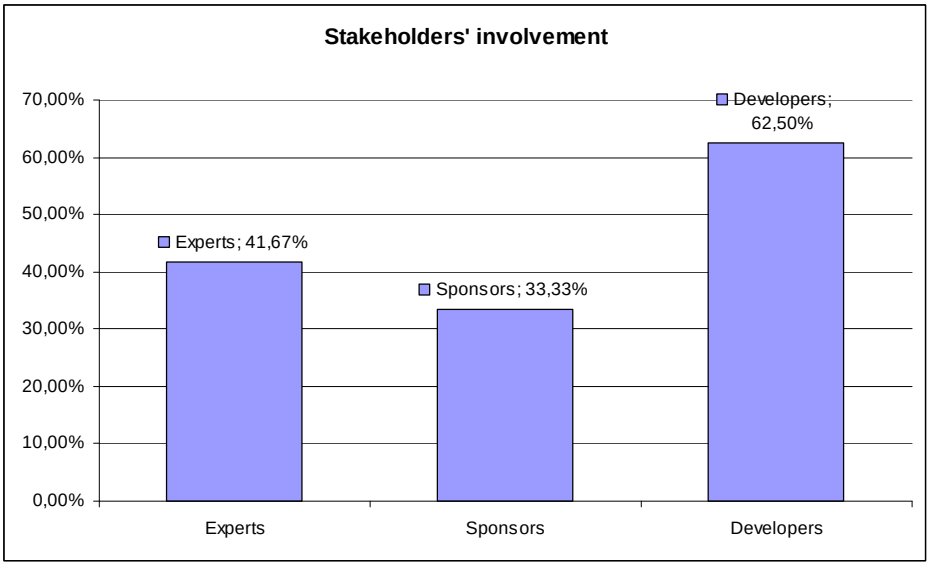


Figure 9: Involvement of different types of stakeholders

As can be seen from Figure 9, all the types of stakeholders are involved in the project. However, this does not fully prove that there is a relationship between the selection factors and the stakeholders in the project because it could also be coincidence that in this project all three stakeholders are involved. To prove that there is a real relationship, multiple projects within Chubb need be observed. However, due to time limitations and authority reasons this was not possible within this research.

Gathering process, techniques and methods
In the iClose and ARCH project the gathering of requirements is done in multiple ways. Techniques that are found during the research are; market research, existing systems, workshops, case scenarios and interviews in terms of face-to-face meetings. Each aspect is further elaborated below starting with the market research.

Market research
One of the first steps for involving market-facing technology in the project is conducting a market research. Several experts of the iClose project mentioned that a market research was done [ICPM1, CFO1, ITM1]. For example the Project Manager stated that they did a market research in order to understand what the customer wants;

"*We also did some market research. So we said to the brokers. What do you think you want? You want this or this. If you have a choice you will get A or B? What is the most important thing for you? How would this fit in your process? So we have done a lot of that*" [ICPM1].

However, in the ARCH project they did not conducted a market research jet but as the Project Manager of the ARCH project suggested during the interview, this is because they did not started with the market-facing technology aspect. She mentioned that when they start with the market-facing technology aspect, they will do a market research;

"*I think the market research comes in when you want to add more addition functionality like the market-facing. But that is the second step*" [ARPM1].

As can be seen from the statements of the Project Managers, in both projects they conduct a market research in which they ask the market several questions to extract their requirements for the new system. A template that is used during the market research of the iClose project was found during the search on available documents. This template is attached in Appendix P.

Existing systems

Another method that Chubb use in market-facing technology projects to gather requirements is by looking at existing systems. These could be systems that are used by competitors as well as systems that are used internally [ITM1, BPSM1, CFO1]. The requirement engineer analyzes these systems and documents what should be considered in the new system. The reason to look at competitor systems, suggested by the IT Manager of the iClose project, is to identify critical success factors through analyzing the indistinctness's of their system;

"*We would look at the competitor to see what they have done, because we don't want to make the same mistakes*" *[ITM1].*

Several documents are found that contain screenshots of systems from competitors, which is in line with the findings from the interviews. Some of these screenshots are attached in Appendix Q.

However, in the ARCH project, the requirements are not gathered through existing systems of competitors but through systems that are used internally [CFO1, ITM2]. For example the CFO, who is one of the sponsor stakeholders in the project, stated that;

"*For the ARCH project it is done through the existing kaizen tools. By analyzing the kaizen tools you are doing the requirements gathering*" *[CFO1].*

Screenshots of these kaizen tools are made and passed through to the developers. Based on these screenshots they understand what functionality has to be in the new system. From this it can be concluded that existing systems within Chubb as well as existing systems from competitors are analyzed in order to the gather requirements.

Workshops

Multiple interviewees from the iClose project stated that they using workshops to extract requirements for the system under development [ICPM1, ITM1, BPSM1]. According to the Project Manager; during these workshops they have a round-table discussion with representatives from the business and IT. In these discussions they brainstorm and use process maps to elaborate their ideas into requirements. The Business Analyst then documents these requirements into a business specification document. However, none of the interviewees of the ARCH project stated that they did a workshop to gather requirements. A possible reason for this could that they did not jet started with building the market-facing technology in the project. On the other side the Business Partner Service Manager, who is involved in multiple market-facing project around the globe at Chubb, stated the following;

"*For the stakeholders' involvement it is all about creating relationships to those people to understanding their business. In that case it is more about; just talking to them, starting of with a workshop and from that workshop making sure you engage all the stakeholders regularly. Market-facing technology, in the example*[†] *that I gave earlier, it was more about Chubb providing something to the market that was easy to use, very easy to use and very quick and efficient. So it was not only about going to the market, going to the brokers and asking; what do you want? It was more about thinking what are we going to launch to the market and what are the objectives. So the requirements were still performed internally and then we came into testing it*" *[BPSM1].*

From his experience and the experience from experts in the iClose project it can be concluded that workshops are used in the market-facing projects within Chubb.

[†] The example that BPSM1 gave is about a market-facing project that is called Peoplesure and was developed in the United Kingdom.

Case scenarios

A technique that is used in the ARCH project to gather requirements is by the use of case scenarios. According to the Project Manager of the ARCH project, they created case scenarios based on the documents that were generated by the existing systems. She stated the following;

"*For documents we do case scenarios, we are getting examples of documents that they already issued because they bring out additional issues that the generator might not think of. So you say; give me your most complex policy and then you look through that and you will see exceptions*". *[ARPM1]*

However, no other sources suggested in the ARCH that case scenarios are used and neither in the iClose project. Probably case scenarios are used but not explicitly, for example during workshops. Therefore, this is also considered as a gathering technique that is probably used in Chubb during the requirements engineering in market-facing technology projects.

Face-to-face meetings

The final requirements gathering method that was found during the research is face-to-face meetings. In face-to-face meetings, the requirement engineer sits down with a specific stakeholder that has a certain expertise and tries to extract the expertise from the specific stakeholder and transforms this into requirements. In the iClose project, the Project Manager mentioned that someone from the business will elaborate the business requirements and the Business Analyst will write this down (see below), which means that they are sitting together to elaborate the requirements.

"It is more of a back and forward process where someone from the business will elaborate the business requirements. The Business Analysts will write it down put it into a document and than back and forward, so it will get refined". [ICPM1]

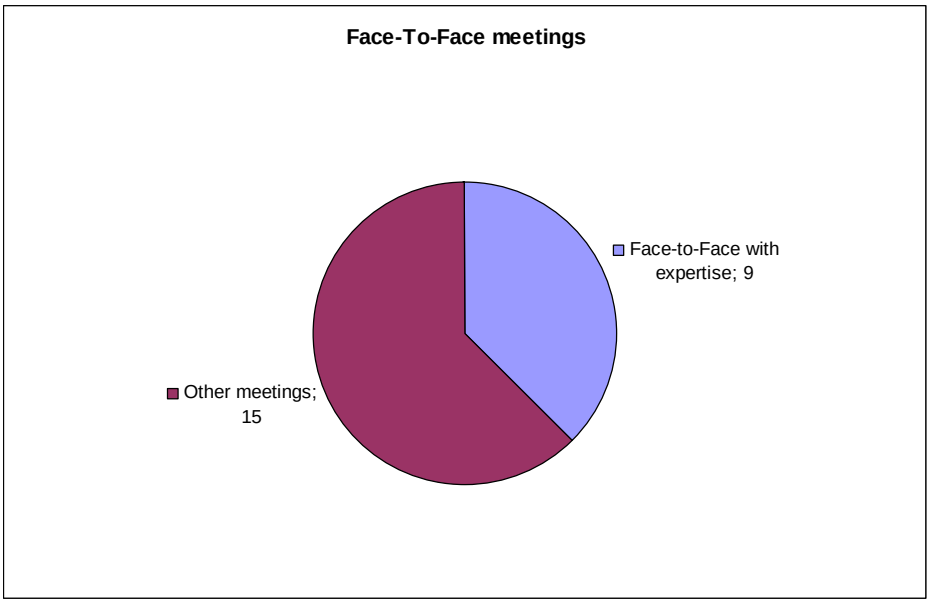


Figure 10: Observed amount of face-to-face meetings

As depicted in the Figure 10, from the observation period, there were 9 of the 24 meetings observed in the iClose project (Appendix R) wherein a business specialist sat down with the Business Analyst or Project Manager to explain and elaborate the requirements. From these meetings, several notes are made. Some of these notes are attached in Appendix S.

Besides the iClose project, the Project Manager of the ARCH project also mentioned that they are using face-to-face (interviews) meetings to gather requirements from experts [ARPM1]. From this it can be concluded that face-to-face meetings are held to gather requirements in market-facing project that involve multiple stakeholders.

Documentation
During the gathering process, in both projects, the requirements that are elicited are documented in a business specification document. Several experts mentioned this in the interviews [ICPM1, ARPM1, CFO1]. These statements are justified with the findings from available documentation in both project and the direct observation in the iClose project. From the available documentation, both projects used a business specification document (Appendix M & N[‡])

In the observation period in the iClose project, several (24) requirements concerns were addressed. These requirements concerns were discussed during meetings and from these discussions the requirements were derived and stored in the business specification document. I.e. as can be seen in the observation analysis, the requirements concern "APRA/SIC/Occupation list" is addressed multiple times. This means that this concerns is discussed in multiple meetings and from these discussions, the result is documented in the business specification document. In the beginning they wanted to use a SIC list consisting of +/- 1000 options and mapping it to ARPA codes. After several discussions they found that the ARPA mapping is not required and that there are to many SIC codes which is opposed to the user friendliness. Therefore, they decided to use an occupation category list that has a limited amount of options to choose from. This list is finally stored in the business specification document and passed through to the developer as a requirement that this functionality has to be in the system (see Appendix M). Further on, no other documentation or templates were found that are used during the requirements gathering process.

Conclusion
In this paragraph, the research questions SRQ 2.4 and SRQ 2.5 are answered based on the findings presented in the previous subsections. Starting with question SRQ 2.4:

Findings from question SRQ 2.4: "How are requirements gathered?"
In summary, before gathering the requirements, stakeholders are selected. To select these stakeholders for the project, the requirements engineer considers three aspects, namely;

- Expertise of the product. Based on the expertise of a stakeholder on a product requirements can be drawn, validated and prioritized.
- Sponsors. People who finance the project are considered in the selection of the stakeholder because they are the persons who define the ultimate goals of the project.
- Deliverables. Based on what needs to be delivered, the people who are able to build it need to be involved in the project. Therefore, deliverables is one of the concerns that is addressed in selecting the stakeholders.

Once the stakeholders are defined and the scope of the project is clear, the requirement engineer starts gathering the requirements. In this process he or she uses the following techniques; doing a market research, analyzing existing systems, organizing workshops, making case scenarios and conducting face-to-face meetings to extract requirements from experts. When the requirements are gathered, they are documented in a business specification document, which is passed through to the developer.

Findings from questions SRQ 2.5: "How many requirements are collected?"
During the observation period in the iClose project it turned out that the exact number of requirements is very hard to measure because the requirements are defined in multiple ways. For example, the requirements are defined through images, tables and examples wherein multiple requirements are addressed. On the other side, during the observations it was possible to measure the requirements concerns that were addressed in meetings. This measurement can also give an indication about the size of the project, if there are other numbers of requirement concerns available from other projects. However, due to that the iClose project is the only project where the number of concerns is available; it was not possible to determine the size of the project in terms of the number of requirements. Therefore, more research is needed within Chubb. The number of requirements concerns that were addressed during the observation period is 24. Due to that the observation is only done in a limited time frame of the project and not during the whole development of the project, it could be that this number not includes all concerns that are addressed.

[‡] Due to authority restrictions, the researcher was only able to retrieve a concealed document of both projects

3.2.3 Current situation of requirements negotiation in MFT projects

Unfortunately, not all requirements are inline and could conflict with each other. During the observations in the iClose project, there were 2 of the 22 requirement concerns that are outliners (Appendix G) in terms of iterations compared to the other concerns, this is depicted in Figure 11 and for simplicity duplicated below.

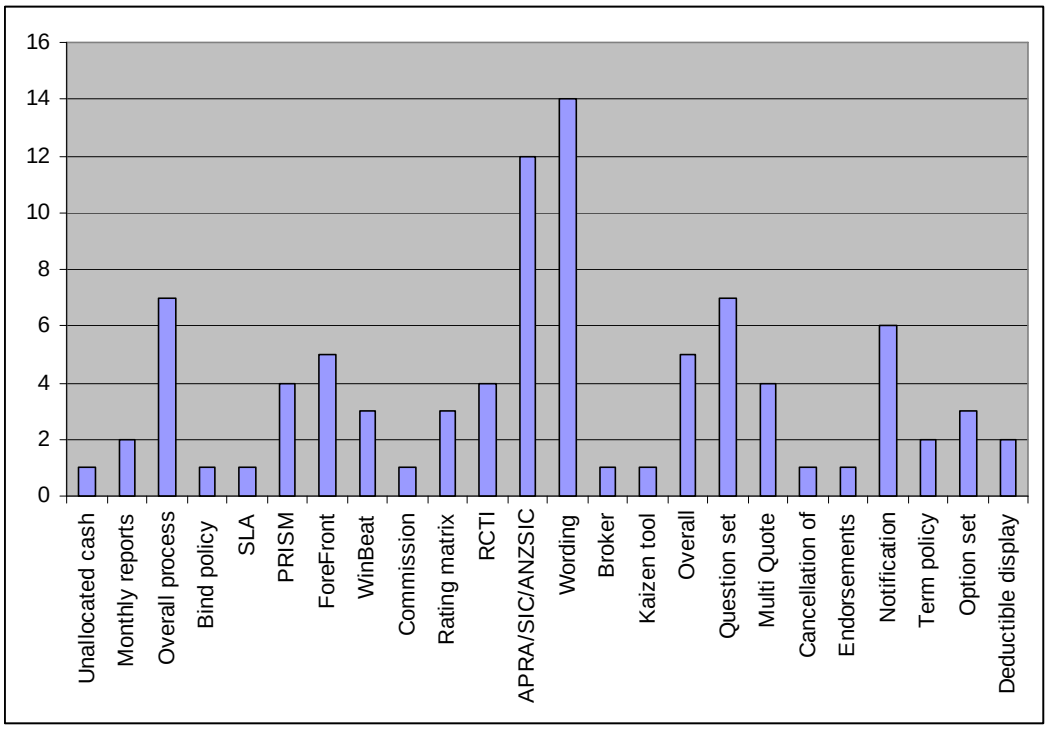


Figure 11: Number of iterations per requirement concern

From this finding, there is a high probability that conflicts occurred in these requirement concerns. In order to solve these conflicts, stakeholders have to negotiate with each other to get to an overall agreement about what they want and what they need. According to the requirement experts, this is done through several methods, where the overall process that incorporates these methods is to solve the conflicts is as follows;
Basically, there are three factors addressed when solving conflicting requirements in market-facing technology projects within Chubb, namely; scope, quality and time & cost. The first thing that is addressed is the Scope. If a conflict occurs in the requirements engineering, the requirement engineer will look at the scope of the project to see if it the requirements are inline with that;

"*You go back to scope. When you pull back to scope, it usually works. In terms of conflicting requirements*" [ICPM1]

If it is not, then the requirements will be dropped. If it is, then the quality of the requirements will be assessed. The requirement engineer will validate the quality of the conflicting requirements by making everyone aware of the essence of the conflict and based on that a decision will be made;

"*Basically you are hooking at the triple constrained and try to negotiate with the stakeholders to explain the position. In IT we can't tell the business what they should or shouldn't have. The business is always the driver of the development process.*" [ITM1]

The third factor that influence the negotiation process is time and cost. The conflicting requirements are measured on how much time it will take to realize it and how much it will cost. Based on that, the requirement engineer decides to extend the time of the project or the costs in order to incorporate these requirements. Several sources suggested that these factors are the main drivers in the

negotiation process on requirements engineering [ICPM1, ITM1, ARPM1]. I.e. as the Project Manager of the iClose project suggested;

"*In a project you got three things you can manage and that is the time frame, scope or requirements and the amount of money you spent on the project. The easy way to negotiate out these requirements is like; you can put that requirement into the systems, by either increase how much you spent and put more resources in the project or you need to let the same amount on the team and extend the time frame. So we can do that, but it will take longer or cost more or we can reduce the requirement. There is a sort of area matrix that you can work through because usually you can't say we can do more with the same amount of money or we can do that with the same amount of time. You need to be able to manage it like that. That is a very good way.*" [ICPM1]

To establish a good negotiation, all the stakeholders that are involved have to understand the requirements. Fortunately, the requirement engineers in market-facing projects at Chubb are aware of this. For example, the IT Manager stated that this avoids scope creep and makes sure that all the stakeholders know what they are talking about:

"*It is obviously very important to for the stakeholders to understand the requirements. And I guess from another perspective, it avoids surprises and scope creep. Having the stakeholder to understand what we are going to deliver, makes sure that everyone is on the same page and there are no surprises.*" [ITM1]

Therefore, they use several techniques to create an understanding of the requirements among the stakeholders. According to multiple sources [ITM1, ICPM, BPSM1, ARPM], these techniques are;

- Visualisation is used to demonstrate how things work or look like. The requirement engineers create process maps to demonstrate the processes that the system under development takes care of (see Appendix A) and they create images about how the system under development is going to look like to demonstrate the functionality.

- Face-to-face meetings to demonstrate the system. In both projects, these meetings are held to allow the stakeholders to see how the system is going to work. Due to this, the stakeholders will better understand the purpose of the system. Several sources stated that these meetings are held in market-facing projects;

"*It is really a kind of demonstrating the system, come with a prototype or document in the requirements and you kind of give a face-to-face explanation*" [BPSM1]

"*Now the videoconferencing allows us to demonstrate the system. They can actually see the person that is speaking and this helps a lot. You got a face-to-face meeting*" [ITM2]

- The use of a collaboration system. This system is only used in the ARCH project but several sources from the iClose project suggested that the use of a collaboration system would be useful in the requirements engineering process;

"*Although I know it is very useful, it could be fantastic, especially at Chubb, because we are multi-sided*." [ICPM1]

"*To be honest we haven't. That is generally not our mode. But on the other side, it can be fruitful because we are widespread and there are a lot of tools that will allow us to collaborate effectively*." [ITM1]

In the ARCH project the requirement engineers and the developers are communicating through a collaboration system that is called SharePoint. In this system, the requirements from the requirement engineers are passed through to the developers and discussed. This allows them to understand what is going on, as the Project Manager of the ARCH project mentioned during the interview;

"*It is a place to store the documentation and then there is a section where you can record an issue and you can respond to it, so you can see the trail of communications. We are trying to use that*

instead of email, because a lot of time in the email a lot of things get lost or not everyone is copied in." [ARPM1]

However, the Project Manager also mentioned that depending on the level of involvement of the stakeholder it is not necessary that all stakeholders have to know exactly what is going on. According to her, some stakeholders do not have to understand the requirements at a very detailed level but they have to understand the key issues [ARPM1]. The reason for is that some stakeholders do not want to be aware of all the details in the development of the project. Their biggest concern is to make sure that the goal of the project will be achieved, i.e. the sponsors and business experts of the project. For example in the iClose project (Appendix O), there are 5 stakeholders of the 11 that participated in less than 5 meetings (Figure 12). These stakeholders probably do not care about all the requirements in details because they are not actively involved in the development of the project.

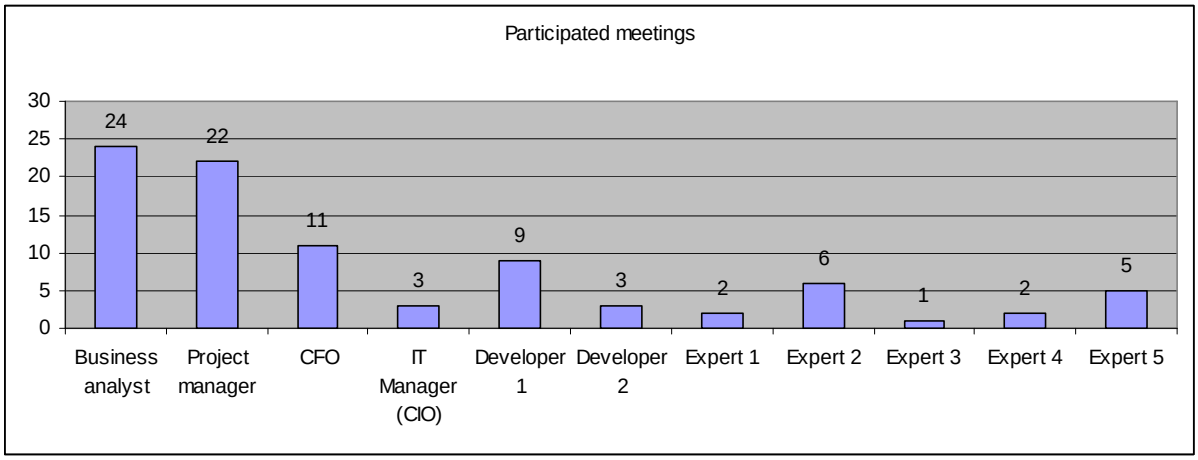


Figure 12: Amount of participated meetings per stakeholder

Although there are several techniques used for improving the understanding, sometimes it happens that the developers do not fully understand the requirements. The Project Manager of the ARCH project stated this occurs and therefore they have to be explicit in formulating the requirements;

"*Yes the developers don't understand the insurance business and that is why the requirements are so important. What we are trying to do is to tell them from a business perspective why something is important*." [ARPM1]

From the observations in the iClose project, the amounts of iterations per requirement concern discussed with the developer and the business are recorded and analyzed (Appendix S). From this analysis, there were no outliners between the business (stakeholders) and the developers among any of the requirement concerns. As can be seen in Figure 13, the number of iterations of developer 1 and 2 are in line with the iterations of the rest of the business (stakeholders). This means that the developers did not have any extra meetings with the Project Manager or Business Analyst to discuss specific requirement concerns, which could indicate that they understood the requirements that were presented by the business. However, due to that not all the communication is done through meetings, it could be that the developers discussed requirement concerns with the Project Manager through an alternative communication channels. For example; through a direct phone call.

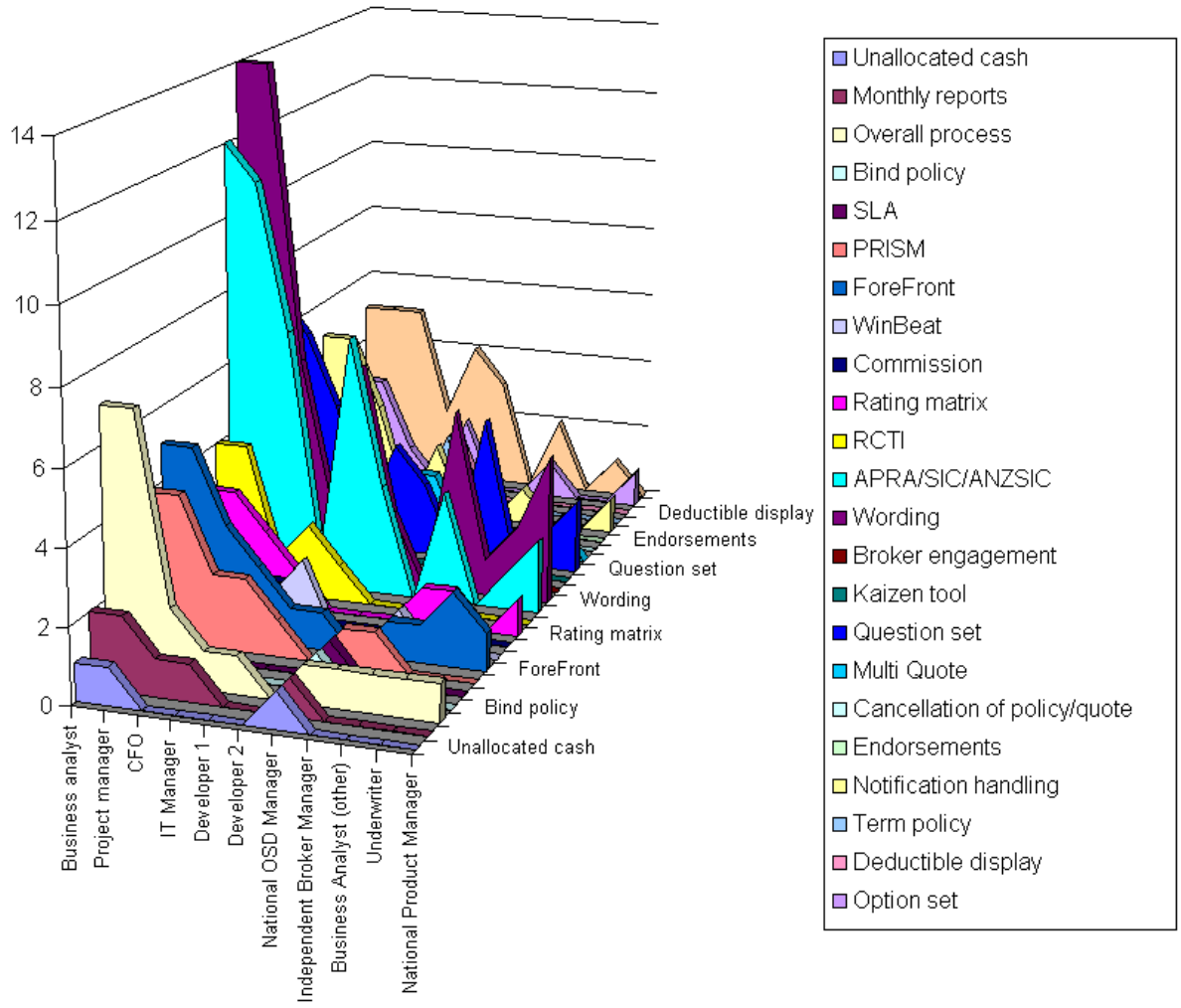


Figure 13: The number of iterations per stakeholder per requirement concern

Conclusion

Reflecting the findings from the case study back to the research questions, during the observations in the iClose project, 11 stakeholders were counted that are involved during the requirements engineering. This answers the first research question about requirements negotiation: SRQ 2.6 “How many stakeholders are involved during the requirements engineering?” Besides this, also the amount of meetings wherein the stakeholders participated was also counted. From this, there are several stakeholders that participated in the meetings less than 5 times and therefore they probably do not understand all the requirements in detail as also is suggested by the Project Manager of the ARCH project. However, for this, more research is required but due to time limitations, this was not possible to carry out in this project.

Further on, also the amount of requirement concerns and the quantity of the times that these concerns are addressed during the meetings are counted in the iClose project in the observation period. There were 22 requirement concerns addressed during the meetings and from these concerns, 2 of them were outliners that were addressed more than 10 times. A reason for this could be that there were possible conflicts and therefore these concerns were addressed many times in order to get to a solution. Also from the notes that were made during the observation, there seems to be some conflicts in these requirement concerns. Reflecting this to the research question SRQ 2.7 “How many conflicting demands occurred during the requirements engineering?” of this case study is 2.

According to several sources, the factors that are addressed to solve these conflicts are (research question SRQ 2.8);

1) Reflecting the conflicting concerns to the scope of the project to determine if it is inline with the scope
2) Validate the quality of the requirements that conflicts by making everyone aware of the essence of the conflict and based on that a decision will be made.
3) Measure the conflicting requirements on how much time it will take to realize it and how much it will cost, based on that it will be decided to extend the time frame of the project or the cost of the project to incorporate and solve these conflicting requirements.

These three factors are addressed during a negotiation process with the stakeholders. In order to start this negotiation process the stakeholders have to understand the requirements. Therefore, the requirement engineers are using several techniques to communicate it to the stakeholders (research question SRQ 2.9 & SRQ 2.10). The techniques that they are using are;

1) Visualisation, like process maps and images to demonstrate the functionality and how the system is going to look like.
2) Face-to-face meetings, were the requirement engineer explains in a face-to-face meeting how the system is going to work.
3) Collaboration systems, that stores the trail of communications in order to increase the understanding of the stakeholders about how and why decisions are made.

Further on, the Project Manager of the ARCH project suggested that the requirements are not always understood by the developers. However, this is not in line with the findings from the observations in the iClose project. The findings from the analysis of the observation in the iClose project did not show any deviations in the iterations per requirements concern between the business and the developers. This could mean that the developer understands the requirements or that this is communicated through other communication channels than normal meetings.

3.2.4 Current situation of requirements validation in MFT projects

The elicited requirements need to be validated in order to make sure that they are correct. Requirement engineers at Chubb are using several techniques or methods to do this. This paragraph describes the validation techniques that are used in the market-facing projects within Chubb, starting with the first, namely; multiple sources.

Multiple sources

The most frequent used method within market-facing projects at Chubb to validate requirements is the use of multiple sources. During the requirements engineering process the requirement engineer uses multiple sources to check if the elicited requirements are right. Multiple requirements experts mentioned this technique during the interviews, although mostly in different fashions. For example the CFO and the Project Manager of the ARCH project gave an example about that the elicited requirements from the stakeholders are validated through the existing systems and documents;

"*The IT Manager did the mapping of the information from the project that is required for Prism. You want be able to get that booking, unless you have that information, so when the underwriters coming up with the requirements, the IT Manager will cross this back to Prism to make sure it is complete.*" [CFO1]

"*What I do is; I go to the people who actually use the system today and ask: how do you do that and what do you actually see? I look at data in the system to see how it actually sits up there to validate about what is going on. It is getting the information from different sources just to validate.*" [ARPM1]

As the Project Manager already suggested, it is about getting the information from different sources. Besides documents and existing systems, the Business Analyst in the ARCH project also gives another example. She mentioned that she is going to multiple experts to verify if the elicited requirement is correct;

"*We decided not to go to one people but multiple. You have to go to two persons that know everything.*" [ARBA1]

These findings are inline with the findings from the observations. During the observations in the iClose project, the Business Analyst and the Project Manager organized multiple meetings with experts to

verify requirements, for example; (Appendix U) observation no. “13 – Meeting about the question set with the Independent Broker Manager” and observation no. “24 – iClose outstanding issues” where in both meetings the question set was discussed with other stakeholders. During the meeting in observation no. 13 the question set was validated with the Independent Broker Manager and during the meeting in observation no. 24 this questions set was verified by other experts, in this case the CFO and National Product Manager.

Sign off
Another technique that is mentioned by several requirement engineering experts is to sign off the requirement by the stakeholders [ITM1, ICPM1, BPSM1]. Once the requirement engineer elicited the requirement from the stakeholders, they go back to the stakeholders or their manager and asked them to sign it off. If they sign off, the specific stakeholder will take the responsibility that according to him or her, the requirements are correct and that the requirement engineer can pass these requirements through to the developer. Most of the time, the specific stakeholder then realizes that he or she have to be sure that the requirements have to be correct and therefore he or she will often involve other stakeholders to verify these requirements;

“*Because you can get all the requirements from someone that can’t sign off and he sends you to Rodney and Rodney goes: Oh these requirements are not right, I can’t sign this off. And you think you collected 80% of them, so then you are in the wrong hole. So the way you setup these questions is very useful. It sort of works you through the process.*” [ICPM1]

In the business specification document (Appendix M and Appendix N), a final chapter that is called “Sign off” or “Approval” is included wherein several stakeholders have to sign off the document. When they sign off, they all agree that the business requirements presented in the document are correct. The Business Partner Service Manager, who is involved in multiple market-facing projects around the world within Chubb, mentioned that this technique is generally used in software development projects within Chubb;

“*A lot of this comes down to a good Business Analyst skill, experience, but also making sure that the person who has given you the requirements has the accountability to sign it off. In other words if they changes their mind all the time, then you get to a state in the project where you say; right this is the requirements, this is writing down in black in white, sign that off.*” [BPSM1]

Changing requirements
However, some requirements are constantly changing, and therefore, these requirements are hard to validate. Once when the requirement is elicited and validated it can be that the requirement is already changed. According to several requirement experts, this concern should be addressed during the requirements gathering. During the gathering of the requirements the requirement engineer must validate with the source if requirements that are gathered can change over time [ICPM1,ARBA1]. If these requirements change over time, and decisions about these requirements are made later on, then, according to the Business Partner Service Manager, this will be considered in the Agile development where in every iteration the requirement will be verified if it is changed or not;

“*really you need an IT team that is kind of prepare for that and that is what Agile does, because you can have a set of a kind of changes that roll in through each iteration*” [BPSM1]

An example of this occurred during the observation of the iClose project. In several meetings, the wording issue was addressed because the responsible stakeholder mentioned that the wordings will change over time. Therefore, the Project Manager validated the wording requirements in several iterations to make sure that they are still correct and flexible enough for future changes;

“*Like now, we are trying to build the system and we are using the current wording policy. We know that within three weeks they will change the wording policy and we know that they do it again three months after that*.” [ICPM1]

Conclusion
In summary, the answer of the question that concerns the validation of requirements “SRQ 2.11: How are requirements tested/validated?” is that in market-facing projects two major techniques are used namely the validation by the use of multiple sources and the sign off principle. By the use of multiple

sources, the requirements engineer consults multiple sources to elicit the requirements. These sources include; experts, documentation and existing systems. In terms of the sign off principle, the requirement engineer asked the stakeholder to sign off the requirements. When the stakeholder signed off the requirement, he agreed that these are the final requirements and that these requirements can be passed through to the developer in order to be realized.

However, some requirements are constantly changing and these are hard to validate if they are not gathered properly. In Chubb, the requirement engineers asked the stakeholders if the requirements, that they suggest, will change over time. If this is the case, then the requirement engineer validates these requirements during the iterations in order to make sure that the concerning requirements are still correct and flexible enough to address changes.

3.2.5 Current situation of requirements prioritization in MFT projects

Unfortunately, often not all elicited requirements can be realized in the system under development due to time and cost constraints. Therefore, requirements have to be prioritized to determine which requirements are crucial to be in the system and which are not. At Chubb, this is also done in the requirements engineering process in market-facing technology projects. In this process, the requirement engineers are mainly using two aspects to determine which requirements are crucial and which are not. These two aspects are; scope of the project and return of investment.

Scope

One of the initial steps in setting up a project in Chubb is determining the scope of the project, wherein the goal and the basic elements are defined (see Appendix V). This scope is more or less the red line through the project and is used to determine which requirements have to be in the system and which have to be left out. When a requirement is elicited, the requirement engineer analyzes it to verify if the requirement is inline with the scope of the project [ICPM1, ARBA1, ITM1, ARPM1]. If the requirement is inline with the scope of the project, then it will be taken into account. If not, the requirement will be dropped out of the project. The Business Analyst in the ARCH project gave an example of this during the interview;

"*You can delay on risk analysis, but that is for your information, (but rating and etc. is helping to calculate the premium) and thus that is the document that goes outside. You need to focus on the goal of the project. I just give you the very big example*" [ARBA1]

In this statement, she mentioned that rating is one of the elements of the scope of the project and the risk analysis is not. Therefore, the risk analysis document will not directly be taken into consideration in the project. This is also inline with what the Project Manager of the iClose project stated about how he prioritizes requirements;

"*In terms of prioritizing; you can go through a process of saying what the objectives are and what is the bare of requirements to fulfil that. What are the dependencies across that to work and how to sequence them and how do we actually prioritize the development and the delivery of it?*" [ICPM1]

In this statement, he mentioned that he is looking at the objectives of the project and based on that he determines if the requirements fulfil these objectives. These objectives are defined in the scope of the project. If the requirements do not fulfil these objectives, they would not be taken into consideration in the project.

Return on investment

Besides the scope of the project, the majority of the requirements engineers also suggested that they take the return on investment into consideration to determine the priority of the requirements [ARBA1, ARPM1, CFO1, ITM1, BPSM1]. If the considered requirement does not support the return on investment, it will gain a low priority or it will be dropped out of the project. As what the Project Manager of the ARCH project stated;

"*It is the same thing as, if you leave something out; you have to know what the impact is. What is the impact of the business by leaving it out? What does it cost to leave it outside of the system? That is what drives the priority.*" [ARPM1]

With this statement, she explained how she analyzes the requirements if it would not be in the system. From that perspective, she tries to find the return on investment and decides to include or exclude the requirements.

During the observation period in the iClose project, 31% of the observations addressed prioritization. In the meetings where requirements were prioritized, in 40% of the cases the accent of the prioritization was on the scope of the project and 60% on the return on investment (Appendix W). Nonetheless, most of the time both mechanisms were used but with one as the main prioritization mechanism and the other as a supporting prioritization mechanism. Figure 14 depicts the percentage of the meetings where in prioritization was done and in Figure 15 the percentage of the main prioritization mechanisms are depicted.

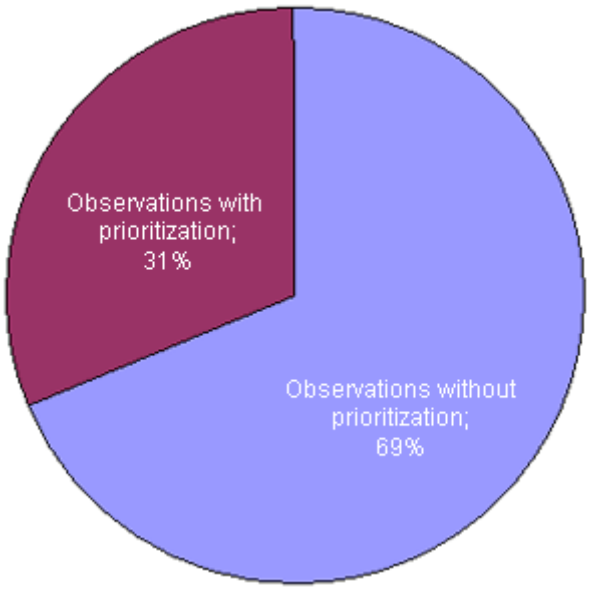


Figure 14: Prioritization during meetings

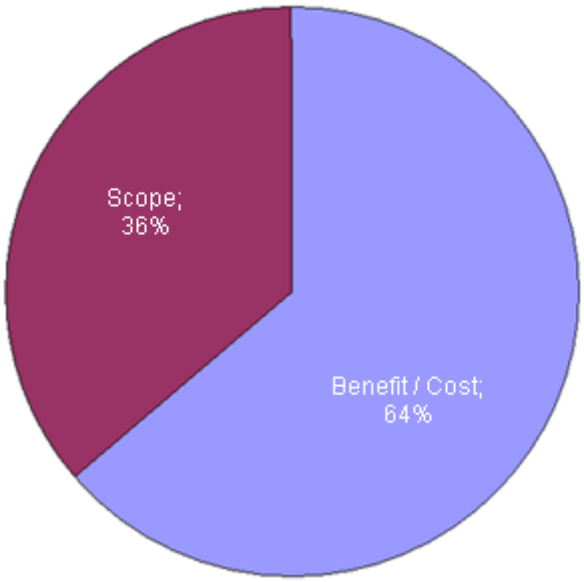


Figure 15: Applied prioritization mechanisms

Other aspects
However, return on investment and the scope of the project are not the only aspects that are considered in prioritizing requirements in market-facing projects. In the statement of the Project Manager of the ARCH project above, she explains that she is also trying to analyze the impact of the requirement in the business processes. Besides this, the IT Manager mentioned another technique that is used for prioritization. He stated that the complexity of the product is also considered as a factor to prioritize the requirement;

"*When you prioritize you looking at things like; is it going to make the product complex? Does one requirement has a greater return on investment?*" [ITM1]

If the requirement increases the complexity of the product, it will have a lower priority or probably dropped out of the project. If the requirement increases simplicity, it will be taken into consideration to realize it.

Requirement prioritization management techniques
Once the priorities of the requirements are clear, they need to be organized and documented. The Project Manager in the iClose project organizes and documents the requirements into three categories, namely; critical requirements that must be in the system, requirements that should be in the system but is not strictly necessary in order to functioning properly and requirements that are not necessary at all but would be nice to have in the system, like requirements on the layout;

"*You can just say; you tell me what you are going to do: look and feel or functionality? They usually choose functionality. And what you than can do is once the systems start working and when you get money coming in we can use some of the money to enhance it so we prioritize the requirements in*

release 1 2 and 3. Critical requirements come into release one. Need to have but not critical comes into release 2 and 3 is all the nice things." [ICPM1]

However, no other documents or sources suggested this organizing technique. A possible explanation for this could be that the Project Manager is the only person who is organizing this and therefore the existing documents and other sources of evidence are possibly only stored on his personal account. Due to restrictions, it was not possible to retrieve these documents.

In the ARCH project, another technique is used to organize the requirements. According to the Project Manager, once the priorities of the requirements are clear, they are organized in a priority list wherein requirements are prioritized on a scale from one to twenty;

"*No, we have various priority lists. Some of them are saying which insurance products you are going to put in. All the insurance products here in Australia are written and prioritized from one to twenty. And then we have additional functionality they are trying to put into the system, and again that is prioritized based on the business need. We work trough those lists and then it is a priority list that goes into a plan.*" [ARPM1]

Similar to the previous mentioned technique, the Project Manager of the ARCH project is the only source that mentioned this. A possible explanation for this, could be the same reason as the previous mentioned technique, namely that the Project Manager is the only person who manage this and therefore no other sources of evidence exists.

Conclusion
In summary, prioritization (SRQ 2.12) is mainly done based on two aspects namely; Scope of the project and return on investment. For example; when a requirement deliver a high return on investment and is inline with the goal and scope of the project, it will gain a high priority. When a requirement has a low return on investment and is not inline with the goal and scope of the project, it will gain a low priority or possibly dropped out of the project. There are also other aspects that are used to prioritize, but these aspects are mostly used in combination with the scope of the project and / or the return on investment. These aspects are; the impact of the requirement in the business processes and the complexity of it.

The organization and documentation of these prioritized requirements are done in different ways. In the iClose project, the organization and documentation is done based on three categories. After prioritizing, the requirements are categorized into; 1) requirements that are critical and must be in the system. 2) Requirements that should be in the system but are not critical. 3) Requirements that would be nice to have it in the system. In the ARCH project, another method is used. In this project, the requirements are listed and classified on a scale from one to twenty.

3.3 Conclusion – Answer to Research Question 2

In conclusion, the second research question, namely: "How does Chubb currently gather, negotiate, validate and prioritize the requirements of projects like the iClose project" of this thesis is answered.

At the moment, the requirements engineering for market-facing projects within Chubb is done based on an iterative approach were requirement concerns are assessed multiple times in order to form the right requirements. In this approach they demonstrate prototypes wherein they negotiate prioritize and validate the requirements. By this, the requirement engineering and development is done simultaneously and therefore, the requirements gathering, negotiation, validation and prioritization are also done simultaneously. During this process, mainly one overall document is used wherein all the requirements, business logic and scope and goals are defined. This document is called a business specification document.

To gather requirements, the requirements engineers have to choose stakeholders to gain information. Therefore, the requirements engineers consider three aspects to select stakeholders, these aspects are; expertise, sponsorship and delivery responsibilities. Once the stakeholders are clear and the scope of the project is defined, the requirement engineer starts gathering the requirements and therefore he/she uses the following techniques; market-research, existing systems, workshops, case scenarios and face-to-face meetings (interviews).

During the negotiation process, the stakeholders have to understand each others concerns. To gain an overall understanding, the requirement engineer uses techniques. These techniques are; visualisation (process maps and images), face-to-face meetings to explain the system and collaboration systems to discuss the status quo of the project and to register the communication. If there are conflicting requirements, the requirement engineer tries to reflect the conflicting concerns to the scope of the project to determine if it is inline with it. Besides this, he or she also validate the quality of the requirement that conflicts, by making everyone aware of the essence of the conflict and based on their opinions the requirement engineer will make a decision. Further on, the requirement engineer also measures the conflicting requirement on how much time it will take to realize it and how much it will cost. Based on this, he or she will make a decision to extend the time frame or to increase the budget of the project in order to solve the conflict.

After the requirements are gathered, they have to be validated. The validation of the requirements in market-facing projects is mainly done due to two techniques, namely; the use multiple sources to validate the requirements and a sign off technique. The sign off technique is used to make sure that the stakeholder, who is responsible of specific requirement concerns, agreed on the formulated requirements from those concerns.
Finally, the prioritization of the requirements is done based on the scope of the project and the return on investment. When a requirement delivers a high return on investment and it is in the scope of the project, the requirement will gain a high priority. When the requirement delivers a low return on investment and is not inline with the scope, it will gain a low priority or it will be dropped out of the project. The next chapter describes the assessment of these findings from the case study by the findings from the literature study and based on that recommendations are drawn.

4. Recommendations on the current situation

Findings from the case study and literature study are analyzed. Based on this analysis, recommendations for market-facing technology projects within Chubb are drawn and described in this section. The first section of this chapter describes the recommendations on the overall process, the second section describes recommendations on the gathering process, the third section on the negotiation process, the fourth section on the validation process, the fifth section on the prioritization process and in the final section an answer is given on the third research question by summarizing the findings from the previous sections.

4.1 Recommendations on the overall requirement engineering process

As mentioned in Section 1.2, a market-facing enterprise is an organization that is sensitive to the needs of the market and customers. These needs are constantly changing. In order to accommodate these needs with a system, the development has to be done rapidly and the highest satisfactions of the market and customer have to be achieved. From the findings of the literature study in Section 2.2, the most suitable development approach for this issue is an AGILE approach. In this approach small iterations are used wherein stakeholders have to work intensively with each other to achieve a high satisfaction on the needs. As the findings from the case study suggested, this approach is already used in the both projects. They use small iterations wherein multiple stakeholders are sitting together to elicit requirements and review current requirements. A drawback of this approach is that there is a chance that important requirements may go unrecognized or recognized too late and got not implemented [41]. However, during the observations in the iClose project there were no important requirements recognized too late. Due to the fact that the project was still in development after the observations, it could not be determined if requirements were not recognized at all.

An approach that can be used to address this gap, is the approach proposed by Sen and Hemachandra in [1]. In this approach, the goals of the stakeholders need to be extracted. Once these goals are extracted, they need to be decomposed from high level goals to lower level or sub goals. From this, the goals can be reviewed in order to see if any goals are missing. The approach revolves around 1 to 4 week cycles that are called Sprints (iteration). Each Sprint consists of three main activities, namely pre-sprint, the sprint it self and post-sprint which refer to the activities before, during and after the elicitation of goals. The information inputs and outputs of each activity are presented in Table 7.

Table 7: Inputs and outputs of activities [1]

Activity		Inputs	Outputs
Pre-Sprint	Examine	▸ Interview Transcripts ▸ Corporate Policies ▸ List of Requirements ▸ Work Flow Diagram ▸ Mission Statement	■ Initial List of Goals
	Count	▸ Stakeholder participating ▸ Initial List of Goals	■ Total number of Stakeholders ■ Total number of Initial Goals
	Data entry	▸ Total number of Stakeholders ▸ Total number of Initial Goals ▸ Name of Initial Goals	■ Software in Execution
	Distribute	▸ Initial List Of Goals to stakeholders	■ Goal Preference Model
	Data Entry	▸ Priority Value of a goal	■ First Compilation Table
Sprint 1…N	● Develop ● Wrap ● Review ● Adjust	▸ First Compilation Table ▸ Activity Card Computer Output-1 to Activity Cards Compiler Output-(N-1)	■ Activity Card ■ Activity card in tabular format ■ Activity Card Compiler Output-1 to Activity Card Compiler Output-N
Post-Sprint		Activity Card Compiler Output-N	Software Requirements Document

During the pre-sprint activity, information is gathered from the stakeholders and formulated into goals. These goals are presented into a first compilation table, shown in Table 8.

Table 8: Format of a First Compilation Table [1]

Name of Initial Goals (without predecessor)	**Priorities assigned by stakeholder** (Sh-1 to Sh-M)				**Priority Of Initial Goals by Sorting in Ascending Order of T.PV** $T.PD=\sum_{PV=1}^{PV=M}$
	Sh-1	**Sh-2**	**…..**	**Sh-M**	
A0	PV_{11}	PV_{12}		PV_{1m}	$T.PV_{a0}$
A1	PV_{21}	PV_{22}		PV_{2m}	$T.PV_{a1}$
A2	PV_{31}	PV_{32}		PV_{3m}	$T.PV_{a2}$
.. ..	.. ..	.. ..	.. ..	.. ..	
An	PV_{n1}	PV_{n2}		PV_{nm}	$T.PV_{ax}$

This table is used in the sprint activity to refine the goals including identifying sub goals and responsibilities. Then these goals are wrapped up into a goal tree and reviewed. In the review process, each goal is separately discussed (negotiated, validated and prioritized) among all the stakeholders and finally adjustments are made based on their feedback. In the post-sprint activity, the requirement engineers document the specifications based on the identified goals and sub goals [1]. However, Sen and Hemachandra only validated their approach on the correctness of the goals and did not verify the completeness of the goals.

4.2 Recommendations on the requirements gathering process

One of the techniques found in the case study to elicit requirements is conducting face-to-face meetings with stakeholders that are experts of certain business processes. These face-to-face meetings are similar to what is suggested by the literature as “interviewing experts”. This technique seems to be well suitable for requirements elicitation in new domains, like market-facing technology

within Chubb. Therefore, these meetings can be more elaborated to improve the overall requirements engineering process. For example cognitive tools can be used. If experts, that are involved in face-to-face meetings, are assessed on their cognitive profiles they would be able to better understand and prioritize the requirements [2]. Carod and Cechich proposed a method that use elicitation techniques based on the characteristics of the stakeholders in [2]. The characteristics wherein stakeholders could be classified are; sensing, intuitive, visual, verbal, active, reflective, sequential and global. Elicitation techniques that can be used based on the classification are; interviewing, questionnaires (in combination with other technique) and (visual) models. In order to assess the stakeholders, the Index of Learning Styles should be used (ILS) in form of a questionnaire [58]. Based on the result of ILS questionnaire, the elicitation technique can be chosen wherein the stakeholder feels the most comfortable with. Findings from their experiment demonstrated that 72,7% of the conflicts appeared when software requirements specifications were not inline with the preferences of the stakeholders. However, this was only tested in a controlled environment and does not suggest that this would also be the case in practice [2].

Besides this, also case scenarios can be used. Currently, this is already used in one of the projects at Chubb. By the use of case scenarios, the stakeholders would be able to understand and to express the environment wherein the new system is going to work. Through this, the requirement engineers can capture the system's purpose and the reason about whether a given design will meet that purpose [11]. Case scenarios are cases wherein the goals are highlighted due to the appearance and behavior of the (new) system. These cases explains what people are trying to do with the system, what procedures are adopted and carried out by the system and the interpretations are by the people about the system [59]

Another technique that could be used to speed up the current requirements validation and prioritization process in market-facing projects is to use standard templates on how a requirement should be stated. Fruhling et al. found in [20], that preparing a requirements' template and suggesting how requirements should be stated were important improvements to speed up the validation process. For example, a statement could be something like "The system should…." [20].

4.3 Recommendations on the requirements negotiation process

Wu et al. stated in [40] that during the negotiation, stakeholders have to collaboratively and incrementally find out what has to be build. During the case study, in both projects, stakeholders work together to determine what the requirements are by discussing requirement concerns multiple times. This means that in both projects the stakeholders are collaboratively and incrementally working on the requirements, which is inline with the statement of Wu et al. mentioned earlier. Wu et al. mentioned that the activities in this process are sharing knowledge, understanding each other, resolving conflict of interest, co-authoring software requirements [40]. In order to enhance these activities, a collaboration system can be used. During the case study it turned out that this is partly done. In the ARCH project a collaboration system is used for requirements engineering but no such system is used in the iClose project. To enhance these activities, the system that is proposed by Wu et al. in [40] could be used. This system is based on the win-win equilibrium theory which links win conditions, issues, options and agreements. Based on these aspects it creates a win-win equilibrium where all the win conditions are covered by agreements and all issues are resolved by options that are covered by agreements and it shows how this is reached. They collected data from projects over a two year period wherein this system is used. Findings from this research showed improvements in cost effectiveness, user satisfaction, and in combination with complementary practices, such as front-end teambuilding, site-visits, mutual learning and prototyping helped the project teams to achieve better outcomes. A drawback of collaboration systems is that non-experts could face some major challenges in understanding these systems [40].

Besides this, also consistency is important during the negotiation process. Especially in the development of systems that involves the knowledge from business experts. Often this knowledge has to be interpreted by multiple stakeholders and this can easily be misunderstood by them. In terms of the market-facing projects in Chubb it could be that some stakeholders do not fully understand the requirements or misinterpreted them. For example as the Project Manager of the ARCH project stated; that sometimes the developers do not fully understand the insurance business. Therefore, the developers can easily misinterpret the requirements from the business. In order to address this gap Jiang and Yang proposed a model in [38] that can be used to elicited requirements based on ontology.

However, this model has its limitations for the use in market-facing projects, it only addresses performance requirements and its purpose is to elicited requirements from customers. Nonetheless, the idea to use ontology for requirement engineering can contribute to reduce the issue of misinterpretations among the stakeholders. In an ontology, explicit formal descriptions are given to concepts in a particular domain [38]. By using these formal descriptions of concepts, everyone knows what the concepts mean, which reduces the gap of misunderstanding these concepts. For example in the iClose project, the concept “bind” can be described as: the process of transforming <computing> a quote <input> into a policy <output>.

4.4 Recommendations on the requirements validation process

During the case study it turned out that requirements are validated by multiple sources. Once the requirements are elicited, the requirement engineer verifies the requirements by multiple stakeholders, documentation and by information extracted from existing systems. However, this can cause some issues because stakeholders often formulate brief, ambiguous and conflicting requirements which are hard to validate. To address this issue, the proposed spiral model of Ahmad [3] can be used (Figure 6).

Figure 6: Requirements negotiation process

The first step in this model is to identify conflicts in the list of candidate requirements that are elicited by the stakeholders. Then alternative solutions that address these conflicts have to be developed. After that, the solutions have to be elaborated by the stakeholders to gain a better understanding. Next, they judge and trade off the requirements based on the defined criteria in the previous step. If all the conflicts are not resolved yet then it has to go trough the next iteration, starting by the first step until all conflicts are resolved and requirements are validated by all stakeholders.
Another method that is used to validate requirements in market-facing projects in Chubb is to sign off the requirements by the stakeholders once these are elicited and negotiated by other stakeholders. The stakeholder who has the responsibility of the requirement will need to sign off the final requirements. When he or she signed off the requirements, he or she agrees that these requirements are final and right. Often these stakeholders will do a final verification before they sign off. In order to improve this process the requirements engineer can use cognitive tools to determine the cognitive profiles of the stakeholders. By determining these profiles, the requirement engineer can adjust the presentation of the requirements. This will increase the understanding of the requirements and goals by the specific stakeholder which will speed up the sign off process. As mentioned earlier in paragraph 4.1.2, Carod and Cechich proposed a method that use elicitation techniques based on the characteristics of the stakeholders in [2]. Parts of this method can be used to improve the sign off process. Stakeholders can be classified in the characteristics (sensing, intuitive, visual, verbal, active, reflective, sequential and global) due to the questionnaire on [58]. Once the characteristics of the stakeholders are know, the requirements can be presented in a way that is most comfortable for them (visual or non-visual, sequential or global, etc.).

4.5 Recommendations on the requirements prioritization process

From the case study it turned out that requirements are prioritized through considering the scope of the project, the business value (return on investment), complexity and impact, where the scope and the business value are the dominant aspects. These findings are to a certain extent inline with the findings from Racheva et al. [4]. As mentioned earlier in paragraph 2.5.4, they did a case study on requirements prioritization methods in agile development approaches. From this case study they found that the aspects that are considered when making decisions on requirement priorities are: business value, effort estimation/size measurement, learning experience, input from the developers and external change. Further on, they also found that the project context has a significant impact on the prioritization criteria and that the business value is a dominating requirements criterion. In reflection of these findings with the findings from the case study at Chubb, in both case studies it turned out that "business value" is a dominating factor and some other aspects have similarities with each other. The aspect "complexity" found in the case study at Chubb is similar to the "effort estimation/size measurement" found by Racheva et al. [4] and the "impact" aspect found in the case study at Chubb is similar to the "external change" aspect found by Racheva et al. [4]. However, Racheva et al. [4] did not explicitly mentioned anything about the scope or goal of the project as a prioritization criteria. They mentioned a criterion called "negative value" in which they assessed the requirements on the necessity to support the main usage scenario. This is to some extend similar to the assessment of scope of the project found in the case study. Although the model that Racheva et al. yielded is a descriptive model, it can be used to optimize the prioritization process in market-facing technology projects at Chubb. Due to the fact that the model describes the prioritization process, it can be used as a guideline in order to make sure that all aspects are considered and the overall process is structurized which can reduce time. This model is depicted in Figure 16.

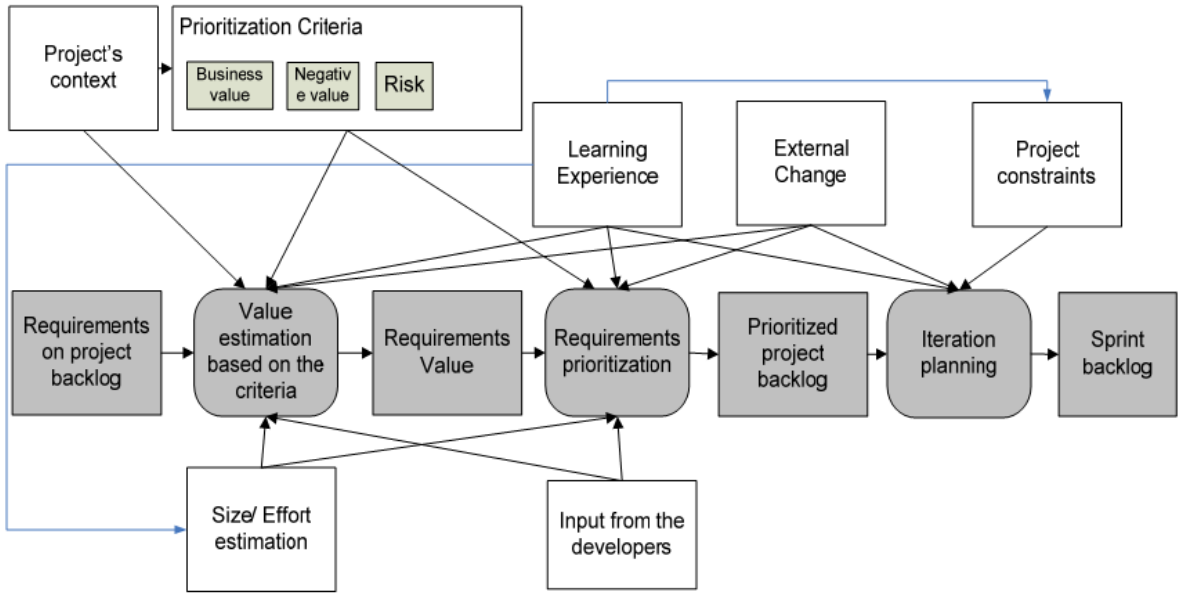


Figure 16: Conceptual model of [4]

The grey boxes in the model are the actual prioritization process and the white boxes around it are the aspects that influence the prioritization activities. The first activity is estimating the value of the requirements based on the criteria. In this activity all the aspects are considered. After estimating the value, the requirements get prioritized. Also in this activity all aspects are considered. Once the requirements are prioritized, the next iteration of the agile approach can be planned wherein the aspects "learning experience", "external change" and "project constraints" are considered. However, in this process the human subjectivity can varied the prioritization because people try to prioritize the requirements independently. To address this gap, the requirement engineer can also use the cumulative voting method described by Leffingwell and Widrig [5]. In this method, each stakeholder can assign in total 100 points to the requirements. These points can be distributed in any way that the stakeholder desires, where requirement with the highest amount is the most important and the lowest amount is the least important. After the stakeholders divided their point among the requirements, the results need to be combined and from that the relative preferences of the requirements can be extracted in a systematic and controlled manner.

4.6 Conclusion - Answer to Research Question 3

In this section, aspects that can be used to improve the requirement engineering process in market-facing projects within Chubb are analyzed and described based on the findings from the case study and the literature study. In relation to the research question 3: “What can be improved in the way Chubb gathers, negotiates, validates and prioritize requirements reviewing the current practice and comparing against scientific literature?”, these aspects answers this research question. In summery, the answer of this research question is;

Improvements on the overall approach

- Current approach can by improved by using the approach proposed by Sen and Hemachandra [1] to reduce the chance that important requirements are unrecognized or are recognized too late. However, this approach is not verified on the completeness of the goals.

Improvements in the gathering process

- Face-to-face meetings can be improved by the use of cognitive tools to gain a better understanding by the business expert about the system and the requirement concerns. Therefore, the method proposed by Carod and Cechich in [2] can be used. In this method, the characteristics of the stakeholders are determined. Based on the characteristics of the stakeholder, a elicitation technique can be chosen wherein the stakeholder feels the most comfortable with. However, this method is only tested in controlled environments.
- Case scenarios can be used in market-facing technology projects to capture the systems’ purpose and the reason about whether a given design will meet that purpose [11].
- Standard templates to state requirements could be used in the gathering process to speed up the requirements validation and prioritization process.

Improvements in the negotiation process

- Negotiation process can be improved by using a collaboration system in each market-facing project. For example the system proposed by Wu et al. in [40]. In [40] they stated that through this system they achieved improvements in cost effectiveness, user satisfaction and outcomes of the project. However, non-experts could face some challenges in understanding this system [40].

- Ontology can be used to reduce the issue of misinterpretations of requirements by stakeholders. In an ontology, explicit formal descriptions are given of concepts in a particular domain [38]. With these descriptions, the stakeholders will gain a better understanding of what is meant by specific concepts in the project.

Improvements in the validation process

- Requirements formulated by stakeholders can be brief, ambiguous and conflicting. By the use of multiple sources, these requirements can be hard to validate. In order to address this issue, the model proposed by Ahmad in [3] can be used. This model consists of multiple iterations wherein requirements are negotiated and validated by all stakeholders.
- By signing off the requirements, the requirement engineer could use the cognitive method proposed Carod and Cechich in [2] too. By presenting the requirements in a way where in the specific stakeholder feels most comfortable with, he of she will better understand what he or she need to sign off on. This reduces effort and time of understanding the requirements by the specific stakeholder.

Improvements in the prioritization process

- To reduce time in the prioritization process, the model of Racheva et al. in [4] can be used as a guideline. By using this model, the requirement engineer makes sure that he or she considers the important aspects in the prioritization process which reduces the chance of wrongly prioritized requirements.

- Sometimes requirements are prioritized by the requirement engineer independently and because of that, human subjectivity can disturb a proper prioritization. To address this issue, the cumulative voting method proposed by Leffingwell and Widrig in [5] can be used, where all stakeholders vote on the requirements to prioritize them. However, when using this technique, the requirement engineer needs to be aware of that all votes are not equal. For example the vote of the developer can have a higher influence then that of a business expert.

5. Recommended situation

In this chapter the Research Question 4: “What would be a better way for Chubb to carry out requirements engineering for market-facing projects?” is answered. The improved approach that Chubb can use to carry out requirement engineering in market-facing project is based on the findings of the case study in Section 3.2 and the recommendations made in Chapter 4. Elements from both sections are adopted in a framework that is given in Table 9. This framework is an elaboration on the framework of Sen and Hemachandran in [1]. As mentioned earlier in Section 2.2, this framework is an Agile framework that consist of small iterations. These iterations are called sprint and each revolves around a 1 to 4 week cycle. In each sprint there are three processes, namely pre-sprint, sprint and post-sprint that refers to the activities before, during and after elicitations of goals and requirements.

Pre-sprint
In the pre-sprint, initial information is examined through different techniques used in market-facing projects within Chubb. These techniques are to some extend elaborated by the suggestions from the literature. The techniques that can be used in this process are;
(i) Market research
(ii) Stakeholder selection based on expertise of product, sponsors and deliverables
(iii) Interviewing based on cognitive profiles
(iv) Existing system analysis
(v) Case scenarios
Market research can be done the gain important information about the current needs of the market. Stakeholders can be selected based on their expertise, sponsorship and responsibilities. Once stakeholders are identified, the stakeholders that have certain expertise can be interviewed to extract information from them. Before the interview, the specific stakeholder can be assessed on their cognitive characteristics and based on that, tools can be used to demonstrate the concerns. Existing systems can be analyzed to extract important information that is required by the business and case scenarios can be made to gain additional information. After the examination, an initial list of goals is produced and counted together with the total number of stakeholders. Then, this information can be transformed in a set of goals and compiled into a first compilation table (given in Table 10). After that, this table is stored in a collaboration system and distributed among the stakeholders. Information that is used in this process consists of; interview transcripts, corporate policies, workflow diagrams, mission statement, existing systems, market information, a default template to state requirements, stakeholders information and the initial list of goals compiled into a first compilation table.

Sprint
This process exists mainly of negotiating, validating and prioritizing the goals and requirements. Once the first compilation table is created and distributed, the goals and requirements in this table will further elaborated in the development, wrap, review and adjustment activities. The stakeholders discuss the initial goals and refine them into sub goals. To facilitate this discussion, an ontology that is described in paragraph 4.1.3 can be used to reduce misunderstandings. Notes can be stored in the collaboration system in order to understand later on why decisions are made. In terms of conflicting goals or requirements, the spiral model of [3] can be used to iteratively and collaboratively solve the conflicts. The goals can be validated through multiple sources and signed off (by presenting the requirements based on the cognitive characteristics) by the specific stakeholders that are responsible for the goals and requirements. Then every stakeholder can prioritize the goals and requirements individually by voting based on business value, negative value, size/effort estimation, input from developers, external change, learning experience, and project constraints. After this, the requirement engineer can merge the results and determine the overall priorities of the goals. Each main goal with their sub-goals is then transformed into an activity card (depicted in Figure 17). All these activities can be executed simultaneously.

Table 10: Format of a First Compilation Table [1]

Name of Initial Goals (without predecessor)	**Priorities assigned by stakeholder** (Sh-1 to Sh-M)				**Priority Of Initial Goals by Sorting in Ascending Order of T.PV** $T.PD=\sum_{PV=1}^{PV=M}$
	Sh-1	**Sh-2**	**…..**	**Sh-M**	
A0	PV*11*	PV*12*		PV*1m*	T.PV*a0*
A1	PV*21*	PV*22*		PV*2m*	T.PV*a1*
A2	PV*31*	PV*32*		PV*3m*	T.PV*a2*
.. ..	.. ..	.. ..	.. ..	.. ..	
An	PV*n1*	PV*n2*		PV*nm*	T.PV*ax*

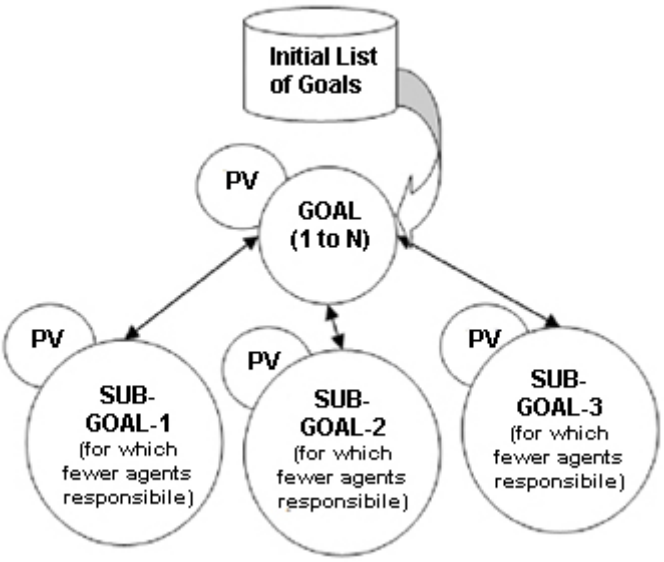


Figure 17: Goal activity card

Post-sprint

In this process, all the activity compiler cards are transformed into requirements and documented in the overall software requirements documentation that is already used within Chubb for market-facing projects. A template of this document is attached in Appendix N.

Table 9: Improved framework for requirements engineering based on [1]

Activity		Inputs	Outputs	Techniques
Pre-sprint	Examine	▸ Interview transcripts ▸ Corporate policies ▸ Workflow diagrams ▸ Mission statement ▸ Existing systems ▸ Market information ▸ Requirement template	■ Initial list of goals	• Interviewing based on cognitive profiles • Market research • Existing system analysis • Case scenarios • Stakeholder selection based on expertise of product, sponsors and deliverables
	Count	▸ Stakeholders participating ▸ Initial list of goals	■ Total number of stakeholders ■ Total number of initial goals	
	Data Entry	▸ Total number of stakeholders ▸ Total number of initial goals ▸ Name of initial goals		• Collaboration system
	Distribute	▸ Initial list of goals to stakeholders	■ First compilation table	
Sprint	• Develop • Wrap • Review • Adjust	▸ First compilation table ▸ Activity card compiler output -1 to activity card compiler output N-1	■ Activity card ■ Activity card compiler output-1 to activity card compiler output-N ■ Identified changing requirements	• Ontology • Collaboration system • Spiral model of [24] for conflicting requirements; identify conflicts, alternatives, judge and trade off • Multiple sources • Requirement value based on business value, negative value, size/effort estimation, input from developers, external change, learning experience, project constraints • Cumulative voting • Sign off by the use of cognitive profiles
Post-sprint		▸ Activity card compiler output	■ Software requirements documentation	• Documenting

In summary, this framework addresses all the concerns that are mentioned in Section 4.1. It uses the overall approach proposed by Sen and Hemachandran in [1] to ;

- Develop the requirements rapidly, due to 1 to 4 weeks sprints
- Achieve the highest satisfaction of the market and costumers' need by actively involving people that know the market
- Addresses the issue that important requirements may go unrecognized or recognized too late due to multiple information source, the actively involvement of all the stakeholders in the discussion about requirements and multiple validation techniques.

As mentioned before, in the pre-sprint, initial requirements are gathered. One of the gathering techniques used in this process is interviewing specific stakeholders. During these interviews, the cognitive tool of Carod and Cechich in [2] can be used to improve the understanding of the requirements and goals by the stakeholders. Further on, the use of templates are addressed in the framework to speed up the validation process by simplifying the formulation of the requirements and case scenario's are used to create a better understanding of the goals and constraints of the system. In the sprint process the following techniques are taken into account;

- A collaboration system to enhance knowledge sharing, understanding of each other and resolving conflicts of interests.
- The use of an ontology in order to keep consistency during the overall development
- The spiral model of [3] to resolve brief, ambiguous and conflicting requirements
- Multiple sources, to validate the requirements
- Requirement value assignment based on the descriptive model of [4] and the prioritization aspect from the case study to make sure that all priority aspects are considered
- Cumulative voting for prioritizing the requirement to avoid human subjectivity
- Sign off based on cognitive profiles to enhance the understanding of the requirements by the stakeholders that have to sign off.

Final, in the post-sprint process, the business specification template found in the case study is used to document the requirements.

6. Evaluation of the recommended situation

This chapter describes how the framework is validated based on the guidelines given by Winbladh et al. in [60]. Further on, it explains how the framework can be evaluated in practise based on metrics and the constraints of the framework are described. In the final section conclusions are drawn and an answer is given on Research Question 5.

6.1 Validation of the proposed framework

The study in [60] showed that many companies do not use textbook requirements engineering processes. According to [60] there are many contributing factors why these approaches are neglected for example through the stakeholders' preferences, habits and the application domain. In order to enhance the adaptability of formal requirement engineering approaches, Winbladh et al. conducted a survey in [6]. In this survey, they used a set of questions that are based on pre-defined characteristics of requirements engineering and needed to be answered by the participants who have experience in software development. They used standard statistics as metrics to evaluate the results of the research. Findings from this survey suggested that the guidelines described in Table 11 should be used to enhance the adaptability of requirements engineering approaches [6].

Table 11: Guidelines to enhance the adaptability of requirements engineering approaches [6]

Guideline 1	**Strive for narrative structures** Requirements with narrative structures, e.g. use cases and stories, describe operational requirements and can capture informal conversations with customers. Narrative requirements are typically easy to understand, could be the starting point for test scenarios, and message sequence charts.
Guideline 2	**Strive for domain descriptions** Consistently referencing defined terms in a domain model can decrease ambiguity. Cleary defined terms can also increase understand-ability. A domain description aids testability as abstract test data can be selected from the model. A domain model can also be a useful starting point to module design.
Guideline 3	**Strive for modularity** Keeping narrative and domain models separate and decomposing these into related functionalities and concepts, is a useful strategy to provide higher levels of maintainability and traceability
Guideline 4	**Strive for relations among requirements** Relations between domain and narrative structures can improve un-ambiguity and understanding of the requirements. Relations can provide guidance for change impact analysis during maintenance. Syntactic and semantic relations could also be the basis for a useful requirements engineering tool
Guideline 5	**Strive for a formal internal model** Providing a formal description of the requirements could increase the level of un-ambiguity and could be used as a basis for automated test support.
Guideline 6	**Strive for a tool support** Tool support could make use of the guidelines above to provide functionality including searching, re-factoring and showing relations. The tool should provide an easy-to-use interface that shields the user from the complexity of internal formal representations.

To evaluate the framework proposed in Section 4.2, these guidelines are used as a metric to determine the usability of it;

Guideline 1 - Strive for narrative structures
In the pre-sprint of the framework, case scenarios and interviews are used to describe the operational requirements that can be formulated into narrative requirements which are easy to understand by the stakeholders.

Guideline 2 - Strive for domain descriptions
In the framework, the use of ontology is considered. By the use of this, specific terms are pre-defined which will increase the understand-ability and decrease the ambiguity among the stakeholders by keeping the terms consistent.

Guideline 3 - Strive for modularity
All decisions and retrieved information are registered in a collaboration system. Because of that, requirements can be traced back to the information on where the requirement is elicit from. This can be from narrative models (case scenarios, interviews, etc.) or domain models (market research, existing systems, etc.). Further on, the proposed collaboration system of [37] in the framework provides the ability to easily maintain requirements through to a Wiki mechanism.

Guideline 4 Strive for relations among requirements
In the sprint process, the requirements are collaboratively engineered based on the information gathered in the pre-sprint process. The information that is used in this process consists of domain and narrative data. This information is combined and registered during the negotiations in the sprint process to indicate syntactic and semantic relations among it that is needed to elicit and maintain requirements. These relations should be used as guidance for change impact analysis.

Guideline 5 Strive for a formal internal model
The Activity card that is used in the proposed framework in Section 4.3 is a formal model to describe requirements that should increase the level of un-ambiguity and could be used as a basis for automated test support.

Guideline 6 Strive for a tool support
The use of a collaboration system in the proposed framework in Section 4.3 provides functionality including searching, re-factoring and showing relations.

However, to measure the real impact of the framework, metrics should be used to gain insights, control and improvements of the requirements engineering process. Therefore, Costello and Lui proposed a set of metrics in [7]. These metrics are presented in Table 12.

Table 12: Metrics for requirements engineering approaches [7]

Metric	**Overview purpose**
Requirements volatility	Indicates changes (additions, deletions, modifications) and reasons for changes to requirements. Provides insight into system maturity and stability. Aids in predicting future requirements, design, and code volatility. Essential in interpreting other metrics.
Requirements traceability	Indicates degree to which development organization maintains accountability for meeting requirements at each stage of life cycle via a requirements traceability matrix. Provides quantitative means for determining whether all required relationships/dependencies are addressed. Assists in exposing incompletely specified, overly specified and complex areas of system. Essential in interpreting other metrics.
Requirements completeness	Indicates completeness of all sections of requirements specifications, whether all allocated higher level requirements are addressed, and the degree of decomposition of allocated higher level requirements.

	Assists in determining readiness to proceed to design.
Requirements defect density	Indicates number of requirement defects that are initially detected during an inspection or walkthrough. Classified by type, critically, and source. Provides early insight into quality, assists in cost/schedule estimation, and indicates effectiveness and extent of inspection/walkthrough process. Useful in predicting product/process volatility. Essential in interpreting other metrics.
Requirements fault density	Indicates number of requirements faults that are initially detected during test execution or posttest analysis. Classified by type, critically, and source. Assists in determining effectiveness of software process and the extent and effectiveness of testing. Useful in predicting product/process volatility and quality. Essential in interpreting other metrics.
Requirements interface consistency	Indicates consistency and completeness of interface information at each level of specification.
Requirements problem report/action item issue	Indicate number of requirement problems detected/issues raised via any process; as such include requirement defects and faults. Characterize problems by source, product, type of problem, finding activity severity, critically, age, and primary reason for closure. Number and status of requirements problem reports indicate quality of requirements engineering and inspection processes. Essential in interpreting other metrics.
Requirements integrated progress	Indicates overall requirements progress. Encompasses measures of volatility, traceability, completeness, defect density, fault density, test coverage, and problem report action/items as appropriate for phase under consideration

The first metric consists of counts of requirements that are changed in each iteration and is classified by the reason of change [7]. When the framework is properly applied, the volatility is probably high in the beginning and decreases over time in the requirements engineering process.

The second metric, documents the set of linkages of requirements in a specification to their origins and descendants due to a requirement traceability matrix [7]. This matrix indicate how well the framework maintain accountability for meeting requirements throughout the requirements specification activities by determining if all required relationships and dependencies are addressed and expose incomplete specified or overly complex areas of the system.

The third metric, quantifies the level of decomposition of higher level requirements allocated to given specifications and indicate the number of “to be completed’ items and the number allocated requirements that trace to items in the lower level specification [7]. If there is a low degree of decomposition, it may mean that;

- The requirement is simple or specifies a very limited function or behavior
- The requirement’s decomposition spans several specifications
- The high level requirement is documented at a too low level of detail
- The analysis of the requirement was superficial and the requirement needs to be decomposed further.

If there is a high degree of decomposition, it may mean that;

- It represents a complex function and should be monitored throughout the life cycle
- The level of requirement itself may be too general and should be spilt into several high-level requirements before going to the next iteration.

In general measuring the degree of decomposition and the number of allocated requirements indicates if all the higher level requirements have been addressed in the lower specifications. If this is the case, then there is a high degree of completeness if not, the requirements are probably incomplete.

The fourth and fifth metric count the number of requirement defects and faults where a fault and a defect differ only in the means by which each is detected. A defect is detected via inspection of the requirements and a fault is detected via testing. In both cases the number of defects/faults found is divided by the size of the item inspected or the size of the item under test to obtain a density measure. Based on this, the framework can be validated among the number of defect or fault requirements. If this is high, it could indicate that the requirements elicitation needs to be improved.

The sixth metric count the number of requirements those are consistent with the design and code in each iteration. This indicates the amount of requirements that are truly realized and therefore are valid requirements.

The seventh metric counts all the problems that occurred during the overall process. This indicates the quality of the requirements engineering.

The final metric involves all the previous mentioned metrics to indicate the overall requirements progress of the framework.

In order to measure the quality of the frameworks, these metrics have to be used in multiple requirement engineering approaches or models that currently are used in market-facing projects within Chubb. Findings from these approaches can be used to benchmark the new framework. Therefore, the proposed framework in Section 4.2 should be implemented in a new market-facing project after or simultaneously with another market-facing project. However, these metrics only measure quantitative data from approaches. In order to evaluate the new framework thoroughly, also qualitative data need to be gathered. Qualitative data can be gained through interviews with stakeholders that are involved in multiple market-facing project including projects wherein the new framework is used. Nevertheless, the fact that every project differs in size and complexity should be taken into consideration during the evaluation.

6.2 Constraints of the proposed framework

The aim of the proposed framework is to improve the requirements engineering process in market-facing projects within Chubb. These projects are in a new domain of technology that is being developed within Chubb. Therefore, most of the knowledge that is needed during these projects are gathered through the business and is not mainly based on existing systems. This implicates that elicitation techniques are used to gather knowledge of the business that is not already addressed in other systems. Because of that, this framework is probably only suitable for new domains within Chubb Australia.

In addition, as the needs of the market are changing rapidly, the market-facing technology has to be developed rapidly too in order to achieve those needs. The proposed framework is addressing this by developing the system iteratively in sprints. Although the proposed framework tries to capture all the goals it could be that some goals still go unrecognized due to time constraints and therefore this framework would probably only suitable for projects that do not address critical business processes.

Another constraint is that the proposed framework is specific designed for Chubb Australia. This framework is elaborated on the existing requirement engineering approaches that are used in market-facing projects within Chubb Australia. The reason that is chosen to elaborate on existing

approaches is to enhance the adaptability by the stakeholders. Therefore, implementing this framework in another context could be challenging.

Further on, in the proposed framework the involvement of multiple stakeholders have to consider. For example in the sprint process the requirements and goals have to be discussed and validated by multiple stakeholders in order to enhance the correctness of it. If this is done by one person, there is a chance that the requirements are subjective and do not meet the goal of the project.

Finally, in the proposed framework also the use of a collaboration system is suggested. However, special knowledge is required by stakeholders in order to use a collaboration system properly. Therefore, stakeholders should already be familiar with similar systems or should need some special training on using these systems.

6.3 Conclusion - Answer to Research Question 5

This chapter answers the last research question: "How to evaluate the proposed framework/methods?". In the beginning of this chapter, the guidelines mentioned by [6] are given and reflected to the new framework to measure the adaptability. From this reflection it turned out that every guideline is covered in the framework. However, these guidelines do not suggests any improvements in practice but increase the likelihood of it. In order to measure and evaluate the new framework in practice, the metrics of [7] can be used. These metrics collects quantitative data about the requirements engineering. Nonetheless, only quantitative data is not enough to evaluate the new framework because every project differs in size and complexity. To evaluate the new framework thoroughly, also qualitative data need to be gathered. This can be gathered through interviewing stakeholders that are involved in projects wherein the new and existing framework is used. Therefore, the best moment to put the framework into use is after or simultaneously with another market-facing project that consider the existing method and is measured with the metrics suggested in [7]. Further on, the overall constraints of the framework that should be taken into account are;

(i) This framework is probably only suitable for new domains within Chubb Australia.
(ii) This framework is specifically made for market-facing projects, and deploying it in another context can be challenging.
(iii) This framework is specifically designed on the involvement of multiple stakeholders
(iv) Stakeholders need some expertise to work with the collaboration system mentioned in the framework

7. Discussion

The purpose of this section is to point out the limitations of this research, new research directions and the recommendations for Chubb. In the first section of this chapter, a summery is given where in all the research questions are answered. Based on this, recommendations are made for Chubb in the second section. Unknown artefacts found in the case study where little information is available on are mentioned in the third section. Finally, the limitations and implications for further research are described in the last section.

5.1 Answers to the research questions

The main research question of this thesis is; “How to gather, negotiate, validate and prioritize the requirements of market-facing projects from a complex multi-stakeholder perspective?” and is answered by five sub research questions. These research questions are answered in the previous chapters and summarized here. The first research question is “What methods/approaches for similar projects and companies are suggested in scientific literature?” and is answered based on a scientific literature study. Articles from the scientific field are found through multiple search engines and selected based on pre-defined inclusion and exclusion search criteria.

The overall requirement engineering approaches found during this literature study are mainly based on plan based and agile strategies. The former strategy generally consists of an overall implementation plan and the latter of small iterations where stakeholders have to work intensively with each other to achieve the highest satisfaction. Techniques found in the literature to gather requirements or to improve the gathering process are; grouping diverse stakeholders, interviewing experts, the use of case scenarios, the use of default templates and documenting requirements and decisions. For the negotiation process, the literature suggests to keep consistency by the use of an ontology, to negotiate iteratively to solve conflicting demands, to consider emotional factors to gain an better understanding of the stakeholders and by the stakeholders and to use a collaboration system to improve the accessibility of the requirements and to document discussions. Techniques that are suggested by the literature to validate requirements are; reviewing viewpoints from different stakeholders, making use of cognitive profiles to gain a better understanding by the stakeholders and the use of automated validations mechanisms in case of vast amounts of requirements to speed up the process. Further on, mechanisms that are suggested for the prioritization process are cost and benefits mechanisms and cumulative voting. Stakeholders collaboratively determine what the cost is of implementing each potential requirement and how much value this requirement will gain in cost and benefit mechanisms. With cumulative voting, each stakeholder is given an amount of 100 points that he or she has to use to vote on his or her most important requirements. The requirements that have the most votes are marked as important and requirements that have the least votes are marked as not necessary or dropped out of the project.

The answer of the second question: “How does Chubb currently gather, negotiate, validate and prioritize the requirements in market-facing projects?” is answered based on an exploratory case study on the iClose and the ARCH project within Chubb. Several experts were interviewed, available documentation was analyzed of both projects and direct observations were done in the iClose project. Findings from this study suggested that the overall requirements engineering in market-facing projects is done based on an iterative approach where in prototypes are used to negotiate, validate and prioritize requirements. Further on, the requirements engineering and development of the system are done simultaneously and a business specification document is used to store the requirements and decisions. Stakeholders in the projects are chosen among three aspects, namely based on their expertise, their (financial) sponsorship and delivery responsibilities. Techniques that are used to gather requirements in market-facing projects are;

market-research to determine if and what need there is for market-facing technology, existing systems (competitor systems and legacy systems) to gain usability and best practices, workshops to brainstorm new ideas, case scenarios to retrieve additional requirements and face to face meetings to extract knowledge for the system from business experts. During the negotiation process, the stakeholders have to understand each other to establish an effective discussion. Therefore, Chubb uses visualization tools (process maps and images), face-to-face meetings and a collaboration system to enhance the understanding by the stakeholders. If there are conflicting requirement concerns; the requirement engineer will reflect these concerns to the scope of the project to determine if they are inline with the project. He or she will validate the quality of the conflicting requirement concerns by making everyone aware of the conflict and based on their opinions he or she will make a decision and the requirement engineer will estimate how much time and money it will cost to realize the conflicting requirements. Based on these aspects he will determine whether to drop the requirement or to add the requirement to the development of the system. The validation of requirements in market-facing technology projects is mainly done through two techniques, namely; the use of multiple sources and a sign off technique. In the former technique multiple sources (existing information and experts) are used to check if the requirement is correct. The latter technique is used to make sure that the stakeholder, who is responsible of specific requirement concerns, agreed on the formulated requirements of those concerns. Final, the prioritization of the requirements in market-facing technology projects is mainly done based on the scope of the project and the return on investment of the requirement. When a requirement delivers a high return on investment and it is in the scope of the project, the probability that it will gain a high priority is most likely. When the requirement delivers a low return on investment and is not inline with the scope of the project, the probability that it will gain a low priority is most likely.

The answer of third research question: “What can be improved in the way Chubb gathers, negotiates, validates and prioritizes requirements reviewing the current practice and comparing against scientific literature?” is;
The overall approach that is currently used in market-facing projects within Chubb can be improved by the approach proposed by Sen and Hemachandra in [1]. Their approach is similar to the approach that Chubb currently is using. By using this approach, Chubb will reduce the chance that important requirements are unrecognized or are recognized too late. In the requirements gathering process, the requirement engineers at Chubb could use cognitive tools to gain a better understanding by the business experts about the system and requirement concerns. Further on, case scenarios can be used to capture the systems’ purpose and reason about whether a given design will meet that purpose. Standard templates could be used in the elicitation process to speed up the requirements validation and prioritization process. In the negotiation process, ontology can be used to reduce the issue of misinterpretations of requirements by the stakeholders and a collaboration system should be used in all projects to improve the cost effectiveness, user satisfaction and outcomes of the project. In the validation process, the model proposed by Ahmad in [3] can be used. This model addresses the problem of ambiguous and conflicting requirements by negotiating and validating the requirements in iterations that consists of pre-defined steps. The signing off technique that is used at Chubb to validate the requirements can be improved by using the cognitive method proposed Carod and Cechich in [2]. With this method, requirements are presented to the stakeholders in a way that the specific stakeholders feel most comfortable with. Therefore, the specific stakeholders will better understand what he or she needs to sign off on. In the prioritization process, the model of Racheva et al. in [4] can used as a guideline to make sure that the stakeholders consider all important requirement aspects in the prioritization process. Further on, the cumulative voting method proposed by Leffingwell and Widrig in [5] can also be used to reduce to gap of human subjectivity in the prioritization process.

The fourth research question is “What would be a better way for Chubb to carry out requirements engineering for market-facing projects?” and is answered based on the findings and recommendations described above. A framework is made that answers this question. This framework is presented on page 59 and is an agile framework that consists of small iterations.

The processes that are considered in these iterations are; pre-sprint, sprint and post-sprint. Requirements are first elicited in the pre-sprint processes due to the following techniques;

(i) Market research
(ii) Stakeholder selection based on expertise of product, sponsors and deliverables
(iii) Interviewing based on cognitive profiles
(iv) Existing system analysis
(v) Case scenarios

After that, in the sprint process the requirements are negotiated, validated and prioritized. To facilitate this discussion, ontology can be used to reduce misunderstandings. Notes can be stored in the collaboration system to understand later on why decisions are made. In terms of conflicting goals or requirements, the spiral model of [24] can be used to iteratively and collaboratively solve the conflicts. The goals can be validated through multiple sources and signed off by the specific stakeholders. Then the goals and requirements can be prioritized based on business value, negative value, size/effort estimation, input from developers, external change, learning experience, project constraints. After this, in the post-sprint process, the requirements and goals of the iteration need to be documented in the business specification document.

The fifth research question: "How to evaluate the proposed framework?" is answered by validating the framework against the guidelines given by Winbladh et al. in [6]. Further on, the framework can be evaluated in practise by the metrics that are proposed by Costello and Lui in [7]. However, these metrics produce only quantitative data on the framework. To evaluate the framework thoroughly, also qualitative data should be considered. Therefore interviews need to be conducted with stakeholders that are familiar with the existing approach and the proposed framework in this thesis.
The implementation of this framework should be done after or simultaneously with a project that use the existing approach wherein the metrics of Costello and Lui [7] are used. In this way, data from both frameworks can be evaluated. However, this framework is not applicable in every situation and therefore it has some constraints. These constraints are;

(i) This framework is probably only suitable for new domains within Chubb Australia.
(ii) This framework is specifically made for market-facing projects, and deploying it in another context can be challenging.
(iii) This framework is specifically designed on the involvement of multiple stakeholders
(iv) Stakeholders need some expertise to work with the collaboration system mentioned in the framework

5.2 Recommendations for Chubb

In summery, Chubb could use the framework that is proposed in this thesis to gather, negotiate, validate and prioritize requirements in a structured way. This could reduce time and could increase the outcomes of the requirements engineering process. Next to this, a collaboration system could be used at Chubb in all market-facing projects. With the use of a collaboration system the communication between the stakeholders are logged. Through this, the requirement engineer has the ability to understand why certain decisions are made or still have to be made. This could increase the decision-making process in validating and prioritizing requirements.
Another aspect found in this thesis is that the process of solving requirements in market-facing projects within Chubb could be improved by the use of the spiral model that is proposed by Ahmad in [3]. This model, demonstrates how conflicts can collaboratively be solved in pre-defined activities. By using this model, Chubb can reduce the amount meetings and time that is needed for solving these conflicts in the traditional way.
Further on, findings in this thesis also suggests that Chubb could use cognitive profiles during face-to-face meetings and in validating requirements. By the use of these profiles, requirements can be presented to the stakeholders in a way wherein they feel most comfortable with. This could increase the understanding of the requirements and requirement concerns by the stakeholders.

Finally, in this thesis the use of ontology is suggested for market-facing projects within Chubb. In ontology, definitions of terms from the business and IT are given. This increases the consistency in the project and the understanding of the terms by the stakeholders and can contribute in gathering, negotiating, validating and prioritizing requirements.

5.3 Lessons learnt from Chubb

During the scientific literature study, no information was found on the scope of the project in the requirements prioritization process. For example the descriptive model of Racheva et al. in [4] and the cumulative voting of method Leffingwell and Widrig in [5] do not specifically suggest that the scope of the project is one of the aspects where on the stakeholders prioritize the requirements. Although this aspect was not found in the literature, in the case study it turned out that the scope of the project is one of the major aspects that is used in prioritizing requirements in market-facing projects within Chubb. In the field of science, researchers could investigate if the scope of the project is also considered in other projects and what the outcomes are of taken this aspect into consideration in the prioritization process.
Further on, in the found scientific literature, not much attention is given on singing off requirements. From the case study it turned out that signing off the requirements is one of the major techniques that is used to make sure that requirements are validated. Specific stakeholders have to sign off the requirements and by this they have to take the responsibility that the requirements he or she checked are ok. Researchers could investigate what the influences are of using this technique in projects and how comfortable stakeholders are with this technique.
Finally, during the case study it turned out that stakeholders are selected in the project by considering three aspects, namely; sponsorship (people who found the project), expertise (people who bring in certain expertise) and deliverables (people who deliver the final product). By using these aspects, the Project Manager makes sure that he or she involves the right stakeholders in the project and reduces the change of wrongly gathered, negotiated, validated and prioritized requirements. Researchers could investigate what the effects are in projects when considering these aspects in selecting stakeholders.

5.4 Limitations and implications for further research

The goal of this study is to explore how to gather, negotiate, validate and prioritize the requirements of market-facing projects from a complex multi-stakeholder perspective. Due to the fact that the goal of the research is dedicated to Chubb, it has its limitations in generalizing the results for the field of science and in practice in other companies. The explorative case study is only conducted in Chubb and reflected with the findings from the literature study described in chapter 2. To generalize the findings and recommendations for the field of science and other practices, this study also needs to be conducted in other companies that are developing market-facing systems with multiple stakeholders.
Further on, the researcher did an extensive literature study and found articles based on a systematic way to reduce the chance of missing important literature on requirements engineering in market-facing projects from a multi-stakeholder perspective. However, due to the vast amount of literature and time limitations it could be that there are other elements in the scientific literature that can be used to improve the proposed framework and recommendations. For this, further research is required.
Several experts in the case study suggested that developers do not always understand the demands from the business. However, quantitative data from the case study showed that developers do not have intensively contact with requirements engineers among certain requirements concerns. This could mean that developers understand the requirements in the iClose project or that there was extensively contact among requirements concerns between the developer and requirement engineer where in the observer was not aware of. Therefore, further research is required to determine if developers understand the business and their requirements.
According to several experts, the selection stakeholders for a project are based on three aspects (sponsorship, expertise and deliverables). However, quantitative data showed that these three

aspects are presented in the project but it did not prove that these aspects are used for selecting stakeholders. Knowing how stakeholders are selected can contribute to the understanding of the requirements by the stakeholders in market-facing projects. Therefore, further research is required on how stakeholders are selected in market-facing projects.
The framework that is proposed in this thesis is based on the existing method in Chubb and the framework that is proposed by Sen & Hemachandra [1]. With their framework they try to address the gap of that requirements may go unrecognized or recognized too late. However, they only tested their framework on the correctness of requirements and not yet on the completeness. Therefore, further research is required to investigate the completeness of requirements in the proposed framework.
Finally, the cognitive method of Carod and Chechic [2] that is included in the proposed framework to gain a better understanding of the requirements by the stakeholders based on cognitive profiles is only tested in a controlled environments. Because of this, further research is required to test this method in practice.

References

1. A.M. Sen and K. Hemachandran, “Elicitation of goals in requirements engineering using agile methods,” *Proc. Proceedings - International Computer Software and Applications Conference*, 2011, pp. 263-268.
2. N.M. Carod and A. Cechich, “Cognitive-driven requirements prioritization: A case study,” *Proc. Proceedings of the 9th IEEE International Conference on Cognitive Informatics, ICCI 2010*, 2010, pp. 75-82.
3. S. Ahmad, “Negotiation in the requirements elicitation and analysis process,” *Proc. Proceedings of the Australian Software Engineering Conference, ASWEC*, 2008, pp. 683-689.
4. Z. Racheva, et al., “A conceptual model of client-driven agile requirements prioritization: Results of a case study,” *Proc. ESEM 2010 - Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 2010.
5. D. Leffingwell and D. Widrig, *Managing software requirements : a use case approach*, Addison-Wesley, 2003.
6. K. Winbladh, et al., “Eliciting required characteristics for usable requirements engineering approaches,” *Proc. Proceedings of the ACM Symposium on Applied Computing*, 2009, pp. 360-364.
7. R.J. Costello and D.-B. Liu, “Metrics for requirements engineering,” *Journal of Systems and Software*, vol. 29, no. 1, 1995, pp. 39-63.
8. N. Gorla, et al., “Organizational impact of system quality, information quality, and service quality,” *Journal of Strategic Information Systems*, vol. 19, no. 3, 2010, pp. 207-228.
9. A. Ivanovic, *Dictionary of Marketing*, Peter Collin Publishing, Ltd., 1998, p. 244
10. CommerceDictionary.com, “CommerceDictionary.com,” 2011; http://www.commercedictionary.com/.
11. B. Nuseibeh and S. Easterbrook, “Requirements engineering: a roadmap,” *Book Requirements engineering: a roadmap*, Series Requirements engineering: a roadmap, ed., Editor ed.^eds., ACM, 2000, pp. 35-46.
12. G. Regev and A. Wegmann, “Where do goals come from: The underlying principles of goal-oriented requirements engineering,” *Proc. Proceedings of the IEEE International Conference on Requirements Engineering*, 2005, pp. 353-362.
13. M. Shibaoka, et al., “GOORE: Goal-oriented and ontology driven requirements elicitation method,” *Book GOORE: Goal-oriented and ontology driven requirements elicitation method*, Series GOORE: Goal-oriented and ontology driven requirements elicitation method 4802 LNCS, ed., Editor ed.^eds., 2007, pp. 225-234.
14. H. Kaiya and M. Saeki, “Using domain ontology as domain knowledge for requirements elicitation,” *Proc. Proceedings of the IEEE International Conference on Requirements Engineering*, 2006, pp. 186-195.
15. H. Kaiya, et al., “AGORA: attributed goal-oriented requirements analysis method,” *Proc. Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on*, 2002, pp. 13-22.
16. A.M. Sen and S.K. Jain, “An Agile Technique for Agent Based Goal Refinement to Elicit Soft Goals in Goal Oriented Requirements Engineering,” *Proc. Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on*, 2007, pp. 41-47.
17. A. Sen and S.K. Jain, “A Visualization Technique for Agent Based Goal Refinement to Elicit Soft Goals in Goal Oriented Requirements Engineering,” *Proc. Requirements Engineering Visualization, 2007. REV 2007. Second International Workshop on*, 2007, pp. 2-2.
18. L. Marcio Cysneiros and E. Yu, “Requirements engineering for large-scale multi-agent systems,” *Software engineering for large-scale multi-agent systems*, 2003, pp. 77-148.
19. J. Martin, *Rapid application development*, Macmillan Publishing Co., Inc., 1991.
20. A. Fruhling, et al., “Designing and evaluating collaborative processes for requirements elicitation and validation,” *Proc. Proceedings of the Annual Hawaii International Conference on System Sciences*, 2007.

21. Z. Balfagih and M.F. Hassan, “Quality model for web services from multi-stakeholders' perspective,” *Proc. Proceedings - 2009 International Conference on Information Management and Engineering, ICIME 2009*, 2009, pp. 287-291.
22. D. Arnold, et al., “Modeling and validating requirements using executable contracts and scenarios,” *Proc. 8th ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2010*, 2010, pp. 311-320.
23. B. Evans, “Global CIO: The Top 10 CIO Issues For 2010,” 2009 http://www.informationweek.com/news/global-cio/security/showArticle.jhtml?articleID=222002799.
24. M. Duvall, “Top 10 Concerns Of CIOs,” 2009; http://www.information-management.com/news/-10016150-1.html.
25. J. Luftman, “Key issues for IT executives 2007,” *MIS Quarterly Executive*, vol. 7, no. 2, 2008, pp. 269-286.
26. Gartner, “Data Modeling Without Meetings: Wiki over Whiteboard?,” 2010; http://www.burtongroup.com/Research/PublicDocument.aspx?cid=1987.
27. Gartner, “New Delivery Models and Requirements Drive the Evolution of Finance,” 2010; http://www.gartner.com/DisplayDocument?doc_cd=209156.
28. L. Kof, “On the identification of goals in stakeholders' dialogs,” *Book On the identification of goals in stakeholders' dialogs*, Series On the identification of goals in stakeholders' dialogs 5320 LNCS, ed., Editor ed.^eds., 2008, pp. 161-181.
29. F. Cafer and S. Misra, “A Cognitive Requirement Specification Model,” *2009 24th International Symposium on Computer and Information Sciences*, 2009, pp. 517-520.
30. D. Port, et al., “Using simulation to investigate requirements prioritization strategies,” *Proc. ASE 2008 - 23rd IEEE/ACM International Conference on Automated Software Engineering, Proceedings*, 2008, pp. 268-277.
31. M. Daneva and A. Herrmann, “Requirements prioritization based on benefit and cost prediction: A method classification framework,” *Proc. EUROMICRO 2008 - Proceedings of the 34th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2008*, 2008, pp. 240-247.
32. J. Webster and R.T. Watson, “Analyzing the past to prepare for the future: Writing a literature review,” *Management Information Systems Quarterly*, vol. 26, no. 2, 2002, pp. 3.
33. R.B. Schwartz and M.C. Russo, “How to quickly find articles in the top IS journals,” *Communications of the ACM*, vol. 47, no. 2, 2004, pp. 98-101.
34. S. Nejati and M. Chechik, “Let's agree to disagree,” *Proc. 20th IEEE/ACM International Conference on Automated Software Engineering, ASE 2005*, 2005, pp. 287-290.
35. X. Liu, et al., “Priority assessment of software process requirements from multiple perspectives,” *Journal of Systems and Software*, vol. 79, no. 11, 2006, pp. 1649-1660.
36. R. Peng, et al., “A Requirements Maturity Measurement Approach based on SKLSEWiki,” *Proc. Proceedings - International Computer Software and Applications Conference*, 2010, pp. 251-254.
37. V. Gaur, et al., “An application of multi-person decision-making model for negotiating and prioritizing requirements in agent-oriented paradigm,” *Proc. DSDE 2010 - International Conference on Data Storage and Data Engineering*, 2010, pp. 164-168.
38. H. Jiang and X. Yang, “Performance requirement elicitation for financial information system based on ontology,” *Proc. IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2009.
39. F. Molina, et al., “An MDE Modeling Framework for Measurable Goal-Oriented Requirements,” *International Journal of Intelligent Systems*, vol. 25, no. 8, 2010, pp. 757-783; DOI 10.1002/int.20430.
40. D. Wu, et al., “Finding Success in Rapid Collaborative Requirements Negotiation Using Wiki and Shaper,” *43rd Hawaii International Conference on Systems Sciences Vols 1-5 (Hicss 2010)*, 2010, pp. 663-672.
41. D. Port and T. Bui, “Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications,” *European Journal of Information Systems*, vol. 18, no. 4, 2009, pp. 317-331; DOI 10.1057/ejis.2009.19.

42. S. Kelly, “Towards an Evolutionary Framework for Agile Requirements Elicitation,” *Eics 2010: Proceedings of the 2010 Acm Sigchi Symposium on Engineering Interactive Computing Systems*, 2010, pp. 349-352.
43. T. Gorschek and C. Wohlin, “Requirements abstraction model,” *Requirements Engineering*, vol. 11, no. 1, 2006, pp. 79-101; DOI 10.1007/s00766-005-0020-7.
44. R. Colomo-Palacios, et al., “A study of emotions in requirements engineering,” *Book A study of emotions in requirements engineering*, Series A study of emotions in requirements engineering 112 CCIS, ed., Editor ed.^eds., 2010, pp. 1-7.
45. P. Laurent, et al., “Towards automated requirements triage,” *Proc. Proceedings - 15th IEEE International Requirements Engineering Conference, RE 2007*, 2007, pp. 131-140.
46. I. Ramos, et al., “Requirements engineering for organizational transformation,” *Information and Software Technology*, vol. 47, no. 7, 2005, pp. 479-495.
47. P. Chatzipetrou, et al., “Prioritization of issues and requirements by cumulative voting: A compositional data analysis framework,” *Proc. Proceedings - 36th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2010*, 2010, pp. 361-370.
48. A. van Lamsweerde, “Goal-oriented requirements engineering: a guided tour,” *Proc. Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, 2001, pp. 249-262.
49. R.O. Briggs, et al., “Collaboration engineering with ThinkLets to pursue sustained success with group support systems,” *Journal of Management Information Systems*, vol. 19, no. 4, 2003, pp. 31-64.
50. I. Jacobson, et al., *Object-oriented software engineering: a use case driven approach*, Addison-Wesley, 1992.
51. M. Lee, “Foundations of the WinWin Requirements Negotiation System,” University of Southern California, 1996.
52. J.A. Russell, et al., “Affect Grid: A single-item scale of pleasure and arousal,” *Journal of personality and social psychology*, vol. 57, no. 3, 1989, pp. 493.
53. R.M. Felder, “Matters of style,” *Asee Prism*, vol. 6, no. 4, 1996, pp. 18-23.
54. R.M. Felder and L.K. Silverman, “Learning and teaching styles in engineering education,” *Engineering education*, vol. 78, no. 7, 1988, pp. 674-681.
55. R.K. Yin, *Applications of case study research*, Sage Publications, Inc, 2003.
56. R.E. Stake, *The art of case study research*, Sage Publications, Inc, 1995.
57. M.Q. Patton, *Qualitative evaluation and research methods*, Sage Publications, Inc, 1990.
58. B. Soloman and R. Felder, “Index of Learning Styles Questionnaire,” 2003; http://www.engr.ncsu.edu/learningstyles/ilsweb.html.
59. J.M. Carroll, *Making use: scenario-based design of human-computer interactions*, MIT Press, 2000, p. 46.
60. J. Aranda, et al., “Requirements in the wild: How small companies do it,” *Proc. Proceedings - 15th IEEE International Requirements Engineering Conference, RE 2007*, 2007, pp. 39-48.

Appendices

Appendix A: Process diagram of client request

CONFIDENTIAL

Appendix B: List of excluded studies with rationale for exclusion

Reason for excl.	Authors	Title	Year	Source
E1	Dasgupta et al.	E-Commerce in the Indian Insurance Industry: Prospects and Future	2002	Electronic Commerce Research
E1	Taylor et al.	Technology readiness in the e-insurance industry: an exploratory investigation and development of an agent technology e-consumption model	2002	Journal of Insurance Issues
E2	Balfagih, Z., Hassan, M. F.	Quality model for web services from multi-stakeholders' perspective	2009	ICIME 2009
E2	Qureshi, N. A., Perini, A.	Requirements engineering for adaptive Service Based Applications	2010	RE2010
E2	Wen et al.	Stakeholders-driven requirements semantics acquisition for networked software systems	2010	COMPSACW 2010
E2	Verlaine et al.	Towards automated alignment of web services to requirements	2010	WeRE 2010
E3	Carod, N. M., Cechich, A	Cognitive influences in prioritizing software requirements	2010	ICSOFT 2010
E4	Daneva, M., Herrmann, A	Requirements prioritization based on benefit and cost prediction: A method classification framework	2008	EUROMICRO 2008

Appendix C: Interview template

Before interview

- Explain objective
- For the purpose of my master thesis, this interview will be recorded.
- This interview will approximately take 30 minutes

Objective

The objective of this interview is to analyse and understand how requirement engineering is currently done in Chubb. With requirements engineering in the terms of: requirements gathering, requirements negotiations, requirements validation and prioritization.

Questions

1. Organizational

1.1	Can you tell me what kind of company Chubb is? (Number of employees, age, business area etc…)
1.2	Can you tell me something about the projects / services the company provide in terms of time on the market, typical customers/ end-users, projects?
1.3	What is your position in the company? (Role, daily tasks, responsibilities, etc)

2. Definition of requirements

2.1	What is a "requirement" to you?
2.2	What is a "good requirement" to you? And to the company? Is the quality of the requirements assessed? How?
2.3	How is Chubb gathering requirements for a project like iClose?
2.4	Is there a static methodology/framework that you are using for requirements engineering?
2.5	Are you using tools/templates for requirements engineering? If yes: What tools/templates are you using? If no: Did you considered using tools/templates? Why (not)?

3. Gathering of requirements

3.1	Which techniques are you using to gather information from these stakeholders? (Interviews, case scenarios, joint applications development etc…)
3.2	Which other sources and techniques do you use to gather information from? (Available documentation, competitor analysis, market research)
3.3	What do you document and what not?
3.4	How are requirements documented? What information and attributes are documented about the requirements?
3.5	How do you choose the stakeholders for a project?

4. Requirements negotiations

4.1	How do you involve the stakeholders during the project?
4.2	How do you solve conflicting requirements?
4.3	How do you determine if the suggested requirements/input from the stakeholder is right?
4.4	Do you think it is important that all the stakeholders understand the requirements? Why?

5. Prioritization and validation of requirements

5.1	How do you determine which requirements are critical and which are not?
5.2	How do you validate the information from the stakeholders?
5.3	How is it decided what to include in the product? How are the requirements prioritized? What is difficult when deciding what to include in the product?
5.4	What kinds of dependencies between the requirements have you come across? Are dependencies documented? Are dependencies actively looked for?

6. General requirements questions

6.1	How many requirements are handled in a typical project? Who suggests the requirements?
6.2	What is the most time consuming part in the requirements process?
6.3	Is it possible to make decisions too late? What can be the effect in that case?
6.4	Have you ever had that project deadlines were delayed? What were the main reasons for that?
6.5	Have you ever considered using a collaboration system for requirements engineering? Why (not)?

7. Case scenarios

7.1	Sometimes there are “wicked problems” in projects, which means that requirements are constantly changing. How would you address this problem?
7.2	What was the worst project that you had to do to gather requirements? What did you do?

Appendix D: Direct observation – Field Evaluation

Location: _____
Report date: _____
Observation period: From: ____- __ - __ To: ____- __ - __
Project _____
Participants: _____
_____

1. Are decision documented?
   ☐ yes ☐ no

2. Are requirements requests from stakeholders documented?
   ☐ yes ☐ no

3. Do stakeholders use templates to formulate their requirements?
   ☐ yes ☐ no

4. Do stakeholders use a collaboration system?
   ☐ yes ☐ no

5. Do stakeholders use cognitive tools to generate an overall understanding of the requirements and the system?
   ☐ yes ☐ no

6. Is the group of stakeholders diverse?
   ☐ yes ☐ no

7. Does the requirements engineer use prioritization mechanisms?
   ☐ yes ☐ no

8. Does the requirements engineer use automated validation mechanisms?
   ☐ yes ☐ no

9. Are the stakeholders consistent in explaining and defining their requirements?
   ☐ yes ☐ no

10. Doe the requirements engineer use interviews to gather the requirements from the stakeholders?
    ☐ yes ☐ no

11. Is the requirements engineering done iteratively?
    ☐ yes ☐ no

12. Number of stakeholders: ___

13. Number of requirement concerns: ___

Requirement concerns:

_____

_____

_____

To what extend do stakeholders participate during the requirements engineering process?

Not at all				A great deal
1	2	3	4	5

Actions that the requirements engineer took to solve conflicting demands:

Actions that the requirements engineer took to gather the requirements:

Actions that the requirements engineer took to prioritization and validate the requirements:

Actions of the stakeholders during the negotiation process of the requirements:

Other observations:

Appendix E: Direct observation – Overall field Evaluation

Report date: _____
Observation period: From: ____- __ - __ To: ____- __ - __
Project _____

1. Total number of stakeholders: ___

2. Total number of requirement concerns: ___

3. Total number of iterations: ___

4. Total number of meetings: ___

5. Total number of iterations per requirement concern:

No.	Requirement concern	Iterations
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

6. Stakeholders participating in meetings

Stakeholder	Stakeholder function	Meetings
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

12		
13		
14		
15		

Appendix F: Interviews

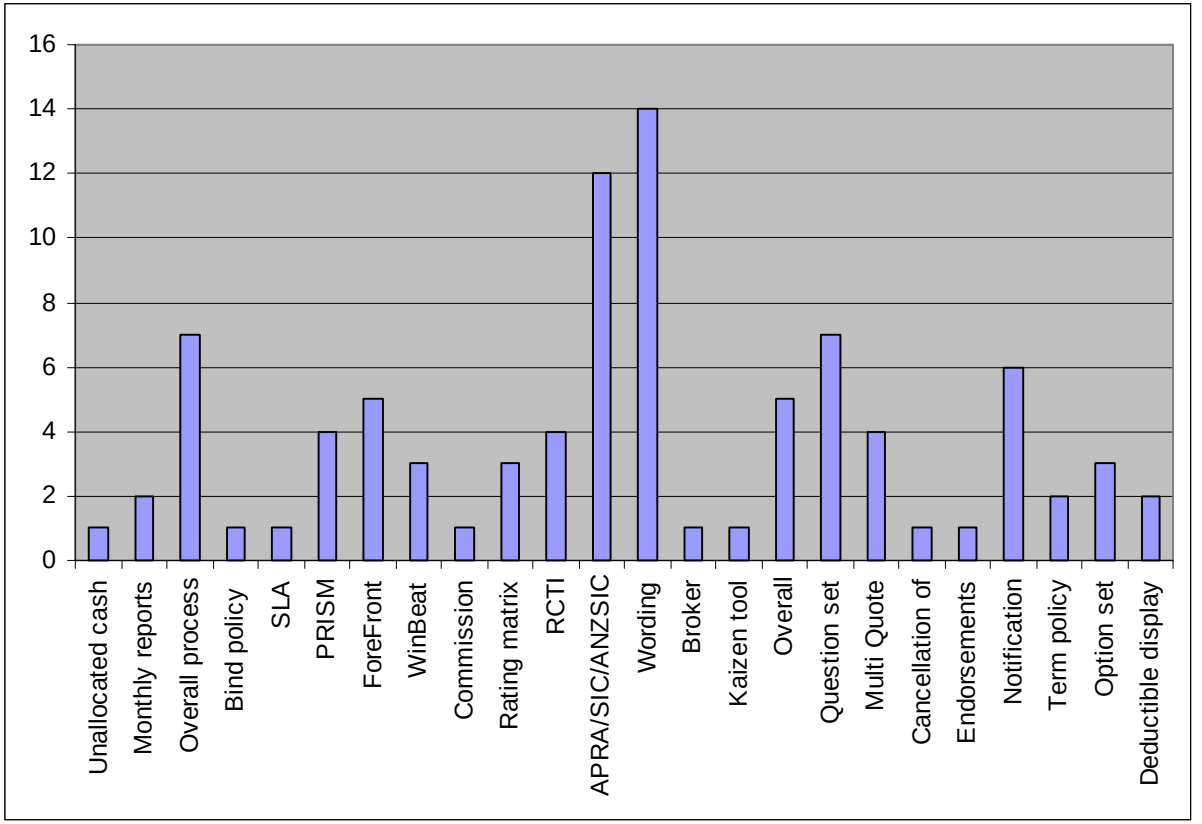
CONFIDENTIAL

Appendix G: Direct observations: Iterations per requirement concern

Iterations per requirement concerns:

No.	Requirement concern	Iterations
1	Unallocated cash	1
2	Monthly reports	2
3	Overall process	7
4	Bind policy	1
5	SLA	1
6	PRISM	4
7	ForeFront	5
8	WinBeat	3
9	Commission	1
10	Rating matrix	3
11	RCTI	4
12	APRA/SIC/ANZSIC	12
13	Wording	14
14	Broker engagement	1
15	Kaizen tool	1
16	Overall development (validation)	5
17	Question set	7
18	Multi Quote	4
19	Cancellation of policy/quote	1
20	Endorsements	1
21	Notification handling	6
22	Term policy	2
23	Option set	3
24	Deductible display	2

Average amount of iteration: **3,21**



Addressed concerns per meeting

Meeting No	Meeting	Requirement concerns
1	25-2-2011	Unallocated cash Monthly reports Overall process Policies binding SLA
2	16-2-2011	Overall process PRISM ForeFront WinBeat Monthly reports Commission Rating matrix RCTI
3	18-2-2011	RCTI APRA/SIC WinBeat Clausing/Wording Overall process
4	21-2-2011	RCTI Clausing/Wording Overall process
5	22-2-2011	WinBeat Clausing/Wording PRISM Broker engagement
6	7-3-2011	Clausing/Wording PRISM RCTI
7	17-3-2011	PRISM
8	22-3-2011	Overall process Wording ForeFront Rating Kaizen tool
9	28-3-2011	Overall process Wording ForeFront
10	5-4-2011	Overall process Wording ForeFront

Meeting No	Meeting	Requirement concerns
13	12-5-2011	Questions
14	16-5-2011	SIC / APRA
15	17-5-2011	SIC / APRA Wording Multi quote Question set
16	24-5-2011	SIC / APRA Wording
17	2-6-2011	SIC / APRA Multi quote
18	8-6-2011	Questions Wordings Multi Quote SIC / APRA Cancellation of policy Endorsements Notification handling Declined / referral handling
19	16-6-2011	Questions set Wordings SIC / APRA Declined / referral handling Rating 10m option
20	20-6-2011	Questions set Wordings (exclusions & endorsements) Options
21	22-6-2011	Questions set Wordings (exclusions & endorsements) Term policy SIC/ANSIC (occupation categories) Referral message Deductible Option set
22	27-6-2011	Term policy SIC/ANSIC (occupation categories) Referral message

11	29-4-2011	APRA/SIC Overall development / validation
12	5-5-2011	APRA/SIC Overall development / validation Wording Questions ForeFront Multi Quote

		Deductible Option set
23	28-6-2011	SIC/ANSIC (occupation categories)

Appendix H: Meetings notes of 24-05-2011 and 22-06-2011

CONFIDENTIAL

Appendix I: Attendance of participants during prototype meetings

CONFIDENTIAL

Appendix J: Software development life cycle documentation

CONFIDENTIAL

Appendix K: Direct observations – Overall field evaluation

Report date: 28-06-2011
Observation period: From: 06- 02 -2011 To: 01- 07 - 2011
Project MFT project - iClose

1. Total number of stakeholders: 11

2. Total number of requirement concerns: 24

3. Total number of iterations: 6
    a. Meeting 16-02-2011
    b. Meeting 22-02-2011
    c. Meeting 29-04-2011
    d. Meeting 05-05-2011
    e. Meeting 17-05-2011
    f. Meeting 16-06-2011

4. Total number of meetings: 24

5. Total number of iterations per requirement concern:

No.	Requirement concern	Iterations
1	Unallocated cash	1
2	Monthly reports	2
3	Overall process	7
4	Bind policy	1
5	SLA	1
6	PRISM	4
7	ForeFront	5
8	WinBeat	3
9	Commission	1
10	Rating matrix	3
11	RCTI	4
12	APRA/SIC/ANZSIC	13
13	Wording	15
14	Broker engagement	1
15	Kaizen tool	1
16	Overall development (validation)	5
17	Question set	8
18	Multi Quote	4
19	Cancellation of policy/quote	1
20	Endorsements	1
21	Notification handling	7
22	Term policy	2
23	Option set	3
24	Deductible display	2

Stakeholders participating in meetings

Stakeholder	**Stakeholder function**	**Meetings**
1	Business Analyst	24
2	Project Manager	22
3	CFO	11
4	IT Manager	3
5	Developer 1	9
6	Developer 2	3
7	Expert (National OSD Manager)	2
8	Expert (Independent Broker Manager)	6
9	Expert (Business Analyst of other project)	1
10	Expert (Underwriter)	2
11	Expert (National Product Manager)	5

Prioritization in meetings

Meeting no.	Type
2	Benefit/Cost
5	Benefit/Cost
11	Benefit/Cost
14	Benefit/Cost
15	Benefit/Cost
18	Benefit/Cost
19	Benefit/Cost
22	Benefit/Cost / Scope
7	Scope
12	Scope
24	Scope

Appendix L: Outstanding issues from meeting 16-02-2011 and 22-02-2011

CONFIDENTIAL

Appendix M: Business specification document

CONFIDENTIAL

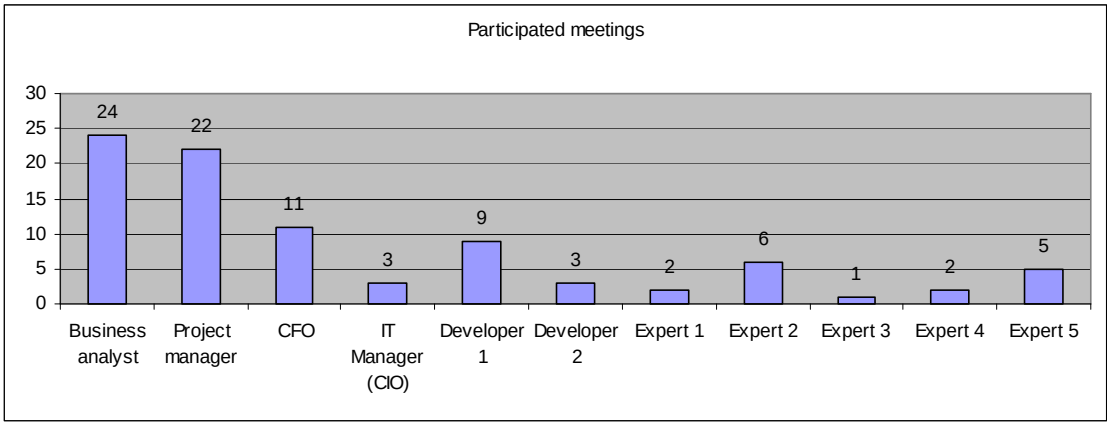
Appendix N: Business specification document

CONFIDENTIAL

Appendix O: Involvement of stakeholders

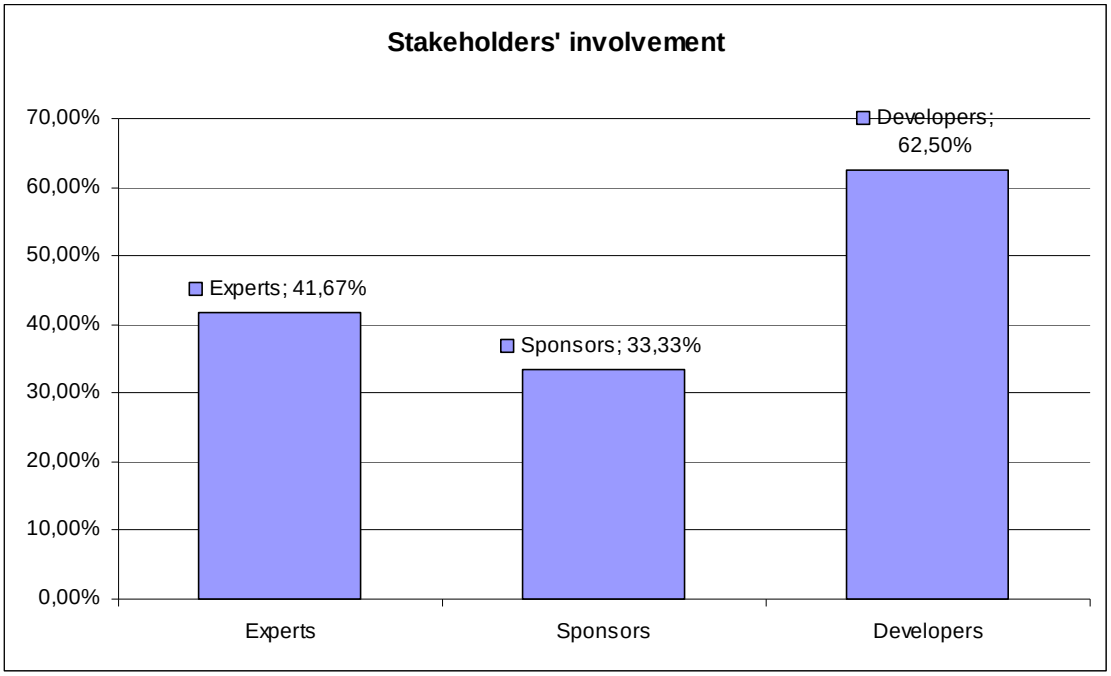
Participated meetings per stakeholder

Stakeholder	Participated meetings	Type
Business Analyst	24	Business Analyst
Project Manager	22	Project Manager
CFO	11	Sponsor
IT Manager (CIO)	3	Expert
Developer 1	9	Developer
Developer 2	3	Developer
Expert 1	2	Expert
Expert 2	6	Expert
Expert 3	1	Expert
Expert 4	2	Expert
Expert 5	5	Expert



Involvement in the meetings per stakeholder type

Stakeholder type	Involvement in %	Amount of meetings involved in
Experts	41,67%	10 of 24
Sponsors	33,33%	8 of 24
Developers	62,50%	15 of 24



Appendix P: Marker research questionnaire

CONFIDENTIAL

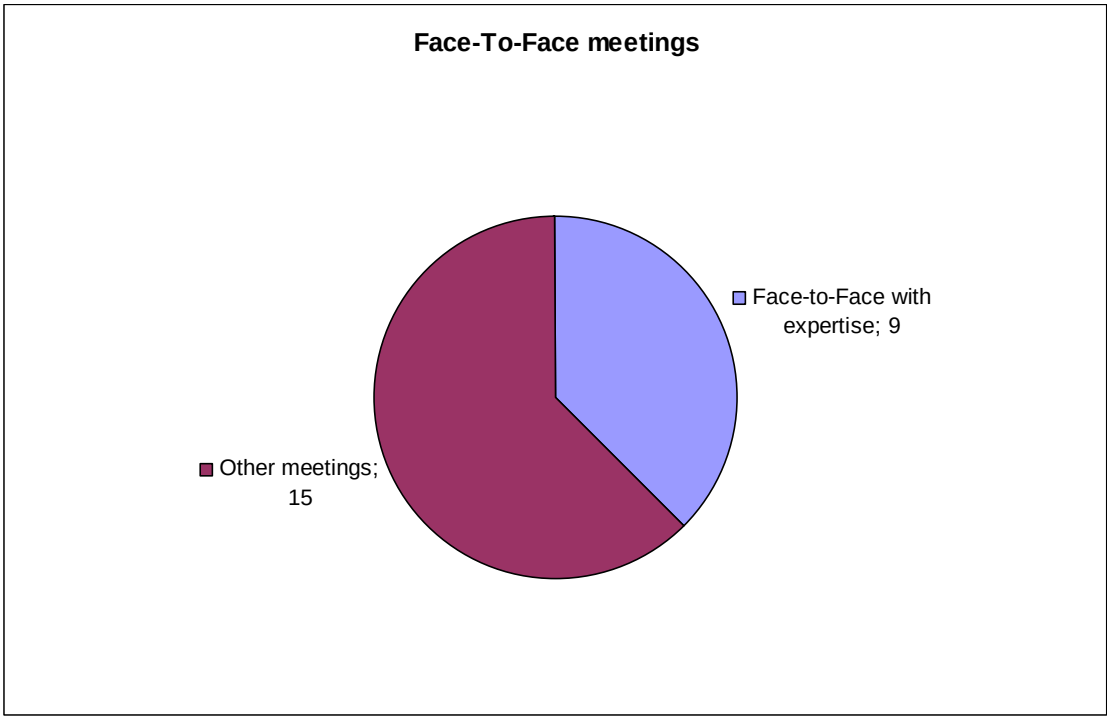
Appendix Q: Screenshots of competitor's systems

CONFIDENTIAL

Appendix R: Observations on face-to-face meetings

Meetings

Amount of meetings	Face-to-Face with expertise	Other meetings
24	9	15

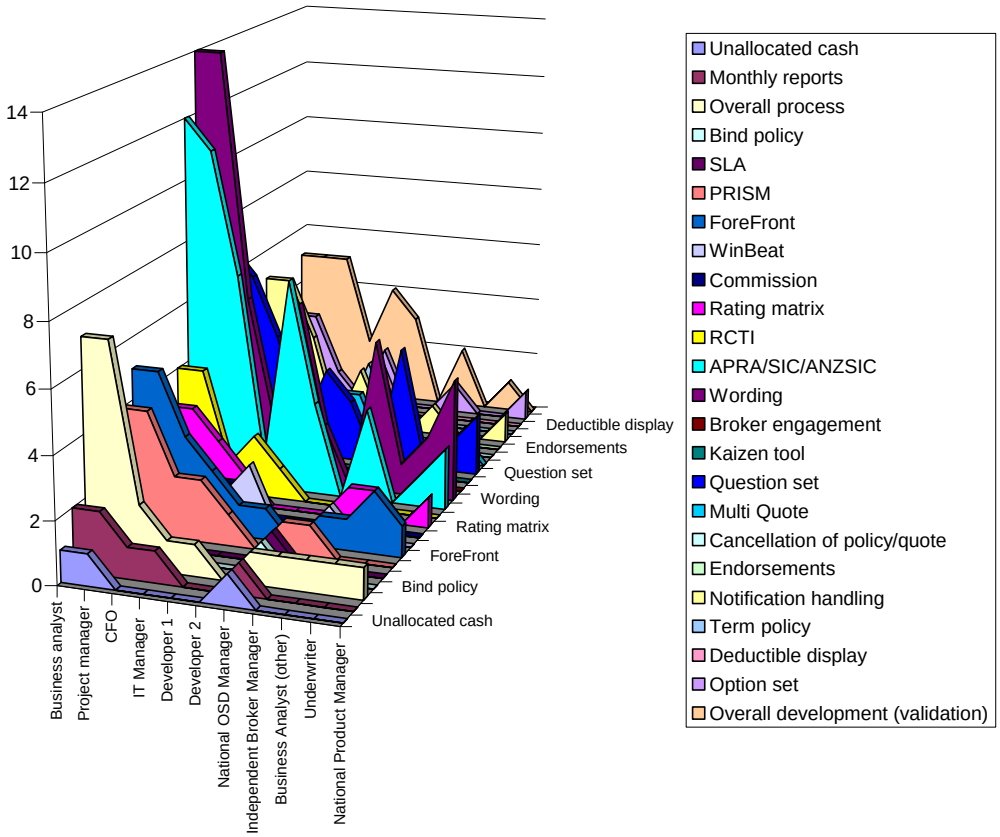


Appendix S: Meetings notes of face-to-face meetings with experts

CONFIDENTIAL

Appendix T: Number of iterations per stakeholder per requirement concern

	Business Analyst	Project Manager	CFO	IT Manager	Developer 1	Developer 2	National OSD Manager	Independent Broker Manager	Business Analyst (other)	Underwriter	National Product Manager	**Total**
Unallocated cash	1	1					1					2
Monthly reports	2	2	1	1			1					5
Overall process	7	7	2	1	1		1	1	1	1	1	16
Bind policy	1	1					1					2
SLA	1	1					1					2
PRISM	4	4	2	2	1		1	1				11
ForeFront	5	5	3	2	1	1		1	1	2	1	17
WinBeat	3	3	2	1	2			1				9
Commission	1	1	1	1								3
Rating matrix	3	3	2	1				1	1		1	9
RCTI	4	4	1	2	1							8
APRA/SIC/ANZSIC	12	11	7	1	7	3		3		1	2	35
Wording	14	14	6	2	6	2		5	1	2	4	42
Broker engagement	1	1	1		1			1				4
Kaizen tool	1	1							1			2
Question set	7	6	4	1	3	2		4		1	2	23
Multi Quote	4	4	3	1	2	2		1		1		14
Cancellation of policy/quote	1	1	1									2
Endorsements	1	1	1									2
Notification handling	5	5	3		2			1			1	12
Term policy	2	2			2							6
Deductible display	2	2			2							6
Option set	3	3	1		2			1			1	11
Overall development (validation)	5	5	5	2	4	3		2		1		22
Total	90	88	46	18	37	13	6	23	5	9	13	


14
12
10
8
6
4
2
0
Business analyst
Project manager
CFO
IT Manager
Developer 1
Developer 2
National OSD Manager
Independent Broker Manager
Business Analyst (other)
Underwriter
National Product Manager
Unallocated cash
Bind policy
ForeFront
Rating matrix
Wording
Question set
Endorsements
Deductible display
Unallocated cash
Monthly reports
Overall process
Bind policy
SLA
PRISM
ForeFront
WinBeat
Commission
Rating matrix
RCTI
APRA/SIC/ANZSIC
Wording
Broker engagement
Kaizen tool
Question set
Multi Quote
Cancellation of policy/quote
Endorsements
Notification handling
Term policy
Deductible display
Option set
Overall development (validation)

Appendix U: Observations No. 13 and 24

Direct observation No. 13 – Meeting about question set with the Independent Broker Manager

Location:	Chubb – Office Independent Broker Manager	
Report date:	16-05-2011	
Observation period:	From: 2011- 05 – 12 10:00	To: 2011- 05 – 12 10:30
Project	iClose – questions set discussion	
Participants:	Business Analyst Independent Broker Manager	

14. Are decision documented?
    Yes

15. Are requirements requests from stakeholders documented?
    Yes

16. Do stakeholders use templates to formulate their requirements?
    No

17. Do stakeholders use a collaboration system?
    No

18. Do stakeholders use cognitive tools to generate an overall understanding of the requirements and the system?
    No

19. Is the group of stakeholders diverse?
    No, only business people no developers

20. Does the requirements engineer use prioritization mechanisms?
    No

21. Does the requirements engineer use automated validation mechanisms?
    No

22. Are the stakeholders consistent in explaining and defining their requirements?
    Yes, based on the question set of the requirements document

23. Doe the requirements engineer use interviews to gather the requirements from the stakeholders?
    No

24. Is the requirements engineering done iteratively?
    Yes

25. Number of stakeholders: 2

26. Number of requirement concerns: 1

Requirement concerns:

- Questions

To what extend do stakeholders participate during the requirements engineering process?

Not at all				A great deal
1	2	3	**4**	5

Actions that the requirements engineer took to solve conflicting demands:

Other observations:

CONFIDENTIAL

Direct observation No. 24 – iClose outstanding issues

Location:	Chubb Sydney – Meeting room 4	
Report date:	30-06-2011	
Observation period:	From: 2011- 06 – 30 14:00	To: 2011- 06 – 30 15:30
Project	iClose – Outstanding issues	
Participant:	Business Analyst	
	CFO	
	Project Manager	
	National Product Manager	
	Developer	

1. **Are decision documented?**
   Yes

2. **Are requirements requests from stakeholders documented?**
   Yes

3. **Do stakeholders use templates to formulate their requirements?**
   No

4. **Do stakeholders use a collaboration system?**
   No

5. **Do stakeholders use cognitive tools to generate an overall understanding of the requirements and the system?**
   No

6. **Is the group of stakeholders diverse?**
   Yes

7. **Does the requirements engineer use prioritization mechanisms?**
   Yes, time and budget

8. **Does the requirements engineer use validation mechanisms?**
   Yes, input from experts

9. **Are the stakeholders consistent in explaining and defining their requirements?**
   No

10. **Doe the requirements engineer use interviews to gather the requirements from the stakeholders?**
    No

11. **Is the requirements engineering done iteratively?**
    Yes

12. Number of stakeholders: 5

13. Number of requirement concerns: 4

Requirement concerns:
- SIC/ANSIC (occupation categories)
- Question set

- Wording
- Referral / declined

To what extend do stakeholders participate during the requirements engineering process?

Not at all				A great deal
1	2	3	**4**	5

Actions that the requirements engineer took to solve conflicting demands:
Wrote it down for further investigation, for example the PI coverage if it should be included or not. The RE will get back on this.

Actions that the requirements engineer took to gather the requirements:
Ask experts and developer about what need to be in the system

Actions that the requirements engineer took to prioritization and validate the requirements:
Validated with other experts to understand and see if it is actually necessary to build into the system. The developer gave his opinions if it will take extra time.

Actions of the stakeholders during the negotiation process of the requirements:
Stakeholders gave their opinions about the decisions of the RE. They said if it should be included or not.

Other observations:
CONFIDENTIAL

Appendix V: Scope of iClose project

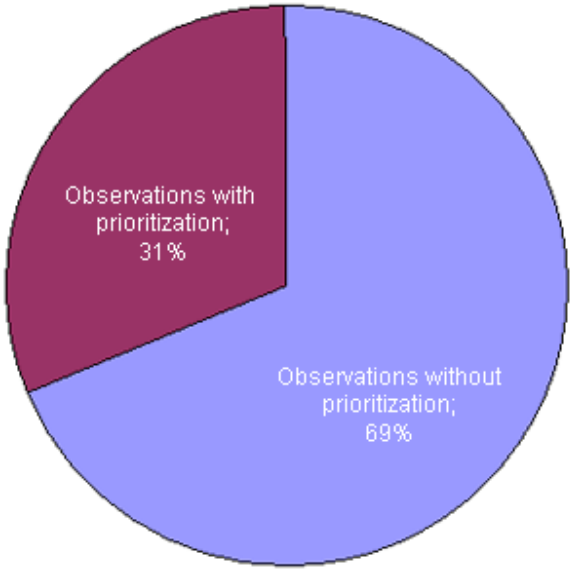
CONFIDENTIAL

Appendix W: Prioritization during meetings

Prioritization during meetings

Meeting No	Prioritization
2	Benefit/Cost
5	Benefit/Cost
11	Benefit/Cost
14	Benefit/Cost
15	Benefit/Cost
18	Benefit/Cost
19	Benefit/Cost
22	Scope
7	Scope
12	Scope
24	Scope

Amount of observations	Amount of prioritization
24	11



Main prioritization mechanisms

Benefit/Cost	Totaal
Benefit/Cost	7
Scope	4
Eindtotaal	11