Customer Involvement in Distributed Requirements Engineering

Master of Science Thesis

Version 8 - November 27, 2009

Christian K. Visser

Customer Involvement in Distributed Requirements Engineering

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Christian K. Visser
born in Den Helder, the Netherlands

Software Engineering Research Group
Department of Software Technology
Faculty EEMCS, TU Delft
Delft, the Netherlands
www.ewi.tudelft.nl

Corporate Product Management
Exact International Development
Poortweg 6, Floor 5
Delft, the Netherlands
www.exact.com

Cover image:
Faculty of Electrical Engineering, Mathematics and Computer Science

Customer Involvement in Distributed Requirements Engineering

Author: Christian K. Visser
Student id: 1100572
Email: viss247052@exactsoftware.com

Abstract

The graduation project is run as an internship at software company Exact. The company wants to reduce the time to market for their products. To achieve this objective, Exact wants to adopt agile development and is aiming towards the implementation of a development process that is both globally distributed and Scrum-like. The project's goal in this objective is to design and validate a process, with which a set of product requirements is engineered and in parallel the products design is being developed at at least two distributed locations. This process needs to iterate between these locations, but at the same time it needs to work towards a non-ambiguous, concrete and focused backlog with which development can do their work. Based on interviews and constraints, the project was scoped down to focus on the verification of requirements with customers. After research in this area, a prototype is created and tested, which should establish a feedback loop on conceptual requirements items in a virtual community. The test results have led to a set of recommendations for customer involvement in a distributed requirements engineering process.

Thesis Committee:

Chair:	Prof.dr. A. van Deursen, Faculty EEMCS, TU Delft
University supervisor:	Dr.ir. D.M. van Solingen, Faculty EEMCS, TU Delft
Company supervisor:	Drs. R. Voets, Corporate Product Management, Exact
Committee Member:	Ir. B.R. Sodoyer, Faculty EEMCS, TU Delft

Preface

Sitting here behind my desk at Exact's Corporate Product Management, I have a perfect view on my educational home for many years. In fact, the view from the office reflects the entire life-cycle of my Computer Science programme at Delft University of Technology; from the old master building where I've spend my first years, to the faculty of EEMCS where the last part resided. This makes Exact's headquarters the perfect spot to do my final internship!

At this point I would like to express my gratitude to Menno, for all the hard work in collaboration between our projects and the frequent phonecalls with Kuala Lumpur; to Rini, for the mental support and critical feedback from university; to Emile, Ronald and Toine, who guided me through all facets of Exact; to all guys at Corporate Product Management, for feedback and quality time in the daily work at the office; to Nelly, for all general support; to the lunch team, for the diversion I sometimes heavily longed for; and, not in the least, to all others that helped me out in one way or another. Without you guys and girls, this graduation project would not have succeeded!

To keep up the tradition I started with my MSc Research Report, I will conclude this preface with a quote. The following wisdom has been expressed by John F. Kennedy:

> *"As we express our gratitude, we must never forget that the highest appreciation is not to utter words, but to live by them."*

Christian K. Visser
Delft, the Netherlands
November 27, 2009

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, the context of the graduation project will be introduced. An introduction on distributed agile software development will also be shared. After that, the project goal will be formulated, together with its constraints.

1.1 Context

The graduation project is run as an internship at the software company Exact. Exact was established in 1984 in Delft, the Netherlands. Today, corporate headquarters are still located in Delft, and the company has offices in more than 40 countries in Europe, the Middle East, North, Central and South America, Africa, Asia and Australia. A global network of distributors and resellers gives advice, sells products and offers support to customers.

Exact provides solutions to small and medium-sized businesses, and subsidiaries of multinationals, with diverse implementation and industry requirements ranging from manufacturing, distribution and retail to trade and service environments. To efficiently accommodate the varied needs worldwide, Exact has structured its expanding global network into four regions: APAC, EMEA, the Americas and the Netherlands. While each region serves its own distinct market in a different way, they all share a commitment to a single solution offering. All linked by a single infrastructure, they provide access to the same customer database and information anytime and anywhere, thereby ensuring a high level of customer service around the world.

Exact also has its development sites throughout different locations around the globe. Reasons for this are product diversity, resource availability, labour cost and acquisitions. Currently, Exact is reorganising and standardising its development processes globally, with the objective to create a development organisation which is able to deliver products in time, in cost and in quality; independent of the location in which development work has taken place.

Exact's corporate products have been developed with a release cycle of one year. For every new release requirements are gathered during the current release and implemented in the next. As a result, the average time to market of new functionality from a customer point

of view was 1.5 to 2 years. Today, release cycles have been shortened to one to two every half year, meaning time to market ranges from six months to about one year at this moment.

At Exact, requirements originate from multiple locations around the globe while requirements development is at one location; Delft. Furthermore, product design and product development take place at different locations. Finally, although requirements development and product design are not co-located, the end-result needs to be one single and concrete backlog for product development.

Exact wants to reduce the time to market for their products even further. To achieve this objective, Exact is aiming towards the implementation of a development process that is both globally distributed and agile. In order to do so, development processes need to be aligned, development skills need to be improved, and the organisation needs to become both flexible as well as in control.

The current corporate product development process at Exact is called the Refined Development Process (RDP). The RDP is built upon several pillars [22]:

- Iterative
- Agility
- Efficiency
- Team responsibility
- Storyboards - a picture paints a thousand words
- Differentiators
- Small projects drive results
- Monitoring

With the introduction of the RDP at Exact, the company started to focus more on iterations and feedback. To enhance this focus, several phases in the process overlap to facilitate collaboration. This can also be seen from Figure 1.1. The phases in the RDP will be explained briefly in the following list.

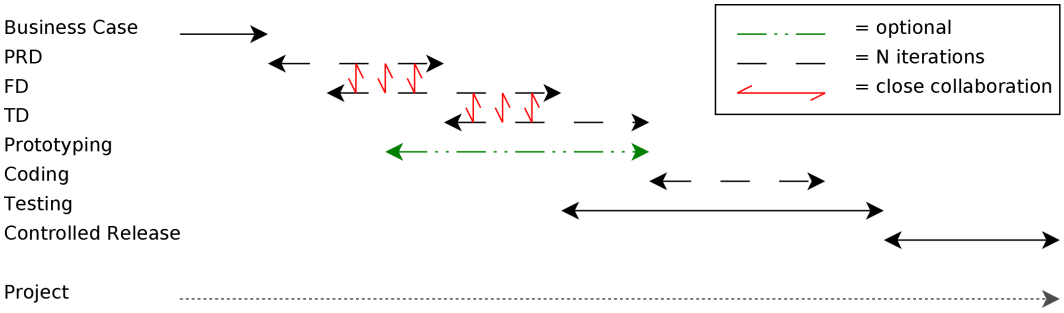

Figure 1.1: Overview of the Refined Development Process

Business Case Input from the regions on suggestions and requirements

PRD Creation of detailed requirements in the Product Requirements Document

FD Creation of the Functional Design of the product

TD Creation of the Technical Design of the product

Prototyping Verification of the designs

Coding Implementation of the requirements

Testing Test of the product in a test environment

Controlled Release Test of the product in a live environment at selected customers

In current business, Exact is continuously improving its processes. This is a challenge for external advisors who want to help to improve a process, because they have to 'jump on a moving train'. They could verify already performed actions with theory, or propose additional enhancements, but chances are small that they can find a subject that has not been reviewed at all. However, introducing agility in a distributed environment brings in more challenges.

1.2 Distributed Agile Software Development

Before diving into the agile world, some explanation on the field of requirements engineering would be appropriate. Requirements engineering is the first phase within software engineering, and involves all requirements handling activities. According to the Software Engineering Body of Knowledge (SWEBOK, [2]), requirements engineering can be split up into four subphases: elicitation, analysis, specification, and validation. While requirements engineering is the first phase, it is not by definition also finished first. Actually, the requirements phases are active throughout the entire software lifecycle, in which the activities evolve from elicitation to change management [2].

In the last decade, a new concept was added to the field of software engineering; agile software development. This concept contains multiple approaches, among which Extreme Programming and Scrum. The main characteristics of these approaches are similar; incremental, cooperative, straightforward, and adaptive [1]. Agile approaches are supposed to be light-weight; one of the biggest differences with traditional software engineering is the importance of documentation. In agile approaches, documentation is only done to support coding activity [13].

In earlier refinements of the development process, Exact implemented elements of the Scrum approach. Scrum is one of the best known agile approaches and focuses on the coordination of work, rather than on the software implementation itself [23]. Scrum can be implemented as a layer on top of existing development processes [41], and thus does not require fundamental organizational changes for adoption.

To get more clarity in the agile world, Dullemond et al. have derived a set of agile aspects, which *"denote the goals agile software development attempts to accomplish"* [16]. These aspects can be found in Table 1.1.

#	Description
A_1	Close collaboration among the members of the development team
A_2	Short iterations, frequent builds and continuous integration
A_3	Decentralizing the decision making
A_4	Customer involvement
A_5	Collective ownership of work
A_6	The system to be built is most important
A_7	Favoring simplicity
A_8	Sustainable pace of development

Table 1.1: Agile aspects [16]

Introducing agility in a distributed development environment brings along some challenges. To begin with, there is a mismatch in the importance and usage of documentation. Exact's geographical separation between requirements development (Delft, the Netherlands) and product design (Kuala Lumpur, Malaysia) limits communication and thus results in a cut in the development process. This makes the process formal and document-oriented, while agile processes mostly depend on informal practices [38]. Allen acknowledges this, by reporting that when team members are separated more than 30 meters apart, their level and type of communication drops to a similar low level as when communicating with colleagues at the other side of the world [3].

For introducing agility in general, Schatz and Abdelshafi identify the obstacles of needing to have *"potentially shippable"* software increments at the sprint review, focusing on short-term deliverables, while loosing sight of long-term maintainability, and having a lack of metrics for the stakeholders to estimate the completion date [41].

In research on distributed Scrum, Sutherland et al. share the following challenges [45]:

- Cultural differences
- Sharing context and priorities
- Managing customers new to agile
- Some work is local
- Tooling for communication and process

Diving further into the field of distributed Scrum, Berczuk shares that the daily stand-up meeting is difficult when distributed [7]. Sinha et al. conclude that distributed projects need more investment in the early requirements engineering phase to make the requirements clear enough for distribution [44]. They also say that for making distributed projects work, a *"healthy culture of collaboration"* is needed. Finally, Sutherland et al. claim that to reach a level of productivity equal to collocation, *"excellent implementation of Scrum and good engineering practices"* are required [46].

1.3 Project Goal

The goal of this project is to design and validate a process with which a set of product requirements is engineered and in parallel the product's design is being developed at at least two distributed locations (Delft and Kuala Lumpur). This process needs to iterate between these locations but at the same time it needs to work towards a non-ambiguous, concrete and focused backlog with which development can do their work (possibly on several distributed locations).

1.4 Constraints

The graduation project is part of a duo-assignment, established in the collaboration between Exact and Delft University of Technology. Both projects will try to improve the development process; this project will be focused on the requirements engineering perspective, while the other project will have its focus on the product development perspective.

There are some constraints that apply to the projects. These constraints originate from both the company and the university.

1.4.1 Exact

- All recommendations for process improvement should be focused on making the development process more agile
- For the proof of concept, the credo for the recommendations is:
  Minimal change, Maximal effect

1.4.2 Delft University of Technology

- All recommendations and research subjects should be related to the Computer Science programme
- Both parts of the duo-assignment should be as close together as possible

1.5 Thesis Overview

In the following chapters will be described how the initial project goal was scoped down based on a series of interviews. Subsequent chapters will contain a report of the outcome of the project; a prototype and proof of concept. The thesis will conclude with a discussion of the findings and recommendations, an evaluation of the project and a vision for the future.

Chapter 2

Interviews

To get a view of the current situation in the development process of Exact, a series of interviews is conducted. In this chapter, the interview series will be described, along with its findings and conclusion.

2.1 Interview Target

The interview questions were targeted on which methods are used, which bottlenecks are occurring, and whether the perception of the process differs between the various layers of the organization, and between theory and practice. As such, the interviews were meant to (a) check whether the development process in practice conforms to the process description and (b) reveal any bottlenecks and points of attention.

2.2 Interview Style

The interview series was set up according to the qualitative research style. This research style has a greater interest in the interviewee's point of view, rather than in the researcher's concerns as is the case with its alternative; quantitative research [10]. In qualitative interviewing, interviewers are not tied to any schedule or guide that is being used. They want rich, detailed information and may thus come up with new questions during the interview and even rephrase and reorder the questions in the guide. Quantitative research is about maximizing the reliability and validity of the measurement of the subject at study. Therefore, quantitative interviewing uses strict, structured interview guides. Because in this interview series the stories and feedback of the interviewees were valued most, qualitative research was chosen as interview style.

In qualitative research, the major interview types are *unstructured* and *semi-structured* [10]. In unstructured interviewing, a small set of notes with high-level subjects is used, whereas semi-structured interviewing uses a list of detailed questions to be covered. Semi-structured interviewing is more likely when the investigation begins with a fairly clear focus on the subject [10]. Because that was the case for this assignment, the interviews were of the semi-structured type.

2.3 Interview Guide

For the selection of interviewees, the direct stakeholders for requirements engineering were extracted from the description of the current development process. The stakeholders were matched with job titles from the organization. Because the scope of the interviews was limited to the Exact development process, only Exact employees were selected. The interviewees worked in the following departments:

- Product Management
- Product Marketing
- Research & Innovation
- Product Development
- Regions EMEA & NL
- Exact Online

To represent the input from the regions, employees were selected from the two largest regions; EMEA and NL. Exact Online is one of the corporate products. That department was mainly included to compare the differences in development processes within Exact, as development for Exact Online is currently still colocated. A complete list of interviewees can be found in Appendix C.

An introduction and invitation for the interviews was sent by email, after which the interview was planned. The interviews were scheduled for one hour, with a possibility to extend the timeframe when the participants' schedules had room for it. The interviews took place in a separate meeting room, to avoid distraction and to enable the interviewee to talk freely.

The interviews could not be recorded, due to the inavailability of such hardware. Therefore, notes were made during the interviews, which were translated into reports shortly after each interview. These reports were validated with each corresponding interviewee, to make sure that the report reflected the interview correctly. After this review, the interview results were frozen.

In total, 20 semi-structured interviews were conducted in a time frame of 4.5 weeks. The duration of the interviews varied from 30 minutes to 1.5 hour. One interview was conducted by phone, the others were all colocated.

2.4 Validity

To preserve the validity of the results and conclusions from the interviews, some measures were taken.

First of all, the interview questions were mapped onto the Software Engineering Body Of Knowledge (SWEBOK, [2]), to make sure that all requirements engineering stages and points of attention were covered. The SWEBOK is a baseline for the body of knowledge

in the field of software engineering, established by the IEEE Computer Society. The list of questions is enclosed in Appendix D, together with a mapping on the relevant SWEBOK entities.

Furthermore, each interview report was verified with the corresponding interviewee, to check its representation of the interview. Any changes in a report were taken into account at concluding the interview series.

2.5 Findings from the Interview Series

The findings from the interviews are separated in (a) the overall perception of the development process and (b) the bottlenecks and points of attention that have been mentioned.

2.5.1 Perception of the Process

Comparing the reports of the various interviews showed that the perception of the development process did not vary greatly between the different management layers or functional groups in the organization. As a matter of fact, the overall structure of the communication flow from customer to development as extracted from the interviews was similar to the development process description. Figure 2.1 shows a visualization of the process in practice.

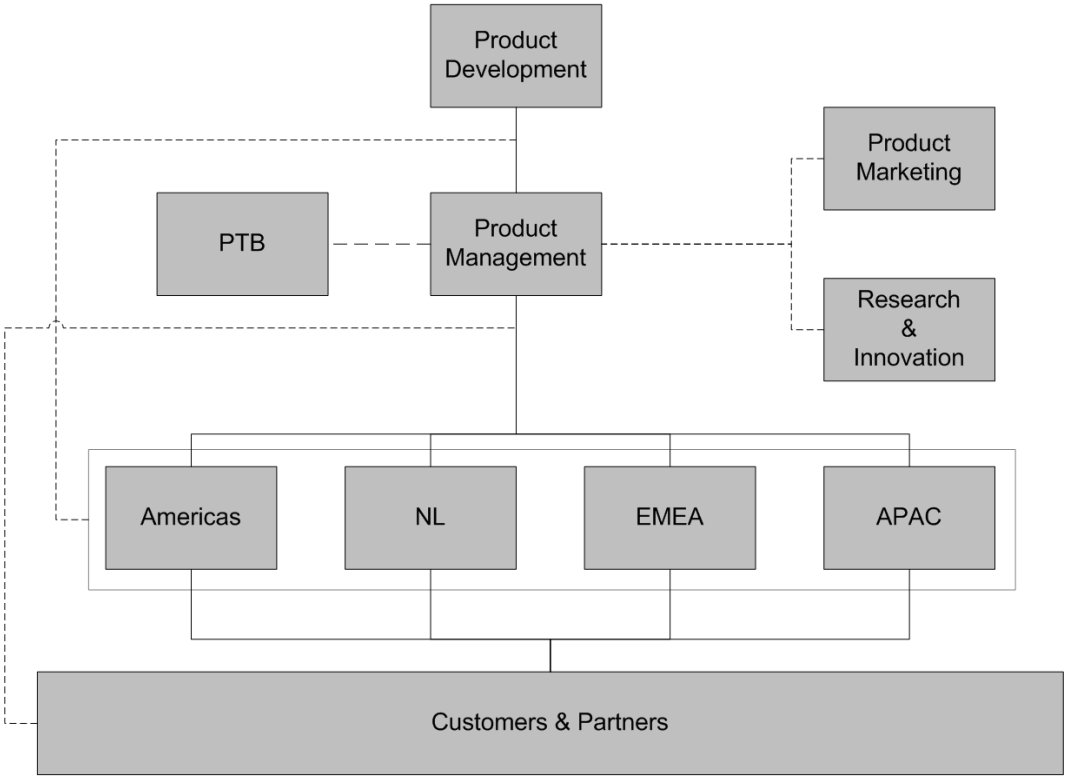

Figure 2.1: Communication flow in practice

The dotted lines indicate incidental communication. This holds for the interaction between Product Management and Product Marketing and that between Product Management and Research & Innovation, as well as for the by-passes around Product Management and the regions.

The dashed line indicates the special interaction between Product Management and the Product Technology Board, which is only consulted to get approval for big projects.

2.5.2 Bottlenecks

The bottlenecks that have been extracted from the interviews are listed in the table below, along with an indication of the originating department. The bottlenecks are grouped by subject.

Table 2.1: Bottlenecks from the interviews

Description	Regions	PM	PMar	R&I	PD
Requirements Elicitation					
Understanding requirements by PM at decision-making can be improved		✓			
Requirements quality can be improved		✓			
Requirements Management					
No structure for suggestion management (incoming requirements)	✓	✓			
Prioritization may be an issue		✓	✓		
Managing sprint backlogs is difficult		✓			
Little structure around requirements		✓			
Planning					
Requirements elicitation runs late on planning for PM		✓			
Focus PM mostly on requirements (RECORD task description)		✓			✓
Process description is not always followed correctly by PM		✓			
Production too much focused on releases		✓	✓		✓
Documentation - Business Case					
Not all suggestions reach PM, because BC creation is too heavy	✓	✓	✓		
Project size measurement is difficult without link to Development	✓				
Measurement of success afterwards is difficult		✓			
Documentation - PRD					
Extracting sprint backlogs is difficult		✓			
Conceptual model misses as link between requirements and FD			✓		

Continued on next page

Table 2.1 – Continued from previous page

Description	Regions	PM	PMar	R&I	PD
Creation is (too) heavy on time		✓			
Overlap in PRD and FD is too large		✓			
Difficult to describe visual elements		✓			
Usability is not really incorporated in the process		✓	✓		
Product Strategy					
Too much focus on current product				✓	
Too little focus on growth perspective / market opportunities			✓	✓	
Products do not evolve				✓	
Long term vision is missing (at least in lower layers)	✓	✓			
Communication - PM Internal					
Too little knowledge sharing		✓			
Too little group feeling		✓			
Too little creative collaboration		✓			
Little structure around communication		✓			
Communication - PM External					
Too little communication (on functionality) with customers/regions	✓	✓			✓
Requirements are mostly only checked with originating region	✓				
Released functionality may differ from requirements		✓			
Feedback to regions misses on decision making and development (planning)	✓				
Human Resources					
Too little dedicated manpower	✓	✓			

2.6 Conclusion from the Interview Series

The purpose of the interviews was two-fold; (a) to check whether the development process in practice conforms to the process description and (b) to reveal any bottlenecks and points of attention.

By checking the perceptions of the development process with the various interviewees, the theoretical description of the development process could be mapped onto the process as it is used in practice. This mapping showed little difference between theory and practice. The main communication lines - the routing of the customer's requirements - are in fact identical. Less communication with the departments Research & Innovation and Product Marketing and some incidental by-passes for clarifying specific issues are the only differences with theory. Thus, there are no critical issues for the development process in this

part.

The interviewees also shared their visions on the bottlenecks in the current process. These points of attention can be grouped into the following focus areas:

- Requirements Elicitation
- Requirements Management
- Planning
- Documentation
- Product Strategy
- Communication
- Human Resources

In the next chapter, these groups shall be used to scope down the project assignment.

Chapter 3

Project Scope

The initial scope of this project, as expressed by the project goal in section 1.3, is too broad for the timeline of a graduation project assignment. This chapter contains a description of how the project was scoped down to a more appropriate size, based on the results from the interview series and a mapping of the current development process on agile practices. After this, the final objective of the project will be formulated.

3.1 Agile practices

The interviews were conducted as a first step to scope down the project assignment. With the points of attention that resulted from the interviews, the assignment could already be narrowed down to a smaller list of focus areas.

However, one of the constraints from Exact was to only focus on agile solutions. For that purpose, agile practices were searched. Table 3.1 lists the agile practices for requirements engineering shared by Cao and Ramesh [11].

#	Description
1	Face-to-face communication over written specifications
2	Iterative requirements engineering
3	Repeated requirement prioritization
4	Managing requirements change through constant planning
5	Prototyping to validate and refine requirements
6	Test-driven development (traceability)
7	Review meetings and acceptance tests

Table 3.1: Agile practices [11]

3.2 Mapping the RDP on Agile Practices

With these practices, the RDP was reviewed to see which agile practices were not yet present in the current development process. The mapping of the RDP on the practices is given in

Table 3.2. The numbers in front correspond with the numbers in front of the practices in Table 3.1.

#	Description
1	Written specifications are still highly valued in the RDP. However, Exact recognizes the importance of face-to-face interaction and tries to maximize it in their distributed setting.
2	The RDP already has an increased focus on iterations in the process.
3	Repetition in the prioritization of requirements is not specified in the RDP.
4	Constant planning is not explicitly specified in the RDP.
5	Prototyping is present in the RDP as an optional phase. However, this phase is focused more on validating designs rather than requirements.
6	The RDP is not test-driven.
7	Review meetings and acceptance tests are present on product-level in the Controlled Release phase.

Table 3.2: Mapping of RDP onto the agile practices

Recalling the focus areas from the interview series (see section 2.6), some overlap can be found with the results of the mapping. Firstly, the issues with 'Repetition' (3) and 'Constant planning' (4) are shared with the focus area Requirements Management. Secondly, the issues with 'Prototyping' (5) and 'Acceptance tests' (7) are shared with the focus area Communication.

The final issue from the mapping is the RDP not being a test-driven process. Trying to solve this would mean to redesign the complete RDP. Given the second constraint from Exact – *minimal change, maximal effect* – this issue will be excluded from the scope.

3.3 Scoping on Focus Areas

The next step was to review the focus areas based on the possibility of agile solutions. To do this, the agile practices were mapped onto the focus areas. Table 3.3 shows the outcome. The numbers again correspond with the numbers in front of the list of agile practices (Table 3.1).

Focus area	Agile practices
Requirements Elicitation	2, 4, 5
Requirements Management	3, 4, 6
Planning	2, 4
Documentation	1, 2
Product Strategy	–
Communication	1, 2, 5, 7
Human Resources	–

Table 3.3: Mapping of agile practices onto the focus areas

The focus areas for which there are no solutions in agile practices are excluded from the scope.

One of the constraints from university was to have both parts of the duo-assignment as close together in focus area as possible. The split in the duo-assignment is at the creation of a Functional Design. Because Product Management is the link between the regions and the Design and Development departments, the constraint automatically implies that the scope should only contain focus areas that include PM as main actor. Looking at the RDP, this would mean that only the phases of PRD and FD creation apply. The focus area Requirements Elicitation did not involve PM as an actor, so that focus area is also removed from the list.

The other constraint from university concerns the relation to the Computer Science programme. This is already taken care of, since all remaining focus areas are related to the study programme.

The last constraint, from the company, is not yet applicable. It concerns the size of the impact of the recommendation and thus only comes in action at the evaluation.

The focus area Documentation was targeted on the collaboration between Product Management and the Design and Development departments. At this point, the student of the other project in the duo-assignment had finished his interview series. He had consulted colleagues from the Design and Development departments, and concluded that they did not share the issues in the Documentation group. Therefore, this group has also been removed from the list.

For choosing the final scope, Product Management was consulted to discuss the remaining list of focus areas for their importance to the company and relevance to the efficiency of the development process. Basically, they were asked to choose the issues which they thought would improve the process best.

The focus area Planning was the first to be excluded. PM focuses continuously on the issues in this group, so there was little reason to let this focus area be the main subject of this graduation project.

Among the issues in the Requirements Management group, the biggest win could be found with introducing a suggestion management system in the regions. This, however, would have had little to do with PM. Because the involvement of PM in the project assigned was obliged by the constraint from university, this was actually not a viable option.

Thus, the only remaining focus area was the Communication group.

3.4 Final Scope

With the focus area selected, it was time to create a final graduation project assignment. Table 3.4 repeats the issues in the Communication group.

The issues in this focus area are separated in items concerning Product Management itself and issues regarding the communication with regions and customers. The internal issues were already acknowledged by PM and initiatives were present to try to solve them. Therefore, the graduation project would focus on the external issues.

Description
Communication - PM Internal
Too little knowledge sharing
Too little group feeling
Too little creative collaboration
Little structure around communication
Communication - PM External
Too little communication (on functionality) with customers/regions
Requirements are mostly only checked with originating region
Released functionality may differ from requirements
Feedback to regions misses on decision making and development (planning)

Table 3.4: Communication issues from the interview report

Looking at the remaining issues, PM proposed to focus on a verification of requirements with customers and regions. This would also fit in perfectly with the values of Exact in general and the Exact 2009 target specifically; Customer Experience. Verification with the customer is also backed up by literature on agility. Cao and Ramesh identify the intensive communication between the developers and customers as the most important agile Requirements Engineering practice [11]. According to Lee and Guadagno, *"communication and interaction lies at the heart of agile practices"* [28]. Finally, Nisar and Hameed share that frequent communication with offshore clients is a recommended agile principle in offshore software development [34].

By having this interaction before the implementation phase begins, the concept and design of new functionality can be adapted relatively easy. The various thoughts on the product-to-be can thus be aligned without having to fix any misalignments in yet another implementation cycle. This makes the goal of involving the customer in the development process twofold:

1. To increase the customer's perception of being involved
2. To improve the product quality and development speed

Goal 1 is customer-oriented. In any business, customer experience is one of the most important focus areas. Meyer and Schwager define customer experience as such [32]:

> *"Customer experience is the internal and subjective response customers have to any direct or indirect contact with a company."*

To involve the customer in the development process – basically, to let the customer co-create the product – would bind the customer with the company and its products, and would thus lift the direct contact to a higher level. Also, the ability to influence the product-in-development could positively affect the customer's self image, which would also lead to an improvement in customer experience [18].

Goal 2 is company-oriented. The valuable feedback from the customer on the requirements of the product-in-development reduces the risk of delivering a product that fails to

meet the customers' requirements [39, 42] and would thus increase product quality. This result would also reduce rework and would therefore positively influence the development speed and serve as a cost reduction.

3.5 Final Objective

The final objective of the graduation project is:

> *To identify how Exact can create a feedback loop between PM and the 'customer' to verify whether the proposed requirements descriptions meet the customer's requirements.*

A customer in this sense could be any stakeholder that wants to participate in verifying and thus enriching the requirements, ranging from end-users to colleagues in the regional offices. For convenience, this stakeholder will just be called 'customer' throughout the remaining chapters of the thesis.

From the objective the following main research question can be extracted:

> *How to involve customers in a distributed agile requirements engineering process to create a feedback loop on requirements items?*

Sub-questions are:

- Which types of requirements items should be shared with the customer?
- At which stages in the process and how often should the customer be consulted?

To answer these questions, research was conducted in literature. In the next chapters, the results from this research will be shared.

Chapter 4

Theory

Before creating a prototype, research is done in literature to see what industry and science share about customer involvement in a distributed environment. The findings and conclusion from that research will be described in this chapter.

4.1 Importance of Customer Involvement

The initial result from the research explains why customer involvement is important in the first place. The corporate product lines of Exact can be categorized as market-driven software. This category is also called packaged software or off-the-shelf software, and is typical for software development in which there is no leading customer. In such a situation, a company deals with *"imagined customers"* and requirements are more or less *"invented"*, rather than elicited [29, 36]. At Exact's Product Management, the customers are also often represented by the colleagues from the regions. Keil and Carmel advice to rely on such indirect links with the customer as little as possible [26].

It is in this setting, that Exact wants to make its development process more agile. One of the agile aspects, as extracted by Dullemond et al. [16] from literature on agile software development, is *customer involvement*. They mention that, by incorporation of this aspect, one can get valuable feedback on the requirements and progress of a system in development. This thought is shared by various other sources. Nisar and Hameed [34] were already mentioned, but Paetsch et al. [35] also recommend frequent communication with the client. The Scrum approach also includes customers' *"controlled involvement at set intervals"*, because frequent interaction decreases the risk of delivering a product that fails to meet the customers' requirements [42]. In fact, Prahalad and Ramaswamy even call consumers *"a new source of competence"* [37]. Holmström concludes that involving customers in packaged software development will improve both products and processes [24]. Involving the customer with a *feedback loop* is recommended by Bar and Riis, saying that such interaction is one of the important elements in the relationship with the customer [6].

4.2 Approaches for Customer Involvement

From their study, Gruner and Homburg conclude that success can be increased by involving the customer during the early and late stages in the development process, particularly excluding the implementation phase [21]. Kaulio has built a framework to analyze different approaches for customer involvement in product development [25]. He identifies three types of customer involvement in this framework; *design for*, *design with*, and *design by* the customer. The other dimension in the framework is the phase of the design process. Kaulio identifies the following phases; Specification, Concept Development, Detailed Design, Prototyping, and Final Product. Given these two dimensions, Kaulio has grouped seven approaches, as shown in Figure 4.1.

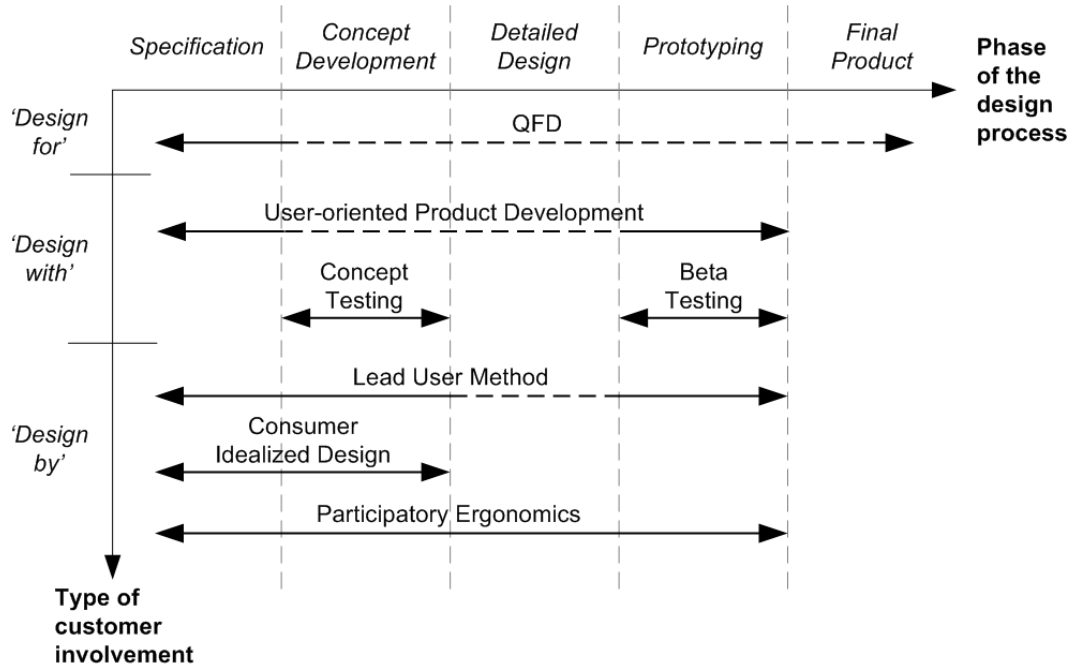

Figure 4.1: Approach analysis framework by Kaulio [25]

Mapping the framework onto the situation at Exact, the following conclusions can be drawn.

First of all, the 'design with' type of customer involvement matches best with Exact's view on involving the customer in the current development process. This notion leaves the approaches 'User-oriented Product Development', 'Concept Testing', and 'Beta Testing' to focus on.

Secondly, the 'Beta Testing' approach is already represented in the RDP by the combination of the *Prototyping* and *Controlled Release* phases. This excludes 'Beta Testing' from further focus.

According to Kaulio's descriptions, 'User-oriented Product Development' is focused mostly on the Specification and Prototyping phases. Because requirements specification

at Exact is mainly done prior to Product Management activity, and prototyping is already covered by 'Beta Testing', this approach is also excluded from further focus.

Therefore, only 'Concept Testing' remains. This approach involves the customer during the design phase of the concept of new functionality. It is beneficial to the organization to get feedback from the customer before starting to implement the requirements, because applying changes to the system during coding is more expensive than during specification and design [19]. For valuable feedback, it is recommended to share requirements items which give a *"realistic description of the proposed product"* [25]. 'Concept Testing' matches perfectly with the existing 'Beta Testing', because Kaulio recommends supplementing 'Concept Testing' with *"later prototype evaluations"*. The combination of the two approaches provides customer feedback both before and after the coding phase, and suits the situation at Exact, as can be learned from Kaulio's words:

> *"Concept and beta testing fit very well in an overall phase-divided product development process model. Both approaches have the characteristics of a method, and introducing them offers an easy way to extend an existing design process."*

Other sources support the idea of involving the customer with 'Concept Testing'; Brockhoff [9] and Gruner and Homburg [21] also recommend involving the customer in the concept development phase. Andriole recommends customer involvement to evaluate designs, requirements and prototypes [4]; items that also reflect the concept. Finally, Eberlein and Leite recommend *"intense customer interaction"* for requirements elicitation [17].

4.3 Conceptual design

The 'Concept Testing' approach is meant to facilitate the verification of requirements with the customer. More specifically, it concerns the *concept* of the product-to-be. In general, however, not every requirement item is suitable to get valuable feedback from a customer. Often, written specifications take too much time for a customer to evaluate thoroughly, while graphical items are easier to understand [19].

Kaulio shares the following recommendation [25]:

> *"Ideally, the presentation of a concept should offer a realistic description of the proposed product(s), in order to facilitate specific responses from customers."*

The responses mentioned by Kaulio correspond to the output of the feedback loop that is to be created. The input consists of the requirements items that will be shared. These requirements items should thus enable the customer to grasp the essence of the concept [39]. The following list gives some examples of items that are able to provide such information [14, 25, 39]:

- Models
- Scenarios

- Storyboards
- Use cases
- Wireframes, sketches
- Diagrams: information flow, data flow, activity
- Prototypes of the product-to-be
- Concrete questions with a concise context

4.4 Requirements for Customer Involvement

According to Gentile et al., customer involvement is affected by the cost of the offering and the impact on the customer's self image [18]. The cost part can be explicit as well as implicit. Explicit costs are items like entrance fees, subscription costs, etc. An implicit cost is for example the time it takes to 'be involved', e.g. to use the system offered by the company. In general, the higher the cost, the lower the participation. This could be handled by having a system with an easy-to-use interface [43], and little to no costs for the end-user. Berkley and Gupta confirm this, by sharing that giving feedback should be made easy in order to receive effective feedback from a customer [8]. The cost could also be viewed from the other side; Lüthje mentions that a financial reward would also motivate a customer to participate [31]. Brockhoff shares that the reward does not directly need to be financial; a better endproduct could already be motivating enough [9]. The customer's self image is somewhat harder to deal with from a company point of view. Lüthje gives a practical example of working with it [31]. He describes, that when a customer improves his self image by problem solving, he would probably be motivated enough already by himself to participate in a customer involvement setting.

Nambisan expresses the need for organizations to carefully examine the customers' roles in the process [33]. This is exactly what is done with the selection of the 'Concept Testing' approach in the previous section. Nambisan also advices organizations to create virtual environments to enable distributed customers to play their roles. In such environments, he says, two types of knowledge creation activities should be supported: knowledge acquisition and knowledge conversion. The acquisition part should handle the distribution of knowledge between the organization and the customers and among the customers themselves. The conversion part should handle the transformation of one type of knowledge to the other; e.g. from tacit to explicit knowledge. In the Exact case, the knowledge resides with individuals, which would lead to a network model being used for the acquisition of knowledge [33]. However, Nambisan also warns for too much transparency in a virtual environment, especially regarding competitive advantage. Therefore, the level of transparency and security, as well as the type of customers to involve, should be carefully defined.

According to literature, attractive types of customers to involve are *lead users*, *financially attractive customers* and *close customers* [6, 21, 25]. However, some caution is recommended. Bar and Riis indicate that only involving lead users may result in a product

which is too sophisticated for the normal user [6]. Christensen and Bower also warn that too close involvement of the customer can lead to a lock-in for product innovation [12]. Therefore, Luteberget advices to involve various kinds of users to keep the product broadly applicable [30].

Virtual environments in which the participants focus on the same topics or have similar interests are called virtual communities. In such virtual communities, as Koh et al. point out, *"posting and viewing of information are fundamental elements in the ongoing life"* [27]. Their study also shows that posting activity is influenced by offline interaction (real-life events etc.) and that viewing activity is affected by perceived usefulness. However, they also argue that offline interaction and the quality of the IT infrastructure might be effective substitutes for one another, thus removing the need for real-life events, etc.

The virtual community is also suggested by Holmström as a viable approach to apply in the packaged software development environment, where many other approaches for involving customers are difficult or insufficient to apply [24].

4.5 Conclusion from Theory

The following conclusions can be drawn to answer the research questions:

- The feedback loop with the customer should be created by means of a virtual community.
- The feedback loop(s) should be run during the conceptual design phase.
- Only requirements items that represent the concept of the product-to-be should be shared.

Following this conclusion, a prototype is proposed. This prototype will be described in the next chapter.

Chapter 5

Prototype

Based on the findings from theory, a prototype is created as a proposal to enhance the RDP with customer involvement. In this chapter, that prototype will be explored by describing the proposal, its prerequisites and the fit in the current development process.

5.1 The Requirements Verification Method

To increase customer involvement at Product Management, theory basically recommends to create a feedback loop in a virtual community during the conceptual design phase. Kaulio already mentions that the 'Concept Testing' approach has method characteristics and it should therefore be relatively easy to implement it in existing processes [25]. These notions can be combined into the *Requirements Verification* method.

A feedback loop is basically an iterating sequence of reacting on previous input. This is the basis of the Requirements Verification method. At some point, the feedback loop should be started. In this case, this would be done by giving specific requirements items as input to a virtual community. After that, the actual iterations of giving feedback start off. During these iterations, the input can also be clarified further if needed. Following the importance of perceived usefulness (section 4.4), the virtual community should be maintained and moderated. Input should be approved, unwanted content should be removed; all to keep the community useful and on-topic. Finally, at some other point, the feedback loop should be closed. To really be meaningful to both sides, there should also be some feedback from the initiators on how the input from the feedback loop is used in the development process.

From this description, the following roles can be derived for Requirements Verification:

Initiator This person initiates the feedback loop

Generator This person gives the feedback

Moderator This person maintains the virtual community

In the prototype for Exact, the initiator role would be performed by Product Management or Functional Design. The generator role would then be performed by customers or colleagues

from the regions. Finally, the moderator role would ideally be performed by the Product Owner.

The description can also be translated into an action flow graph. Figure 5.1 shows a graphical representation of the Requirements Verification method.

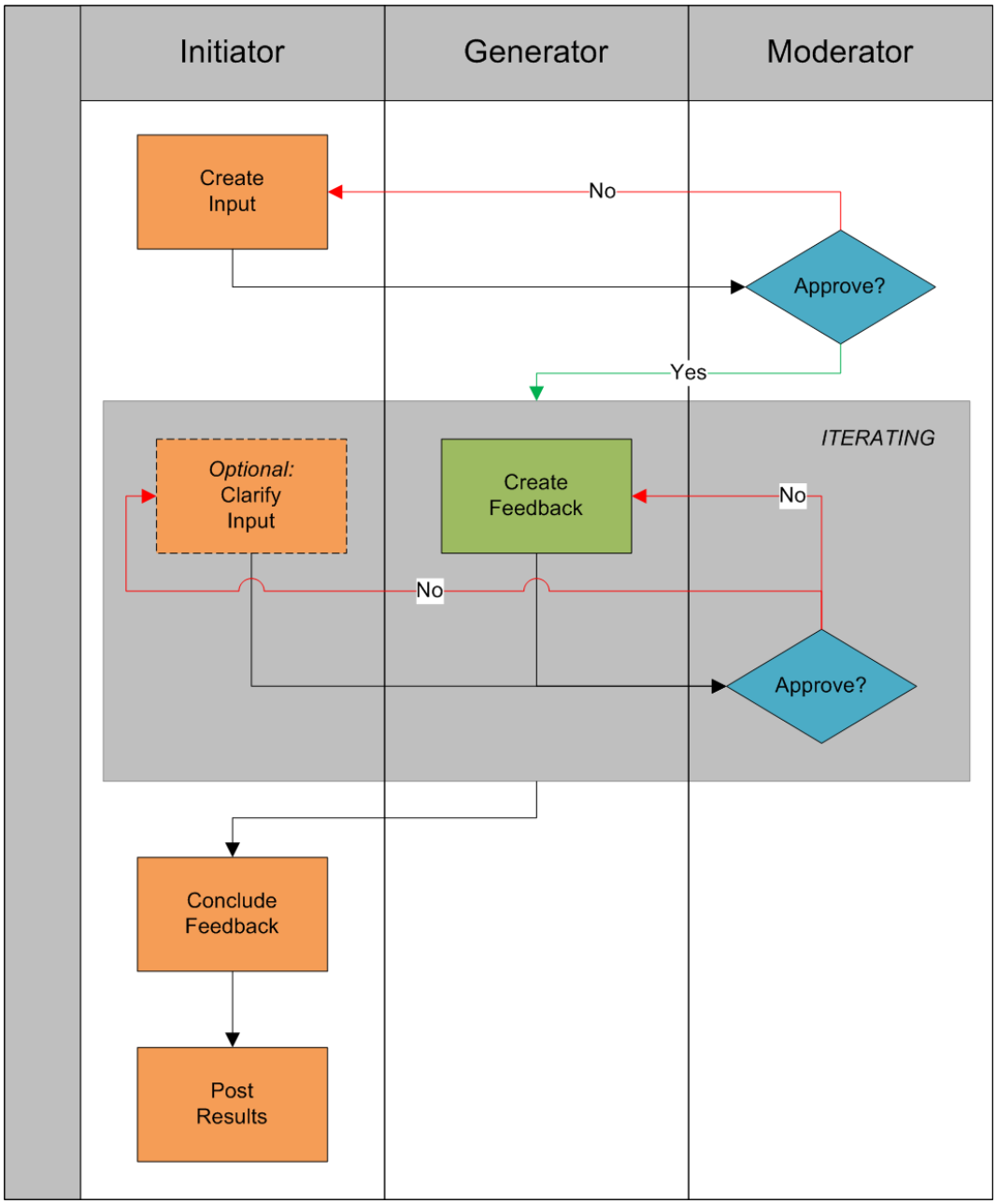

Figure 5.1: The Requirements Verification method

5.2 Prerequisites

The Requirements Verification method is meant to establish a feedback loop with the customer. Table 5.1 lists the prerequisites created for the prototype, which should be satisfied for a successful feedback loop in a virtual community.

#	Description
1	The virtual community should be globally available 24/7. If not, the actors may be limited in contributing.
2	The virtual community should be kept on-focus. If not, the perceived usefulness of the community may degrade, which may affect viewing activity by the actors.
3	There should be a group of generators, willing to collaborate in the development process. If not, a feedback loop will not be possible.
4	The group of generators should not be too large. If so, the community may not be able to manage it.
5	The generators should have the discipline to interact frequently enough in the community. If not, the incentive for the initiators to create feedback loops may degrade.
6	There should be a group of initiators, willing to involve the generators in the development process. If not, a feedback loop will not be possible.
7	The initiators should create requirements items to share. If not, there is no input for the generators to give feedback on.
8	The initiators should pose their questions in such a way that the generators are triggered to give valuable/concrete feedback. If not, the feedback will not be very useful for the verification of the requirements.
9	The initiators should give feedback on how the generator's feedback is used in development. If not, the generators may think they are not listened to and lose motivation to participate in the community.
10	The initiators should have the discipline to interact frequently enough in the community. If not, the incentive for the generators to collaborate may degrade.
11	There should be a (group of) moderator(s), willing to keep the community on-focus. If not, the community may lose its focus and usefulness.

Table 5.1: Prerequisites for success

5.3 Fit in Refined Development Process

Given its appearance as a method, it should be easy to plug Requirements Verification into the RDP. However, the RDP does not contain a 'conceptual design' phase. Looking more closely to the current phases, testing conceptual items would fit best during the iterating creation phases of the Product Requirements Document (PRD) and Functional Design (FD). Once conceptual items are created or concrete questions have come up, a feedback loop can be initialized.

Customer involvement in this prototype is limited to the work on the PRD and FD. For the Technical Design (TD), the customer would typically not be able to provide any useful feedback anymore, because of the need for technical (product) knowledge at that point. Therefore, the underlying conceptual design phase will be finished when the FD is completed. The action of deciding when the PRD or FD is completed remains the same as in the current process.

Figure 5.2 shows a graphical representation, with the conceptual design phase marked in blue.

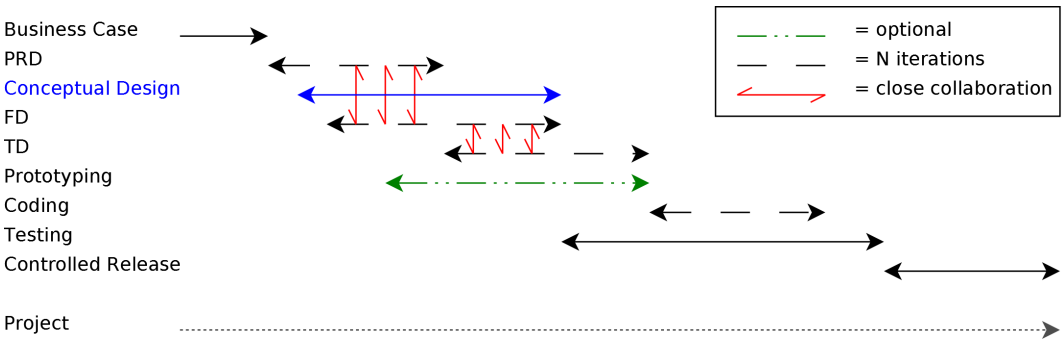

Figure 5.2: Conceptual Design in the Refined Development Process

One constraint by the company still remains. As can be seen from this section, implementing Requirements Verification changes the current process only little. Which effect the method will have on Exact's product development process will be the topic of the next chapter.

Chapter 6

Proof of Concept

To check whether the prototype really reaches its goals, a proof of concept (PoC) was created. This chapter contains a description of the implementation of the prototype and the tests that were conducted.

6.1 Implementation of a Virtual Community

The recommended 'virtual community' is still a broad term, including many possibilities for integration into the development process. To retrieve more details, we have to dive further into the background of the feedback loop that Exact wants to create. Firstly, Exact's customers are scattered all over the globe. Therefore, the customer involvement system has to overcome distances in time as well as space. This implies that the communication style would be asynchronous. Secondly, creating a feedback loop means that the communication would typically be two-way between customer and organization. Thirdly, the organization might not know in advance for which participating customers a feedback loop on certain requirement items might be usefull. Therefore, an indirect communication system (in which the recipients are not explicitly known in advance) would be preferable.

In their research on agile aspects, Dullemond and Van Gameren also listed technologies that support agile global software development [15]. Taking into account the requirements from theory and the characteristics of the Exact feedback loop, we end up with the following possible technologies: a forum, a wiki, and a mailing list. These technologies are also shared by Robertson and Robertson [39]. All of these technologies are internet-based. According to Sawhney et al., this allows organizations to reach a larger number of customers than with traditional communication media, without loosing significantly on the richness of the interaction [40]. Synonyms for a forum are bulletin board or message board.

Looking at the results from a case study by Keil and Carmel for customer-developer links in packaged software projects [26], similar conclusions can be drawn. Taking into account the same restrictions as above, the following links come up for successfull projects: user-interface prototyping, user group, and requirements prototyping. These links fit perfectly into the technologies mentioned already.

The proposed technologies also correspond with the conversational technologies as described by Wagner [47]. He states that conversational technologies are especially suitable for environments in which multiple, possibly distributed owners hold the knowledge, instead of a centralized environment. Wagner also elaborates on the difference between a wiki and a forum or email; with a wiki, content is organized by topic, whereas with the other options content is organized chronologically first.

Elaborating further on these technologies shows that the mailing list can already be excluded from the options. Firstly, collaboration via email of a globally distributed community with many participants would probably result in an information overflow [5]. Secondly, organizing all the emails would require additional manual handling, or at least some organizing logic in software on each participant's computer. This would be an extra hurdle for participation, which makes it preferable to have the organizing logic centralized. Therefore, only the wiki and the forum remain.

Both the wiki and the forum have a similar impact on the customer regarding posting and viewing content. However, a wiki is just a set of linked webpages [47]. Using this technology for a virtual community would be like collaboratively creating a document. A forum would thus be better suited for a feedback loop, because of its natural distinction between posts and its automatic chronological ordering.

Therefore, the proof of concept is implemented in a forum. Given the recommendations from theory (section 4.4), Table 6.1 lists the restrictions that apply to the forum.

#	Description
1	Given the recommended precaution regarding competitive advantage, the forum should be hosted on a company owned server for maximum controlability, and should be set up to an appropriate security level.
2	The forum should be world-wide available.
3	The forum should enable the Requirements Verification method to be implemented, which means that it should be possible to share conceptual requirements items on the forum.
4	Having an open-source or freeware forum would limit the cost for the company, thus automatically limiting the cost for the participating customer.
5	The forum should have an easy-to-use interface with acceptable response times to limit the implicit cost.
6	The forum would automatically enable knowledge conversion and acquisition, by its posting and viewing capabilities.
7	Ideally, the forum should have a moderating option. This should be used to keep the discussions on-topic, thus ensuring that the forum keeps a high level of usefulness.

Table 6.1: Restrictions for the forum

First step now was to find a server. Exact provided a local server running Windows, which led to the additional characteristic that the forum would be implemented in the programming language ASP.NET. Using the recommendations, restrictions and characteristics, a comparison was made for forum software. This resulted in the open-source forum soft-

ware YetAnotherForum.NET [49] being chosen and deployed on the server.

6.2 Technical Test

After deployment of the forum software, a technical test was performed. During five days, a team of volunteers from Product Management simulated a feedback loop to reveal any issues with the forum software and, to a minor extent, to evaluate a basic implementation of the Requirements Verification method. The evaluation of the technical test can be found in Table 6.2.

Issue	**Solution**
Remarks related to first-time use of forum software	
The need to log in was unclear	Explained to actors
The need to first select a forum was unclear	Explained to actors
The action flow to add an attachment was unclear	Explained to actors
The action flow to add an image was unclear	Explained to actors
Remarks related to the forum software	
PDF documents were not allowed as attachment	Added PDF to the list of allowed documents
Default view of 'Active Topics' was not ideally set up	Changed default view to 'Since last week'
Closing or locking own topics was not possible	Added moderator access mask to initiator roles
Changes in posts were not tracked	Not fixed; not really crucial for PoC
If a forum was watched by email, an email was received with every addition or change in that forum	Not fixed; could still be desirable for other actors
If a forum was watched by RSS, only new topics were mentioned; no replies to topics	Subject of RSS feeds can be chosen; forums, topics, or active topics
Remarks related to the infrastructure	
The server allowed a limited number of connections to the website	Increased maximum number of simultaneous connections to 40 (max)
Remarks related to the concept-testing method	
The need to wait for post approval did not suit the small group	Removed full-moderated status from forum and added recommendation to thesis
The number of new threads should be limited to keep a clear overview	Added recommendation to instruction and thesis

Table 6.2: Evaluation Technical Test

6.3 First Feedback Loop Test

The second test involved running a feedback loop between PM and the customers and/or regions. This loop would continue for three weeks during the development phase of PRD and FD. In this loop, PM would have the role of initiator, while the customers and/or regions would be generators. PM would create requirements items that reflect the concept of the new functionality, and would share them, along with their questions, in the virtual community. The generators in the virtual community would then give their feedback on the items and questions. In the end, the initiators would reply to the generators what was done with their feedback in the development process. After running the test, the initiators and generators would receive an evaluation form. The evaluation would indicate whether the Requirements Verification method reaches the goals mentioned in section 3.5.

As soon as a project was assigned to the test, a kick-off meeting was held with the Product Manager handling the test-project. Four weeks after that, the evaluation started by giving the participants access to the evaluation forms. The questions from the forms can be found in Appendix F. In this test project, the originator was a Product Manager and the generators were colleagues from the regions.

Despite the fact that the test was run for a month, there was only little interaction on the forum. Actually, the feedback loop was not initiated before the third week. This was mainly due to the busy schedule of the initiator.

Besides the late initiation, the proposal also did not result in any replies. The attachment, however, did get downloaded by all generators. The proposal proved to be discussed already in a face-to-face meeting, leaving no subjects for discussion on the forum. Completely new and more concrete input would be better to start a feedback loop. However, the creation of the recommended requirements items was not yet standard procedure. The Usability Engineer within PM had only recently started a project on including interaction design in the development process. At the moment of the test, this design phase was not really incorporated in the process yet, giving the initiator little standard input to share on the forum.

The participants in the test were also not really distributed. Being one floor apart, frequent face-to-face meetings were arranged. In such a setting, the forum proved to be less useful in enabling communication.

Due to the absence of a real feedback loop, no forms were filled in and returned. The participants did, however, share some additional thoughts besides the points above. First of all, the look and feel of the forum was thought to be complex. Secondly, it was suggested to join Exact's existing forums. Also, a forum appears to be a good way to interact in a community. Finally, the feedback loop would be suitable to serve as follow-up or even replacement of customer feedback days. On such days, a group of customers meet with Product Management face-to-face to discuss the product.

Given the results of the second test and the small number of participants (1 initiator, 3 generators), a third test would be organized to be more certain at concluding the evaluations.

6.4 Second Feedback Loop Test

As soon as another project was found, a kick-off meeting was held to start the test. However, the new feedback loop proved not to be set up within a week. This was again due to the busy schedule of the initiator. Because the available timeline for the proof of concept phase had reached its end, there was no time left to wait any longer or set up another test.

6.5 Additional Feedback on Prototype

During the proof of concept phase, one of the product managers conducted a test of his own by creating a feedback loop on the company's product blog. In this loop, feedback was asked on some changes in the user interface.

It proved to be difficult to perform a textual feedback loop with a large group right. With reading and writing, the message can easily be interpreted wrong, just like with email conversations. This may be caused by a lack of time to read the input properly, but the context will stay different for every reader anyway.

This experience also revealed that a blog – and in that sense a forum too – is less suitable for a broad discussion. Both technologies are static and thus lack the ability to provide the interactivity needed for such discussions. However, one might wonder whether this kind of communication should be handled in an asynchronous setting.

The blog and forum would be better suited for concrete questions. For example, instead of asking "What do you think of this new overview screen?", one could better pose a question like "We are about to implement this new overview screen. Is any information missing from the prototype?" [39]. Another option would be to present several options to implement and let people vote on them.

The product blog is freely available, so one could wonder about the risk of exposing business intelligence. In this case, however, that risk was not really present. The screen was already created, so the feedback loop was only a verification afterwards. Earlier input had already been gathered via other techniques.

One last remark was that an additional post with feedback from the company on the usage of the replies from the feedback loop was beneficial for the mutual understanding. This would therefore be recommended for future feedback loops.

Chapter 7

Discussion

In this chapter the validity, recommendations, conclusion, and evaluation of the graduation project and a vision for the future will be discussed.

7.1 Validity

The check for validity is separated into the four validity types mentioned by Wohlin et al: conclusion, internal, construct and external [48].

The conclusion validity looks at the ability to draw correct conclusions. Since the implementation of the proof of concept reflected the prototype description, the reliability of the implementation was good. However, due to the limited availability of projects, the sample size of the proof of concept was small. Also, because the initiators did not succeed in setting up a feedback loop, the evaluation forms could not be used to provide feedback. Therefore, the evaluation could not show whether the prototype reaches its goals. Actually, the only items that could be evaluated focus more on technical issues regarding the usage of the prototype, rather than functional issues or aspects regarding requirements development. More tests should be done, but it is recommended to wait until the creation of conceptual requirements items will be part of the standard development process.

The internal validity concerns the subjects and events both before and during experiments. The main input source for understanding the current process was a series of interviews. Conducting interviews in research projects introduces some constraints and risks for the validity of the results. First of all, one is dealing with the perceptions of individuals, which could for example be influenced by emotions or personal characteristics. Furthermore, interview results may be influenced by the ideas, values or biases of the interviewer [10]. Formulation of the answers by the interviewee and interpretation of them by the interviewer might also be a challenge, leading to the wrong message being shared. Finally, in this special case, no recording device was used to aid the memories of the interviewer [10], which could also lead to a report that does not reflect the interview. Then again, the interview series were actually *meant* to retrieve the individual's perception. Besides that, the interviewees work daily with the subject of study. They thus talk from experience rather than memory, which improves the accuracy of the reported perceptions. And, to overcome

any risk of reporting the wrong message, all interview reports were verified with the interviewees. Finally, the interviews took place in a separate meeting room, to avoid distraction or discussion with colleagues and to enable the interviewee to talk freely. These measures bring about that the results from the interviews are valuable enough to serve their purpose. The proof of concept, however, experienced some negative influences from the participants having face-to-face meetings next to the interaction on the forum and posting already discussed items as input for a feedback loop.

The construct validity discusses whether the findings from the proof of concept are really an indication for the subject that is to be evaluated. Because the evaluation would have been dealing with perception only, the evaluation would be regarded valid.

The external validity concerns the generalizability of the conclusion. The absence of test projects that involve real customers already makes the thesis result not external valid. The timing of the proof of concept was also narrow towards the end, which may also have affected the results. The environment of the test, however, did reflect the target environment.

7.2 Recommendations for the Prototype

Given the validation and the evaluations of the tests, the amount of recommendations that can be shared is only limited. Table 7.1 lists the recommendations for the prototype.

#	Description
1	The virtual community should only be used in a truly distributed setting, in which participants do not easily meet face-to-face, or for example in projects with large email discussions.
2	Ideally, the customer should only have to visit a limited number of different locations for his interaction with the company. Implementing new feedback loops and virtual communities should be done on existing mediums whenever possible.
3	A static environment like a forum or blog should be used for concrete questions only. For broader discussions, a more interactive environment would be better suited, but the combination of interactivity and asynchronous communication sounds quite challenging.
4	An initiator should only share requirements items that were not shared in previous meetings, to maximize the usefulness of the resulting feedback.
5	As soon as a virtual community is (semi-)open to the public, the community should be under full moderation to protect its usefulness. To the more closed communities this extra hurdle could be an impediment for participation.
6	The moderator should keep the overview of concurrent feedback loops clear. Limiting the number of concurrent feedback loops could for example prevent generators from losing focus.

Table 7.1: Recommendations

7.3 Conclusion

The graduation project was set up to improve Exact's requirements engineering process as part of a larger project to increase agility in the product development process. A series of interviews revealed bottlenecks and points of attention of the current development process. Together with a mapping of the current process on agile practices and constraints from both university and Exact, the interviews were used to scope down the initial thesis assignment to a more appropriate size for the graduation project.

The objective of the final thesis assignment was to identify how Exact could create a feedback loop between PM and the 'customer' to verify whether the proposed requirements descriptions meet the customer's requirements. The goal for customer involvement in the process was (a) to increase the customer's perception of being involved, and (b) to improve the product quality and development speed.

Research in literature showed that it was recommended to create a virtual community to facilitate a feedback loop with the customer. This feedback loop should be run during the conceptual design phase with requirements items that represent the concept of the product-to-be.

Based on the recommendations from theory, a prototype was created, which was called the Requirements Verification method. This method contained three roles: Initiator, Generator, and Moderator. An initiator starts the feedback loop, after which the generators provide their feedback. Moderators are meant to maintain the feedback loop; keep it on-topic and useful. The prototype also lists a set of prerequisites for feedback loop success. The method should be implemented in a conceptual design phase, but such a phase was not yet present. This phase would fit in best during the iterating creation phases of the Product Requirements Document (PRD) and Functional Design (FD) and would as such change the current process only little.

To check whether the prototype reaches its goals, a proof of concept was created. The virtual community was implemented with a forum as medium. The prototype was evaluated in multiple tests. Whether the prototype meets its purpose could not be concluded, but the evaluation did provide enough input to lead to the recommendations in the previous section.

Even though the prototype could not be validated within the time frame, the graduation project does contribute a number of things to Exact. First of all, it combines findings from literature concerning customer involvement in product development and extracts from that combination a practical recommendation for the specific situation at Exact. Secondly, it revealed a set of points of attention for the current product development process, on which Exact could focus to improve the process even further. Last, but not least, collaboration on the graduation project reinforced the focus on the customer within the company.

7.4 Evaluation

The evaluation of the graduation project is separated into the thesis assignment and the working method that was used.

7.4.1 Thesis Assignment

The start of the graduation project actually was a road of discovery towards the final assignment definition. Eventually, that checkpoint was only reached about halfway the duration of the project. If that had been fixed sooner, I would probably have done more thorough testing iterations in the proof of concept, involved real customers in these tests and maybe participated during a complete project run to really check the added value of the Requirements Verification method. At this point, for example, it may take some more tests to find out which medium would be ideal in which situation.

The testing that *was* performed failed to deliver appropriate results to be able to completely validate the prototype. The main reasons were the busy schedules of the initiators. They just could not find the time to create the necessary requirements items next to their normal businesses. One could also wonder whether the problem was at the planning side of the graduation project. At the start of the proof of concept phase, it was decided to concentrate on one large test, rather than several small ones, so the test could be better integrated with a project. The proof of concept phase eventually lasted for six weeks from the start of the search for test projects to the start of the thesis finalization phase. This is roughly 16% of the total project duration, which is a reasonable share for testing within a project. Maybe the proof of concept would have succeeded better when multiple tests would have been run simultaneously, but finding a project proved to be difficult enough already even for a single project.

During the project, the constraints from the company and university were taken into account. An evaluation of the end results leads to the following notions. The main recommendation is to use the Requirements Verification method, which implements an agile aspect – this satisfies the first constraint. The method can also easily be plugged into the current process, so the recommendation has little impact on that process. However, because none of the tests in real settings could actually be evaluated, it is not yet clear which effect the method has on the process and product – the second constraint is satisfied only partly. Because the project concerns improving the requirements engineering process in a software development process, the assignment was indeed related to the Computer Science programme – this satisfies the third constraint. The fourth constraint – keeping both projects from the duo-assignment as close together as possible – deserves some special attention. The scoping iterations were meant to search for the most worthwile focus areas from PM point of view. Ultimately, this meant a split between the projects, leaving PM as a vague link in between. In the end, however, the link between the two projects appears to be stronger than anticipated. The requirements engineering project tried to introduce iterations in communication between PM and the customers and regions, while the other project supported the existing iterations in communication between PM and the departments Development and Design by automation of the development environment. As is depicted in Figure 7.1, the mutual target of the projects could thus be described as “to support agility in the RDP by means of frequent, iterating interaction”. This notion closes the circle and brings the two parts of the duo-assignment back together again.

The distribution in the requirements engineering process also reflected on the graduation project. The separation between the two projects of the duo-assignment was not the only

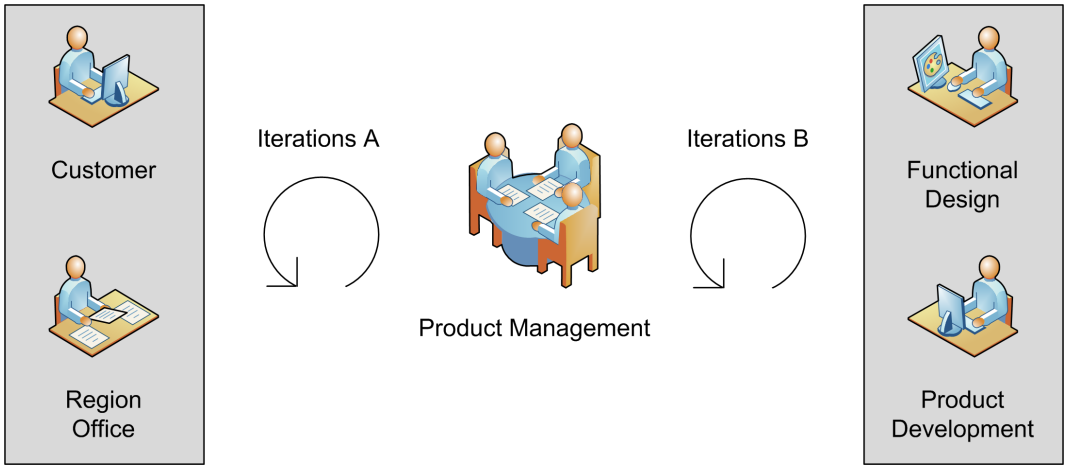

Figure 7.1: Overview of the new process

thing that increased the level of distributed collaboration. Some of the interviews also took place between Delft and Kuala Lumpur (KL), and to plan meetings with colleagues that visit KL frequently at least creates global awareness. Unfortunately, the proof of concept phase turned out not to be as distributed as hoped for, due to the inavailability of projects that involved distant actors.

7.4.2 Working Method

Despite the busy schedules and the spreading of holidays and visits to KL, we succeeded in establishing enough feedback moments for the supervisors. On average, the TU was met once a month and meetings with the Exact supervisor were scheduled every two weeks, next to the spontaneous meetings or conversations on the floor.

During six months, the other project of the duo-assignment resided in KL. The interaction between the two projects was provided by scheduled phone calls twice a week. In these calls, we discussed the things done and the things to do, and concluded with casual small talk to keep up the valuable knowledge of each other's occupations.

The meetings with other colleagues from PM, however, were less frequent. Participation in running projects was also not arranged. This proved to limit the knowledge of the current practice, despite documentation and interviews. Real involvement in the meetings and work of PM would probably have resulted in a better picture of PM at an earlier stage.

To gain experience in agile development, the graduation project itself was supposed to reflect the Scrum approach in its working method. Initially, the sprints had a length of three weeks. The first phase with the interviews proved already not to be suitable for scheduling in sprints, due to the absence of a real deliverable and the time it took to arrange and complete the interviews. The second phase, which scoped towards the final assignment definition, was also not ideal for the Scrum approach, because the scoping iterations had varying lengths. On the other hand, these phases could also be seen as the preparation phase for the product backlog, which usually preceeds the sprint planning. After defining the final assignment,

the deliverables proved to be too abstract for creating a meaningful product backlog, which hampered the sprint planning for the remaining weeks. Still, however, the rhythm of sprint-like feedback iterations with the supervising team was good to keep everybody informed and on target. Eventually, though, the final schedule reflected the waterfall approach more than the Scrum approach. Appendix B shows an overview.

A final element to evaluate from the working method is the writing of the thesis. The thesis was written along during the entire project, which made the end of the project less stressful. Due to the fact that all previous actions and results were already written down, not much new text had to be written. Finalization of the document was nothing more than just reviewing the contents, rather than recalling the actions from months before and writing them down for the first time.

7.5 Vision for the future

Exact just introduced its new corporate identity. The identity has a strong focus on the customer with the cornerstones *People*, *Collaboration*, *Structure* and *Results* [20]. These cornerstones are used to support the customer. The idea behind the prototype from the graduation project maps perfectly on the cornerstones, which strengthens the recommendations made and serves as an extra trigger to really proceed towards the incorporation of the new method in the current development process.

The thought of enriching the development process with customer interaction was actually already present in Exact. During the entire project I have experienced much enthousiasm from various sides when talking about the project. Colleagues were interested in my findings and glad to help out when possible. Some even picked up solutions from theory before I could share them officially. The opportunities with involving the customer to improve both product and process thus appear to be shared widely.

The graduation project is also linked to other initiatives within Product Management on customer experience. The productblog and its potential for feedback loops were already mentioned in section 6.5. Next to that initiative, the Usability Engineer is currently trying to increase the focus on interaction design during development. This would be a great preparation for the Requirement Verification method, because the models, diagrams, etc. created by interaction design are exactly the kind of items that are recommended to share with the customer. As a final example, PM has recently started to organize frequent customer feedback days. To relieve the organization around these meetings, a feedback loop in a virtual community could serve as a follow-up or even replacement.

In practice, however, Exact is not really ready for structural incorporation of feedback loops with the customer yet. The results from the proof of concept show that it is difficult to initiate a feedback loop without the recommended requirements items as standard deliverables in the process. It would therefore be recommendable to first include interaction design in the process and get used to its deliverables, before including requirements verification with the customer. After that, I would recommend to integrate the method in some projects in which Product Management can have really distributed interaction with real cus-

tomers. When initiated and run with care and devotion, these tests should show whether the prototype really works for Exact and delivers its promises.

The idea behind the prototype itself could also be used more broadly than just for requirements verification in the conceptual design phase. Actually, a feedback loop in a virtual community could in principle be used for any interaction with distributed actors, regardless of subject or phase. For example, an extensive email conversation with multiple persons could be replaced by a discussion in a forum topic. This would improve both readability and traceability without any additional actions.

Other future work could consist of more psychology-related issues; how to tackle the interaction in a virtual community, which questions to ask with certain items to get useful answers, etc. These questions also touch the area of human-machine interface (HMI); which level of hands-on experience results in the most convenient feedback, etc. The aforementioned issues also relate to the questions how to bridge the time gap between having the concept of requirements and having a workable prototype, and how to give the customers hands-on experience as early as possible in the development process.

7.6 Final Closure

Given the corporate identity and focus on customer experience, Exact in theory looks to be ready to involve the customer more and more in its development process. There certainly are enough triggers to take that extra step and create more Exact communities. With this thesis, a suggestion is provided in the specific area of requirements engineering, but the underlying idea could easily be introduced throughout the company. Extending and structurizing customer interaction in more phases of the development process as well as in other parts of the company would establish the pleasure of working with the customer for many years to come.

Bibliography

[1] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta. Agile software development methods: Review and analysis. *Espoo 2002*, page 107, 2002.

[2] A. Abran, J.W. Moore, P. Bourque, R. Dupuis, and L.L. Tripp. Swebok: Guide to the software engineering body of knowledge. *IEEE*, 2004.

[3] T.J. Allen. *Managing the Flow of Technology*. MIT Press, Cambridge, MA, 1977.

[4] S.J. Andriole. Fast, cheap requirements: Prototype, or else! *IEEE Software*, 11(2):85–87, 1994.

[5] O. Bälter. *Electronic mail in a working context*. PhD thesis, KTH Royal Institute of Technology, Department of Numerical Analysis and Computer Science, 1998.

[6] F. Bar and A.M. Riis. Tapping user-driven innovation: A new rationale for universal service. *The Information Society*, 16(2):99–108, 2000.

[7] S. Berczuk. Back to basics: The role of agile principles in success with an distributed scrum team. In *AGILE '07: Proceedings of the AGILE 2007*, pages 382–388, Washington, DC, USA, 2007. IEEE Computer Society.

[8] B.J. Berkley and A. Gupta. Identifying the information requirements to deliver quality service. *International Journal of Service Industry Management*, 6(5):16–35, 1995.

[9] K. Brockhoff. Customers' perspectives of involvement in new product development. *International Journal of Technology Management*, 26(5-6):464–481, 2003.

[10] A. Bryman. *Social Research Methods*. Oxford University Press, March 2008.

[11] L. Cao and B. Ramesh. Agile requirements engineering practices: An empirical study. *IEEE Software*, 25(1):60–67, 2008.

[12] C.M. Christensen and J.L. Bower. Customer power, strategic investment, and the failure of leading firms. *Strategic Management Journal*, 17(3):197–218, 1996.

[13] M. Cohn and D. Ford. Introducing an agile process to an organization. *IEEE Computer*, 36(6):74–78, 2003.

[14] A. Cooper, R. Reimann, and D. Cronin. *About Face 3: the Essentials of Interaction Design*. John Wiley & Sons, Inc., New York, NY, USA, 2007.

[15] K. Dullemond and B. van Gameren. Technological support for distributed agile development. Master's thesis, Delft University of Technology, Delft, the Netherlands, June 2009.

[16] K. Dullemond, B. van Gameren, and R. van Solingen. How technological support can enable advantages of agile software development in a gse setting. In *ICGSE '09: Proceedings of the 4th IEEE International Conference on Global Software Engineering*, pages 143–152, 2009.

[17] A. Eberlein and J.C.S. do Prado Leite. Agile requirements definition: A view from requirements engineering. In *TCRE '02: Proceedings of the International Workshop on Time-Constrained Requirements Engineering*, 2002.

[18] C. Gentile, N. Spiller, and G. Noci. How to sustain the customer experience: An overview of experience components that co-create value with the customer. *European Management Journal*, 25(5):395–410, 2007.

[19] H. Gomaa and D.B.H. Scott. Prototyping as a tool in the specification of user requirements. In *ICSE '81: Proceedings of the 5th International Conference on Software Engineering*, pages 333–342, Piscataway, NJ, USA, 1981. IEEE Press.

[20] Exact Group. *Exact company brochure: Being there for your business*. 2009.

[21] K.E. Gruner and C. Homburg. Does customer interaction enhance new product success? *Journal of Business Research*, 49(1):1–14, 2000.

[22] V. Ho. Exact's refined development process. Exact intranet, document 18.912.209, April 2009.

[23] S. Hole and N.B. Moe. A case study of coordination in distributed agile software development. *EuroSPI '08: Communications in Computer and Information Systems, Proceedings of the 15th European Conference on Software Process Improvement*, 16:189–200, 2008.

[24] H. Holmström. *Community-based customer involvement for improving packaged software development*. PhD thesis, Göteborg University, Department of Informatics, November 2004.

[25] M.A. Kaulio. Customer, consumer and user involvement in product development: A framework and a review of selected methods. *Total Quality Management*, 9(1):141–149, 1998.

[26] M. Keil and E. Carmel. Customer-developer links in software development. *Communications of the ACM*, 38(5):33–44, 1995.

[27] J. Koh, Y.-G. Kim, B. Butler, and G.-W. Bock. Encouraging participation in virtual communities. *Communication of the ACM*, 50(2):68–73, 2007.

[28] C. Lee and L. Guadagno. Fluid:echo agile requirements authoring and traceability. In *Proceedings of the Midwest Software Engineering Conference 2003*, pages 50–61, 2003.

[29] M. Lubars, C. Potts, and C. Richter. A review of the state of the practice in requirements modeling. In *RE '93: Proceedings of IEEE International Symposium on Requirements Engineering*, pages 2–14, 1993.

[30] A. Luteberget. Customer involvement in new service development: How does customer involvement enhance new service success? Master's thesis, Agder University College, Grimstad, Norway, May 2005.

[31] C. Lüthje. Characteristics of innovating users in a consumer goods field: An empirical study of sport-related product consumers. *Technovation*, 24(9):683–695, 2004.

[32] C. Meyer and A. Schwager. Understanding customer experience. *Harvard Business Review*, 85(2):116–126, 2007.

[33] S. Nambisan. Designing virtual customer environments for new product development: Toward a theory. *Academy of Management Review*, 27(3):392–413, 2002.

[34] M.F. Nisar and T. Hameed. Agile methods handling offshore software development issues. In *INMIC '04: Proceedings of 8th International Multitopic Conference*, pages 417–422, 2004.

[35] F. Paetsch, A. Eberlein, and F. Maurer. Requirements engineering and agile software development. In *WETICE '03: Proceedings of the Twelfth International Workshop on Enabling Technologies*, page 308, Washington, DC, USA, 2003. IEEE Computer Society.

[36] C. Potts. Invented requirements and imagined customers: Requirements engineering for off-the-shelf software. In *RE '95: Proceedings of the Second IEEE International Symposium on Requirements Engineering*, pages 128–130, Washington, DC, USA, 1995. IEEE Computer Society.

[37] C.K. Prahalad and V. Ramaswamy. Co-opting customer competence. *Harvard Business Review*, 78(1):79–88, 2000.

[38] B. Ramesh, L. Cao, K. Mohan, and P. Xu. Can distributed software development be agile? *Communications of the ACM*, 49(10):41–46, 2006.

[39] S. Robertson and J. Robertson. *Mastering the Requirements Process (2nd Edition)*. Addison-Wesley Professional, 2006.

[40] M. Sawhney, G. Verona, and E. Prandelli. Collaborating to create: The internet as a platform for customer engagement in product innovation. *Journal of Interactive Marketing*, 19(4):4–17, 2005.

[41] B. Schatz and I. Abdelshafi. Primavera gets agile: A successful transition to agile development. *IEEE Software*, 22(3):36–42, 2005.

[42] K. Schwaber. Scrum development process. In *OOPSLA '95: Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications*, pages 117–134, 1995.

[43] W. Shen and D.H. Norrie. An agent-based approach for manufacturing enterprise integration and supply chain management. In *PROLAMAT '98: Proceedings of the Tenth International IFIP WG5.2/WG5.3 Conference on Globalization of Manufacturing in the Digital Communications Era of the 21st Century*, pages 579–590, Deventer, The Netherlands, 1998. Kluwer, B.V.

[44] V. Sinha, B. Sengupta, and S. Chandra. Enabling collaboration in distributed requirements management. *IEEE Software*, 23(5):52–61, 2006.

[45] J. Sutherland, G. Schoonheim, E. Rustenburg, and M. Rijk. Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. In *AGILE '08: Proceedings of the Agile 2008*, pages 339–344, Washington, DC, USA, 2008. IEEE Computer Society.

[46] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov. Distributed scrum: Agile project management with outsourced development teams. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, page 274a, Washington, DC, USA, 2007. IEEE Computer Society.

[47] C. Wagner. Wiki: A technology for conversational knowledge management and group collaboration. *Communications of the Association for Information Systems*, 13(19):265–289, 2004.

[48] C. Wohlin, P. Runeson, M. Höst, C. Ohlson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An introduction*. Kluwer Academic, Boston, MA, 2000.

[49] YetAnotherForum.NET. The open source discussion forum for sites running asp.net. `http://www.yetanotherforum.net/`, October 2009.

Appendix A

Glossary

This appendix gives an overview of frequently used terms and abbreviations.

Agile Software Development Refers to a group of software development approaches based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams

APAC Asia & Pacific Region

BC Business Case

Corporate Product Lines The main product lines of Exact; Globe, Synergy, Synergy Enterprise

Corporate Product Management The management team that bundles requirements from all regions and processes them for the global development of Corporate Product Lines

EMEA Europe, Middle-East & Africa Region

Exact ADC Exact Asia Development Center

KL Kuala Lumpur, Malaysia

NL Netherlands Region

PD Product Development

PM Product Management

PMar Product Marketing

PoC Proof of concept

PTB Product Technology Board

R&I Research & Innovation

RDP Refined Development Process

Scrum An iterative incremental framework for managing complex work, commonly used with agile software development

SWEBOK Software Engineering Body Of Knowledge

Appendix B

Approach

In this appendix the approach that has been used for the graduation project will be described.

B.1 Action plan

To be able to track progress, the project had to be structurized. For this purpose, an action plan was created, which contained the following sequential steps or milestones:

1. Context
2. Challenges
3. Goal
4. Constraints
5. Initial Scope
6. Preliminary Research
7. Action
8. Interim / Final Scope
9. Research
10. Conclusions & Recommendations
11. Proof of Concept
12. Results
13. Final Conclusions & Recommendations

Steps 7 and 8 will be repeated until the actors agree that the scope is narrow enough to proceed to the Research step.

B.2 Schedule

The project has run for a total of 36 weeks. In this time frame, the last 18 (effective) weeks have been used to work on the final thesis assignment. These weeks have been subdivided into sprints.

The first two weeks have been a pre-sprint, in which answers on the thesis questions have been searched and the initial product backlog has been created.

Then, the regular sprints started off. The first sprint has lasted for four weeks to establish a protocol. The remaining sprints have had a length of two weeks and served to iteratively create the system and proof-of-concept. Table B.1 shows the final planning.

Period	**Description**
Feb 3 - Feb 20	Thesis proposal & Preliminary research
Feb 23 - Mar 13	Preliminary research & Interviews
Mar 16 - Apr 3	Preliminary research & Interviews
Apr 6 - Apr 24	Interviews & Scoping
Apr 27 - May 15	Interview documentation & Scoping
May 18 - Jun 5	Research & Impact analysis
Jun 8 - Jun 26	Assignment definition & Research
Final assignment sprints:	
Jun 29 - Jul 10	Pre-sprint
Jul 20 - Aug 28	S1: Create process description
Aug 31 - Sep 11	S2: Create environment
Sep 14 - Sep 25	S3: Run internal simulation
Oct 5 - Oct 16	S4: Run proof of concept
Oct 19 - Oct 30	S5: Run PoC & Documentation
Nov 2 - Nov 13	S6: Run PoC & Documentation
Nov 16 - Nov 27	S7: Finalize documentation & Presentations

Table B.1: Planning overview

Appendix C

Interviewees

The following tables list all interviewees and their job titles, first alphabetically and then ordered by department.

Name	Job Title
Alexander Kutilov	Senior Product Management
Dion Brands	Principal, Customer Support
Edgar Wieringa	Principal, Product Management
Emile van Bergen	Product Line Manager
Gerard van de Munt	Product Management
Jaap Jan de Lange	Principal, Product Marketing
Jorn Munnik	Functional Design
Kim ten Brink	Senior Marketing Communications
Marcel van de Sandt	Product Management Director
Marco van Dijk	Senior Product Management
Martin Ortgiess	Product Management
Michiel van Rooijen	Product Management
Nenad Borota	Principal, Software Architecture
Peter van Katwijk	Senior Functional Design
Remko Weijers	Senior Functional Design
Richard Smits	Product Management
Rob Cools	Product Marketing Director
Ronald Voets	Product Line Manager
Toine Hurkmans	Principal, Research Engineering
Vladimir Bataev	Product Management
Wiegert de Vos	Senior Marketing Management

Table C.1: List of interviewees, alphabetically ordered

Name	Job Title
Product Management	
Edgar Wieringa	Principal, Product Management
Emile van Bergen	Product Line Manager
Gerard van de Munt	Product Management
Jorn Munnik	Functional Design
Marcel van de Sandt	Product Management Director
Martin Ortgiess	Product Management
Michiel van Rooijen	Product Management
Richard Smits	Product Management
Ronald Voets	Product Line Manager
Vladimir Bataev	Product Management
Product Marketing	
Kim ten Brink	Senior Marketing Communications
Rob Cools	Product Marketing Director
Wiegert de Vos	Senior Marketing Management
Research & Innovations	
Toine Hurkmans	Principal, Research Engineering
Regions EMEA & NL	
Alexander Kutilov	Senior Product Management
Dion Brands	Principal, Customer Support
Jaap Jan de Lange	Principal, Product Marketing
Marco van Dijk	Senior Product Management
Exact ADC	
Nenad Borota	Principal, Software Architecture
Exact Online	
Peter van Katwijk	Senior Functional Design
Remko Weijers	Senior Functional Design

Table C.2: List of interviewees, ordered by department

Appendix D

Interview Questions

This appendix shows the questions that were used to guide the interviews. These questions are also mapped onto the requirements engineering entities from the SWEBOK [2].

Table D.1: List of interview questions

#	Question
	Requirements Elicitation
1	Which milestones are involved in the process?
2	Which parties are involved?
3	How is the communication arranged around the milestones?
4	How many layers of management have to be passed?
5	Are there specific methods used to elicit requirements from the field?
6	How are elicited requirements prioritized?
7	How are elicited requirements maintained?
8	Are the skills within the various stages of Requirements Engineering in Exact sufficient to let the process efficiently result in clear and unambiguous requirements?
9	Do you experience any bottlenecks in the process? If so, which?
10	What is the support for tooling? Which tools are used? Do all parties use the same tools?
11	What is done with a request when it does not fit in the strategy of the product?
	Product Requirements Document
12	What is the action flow for establishing a PRD?
13	What is the purpose of a PRD? How is it used?
14	What is the scope of a PRD?
15	How are requirements documented in a PRD?
16	How much time is normally incorporated between recognition of a requirement and finalization of a PRD?
17	At which point in time does Software Development start development of PRD items?

Continued on next page

Table D.1 – Continued from previous page

18	Will Software Development work directly with a PRD or is it sub-divided into smaller blocks to work on?
19	Which actions are taken when a requirement is not clear enough for development?
20	What will be done with the PRD when a requirement is elicited or changed during development?
21	To what extent is visual/imagery feedback used to enhance documents or communication?
	Communication
22	How often do you communicate regarding requirements? What is the (normal) duration of a conversation?
23	What is the frequency of face-to-face meetings? Is that by video connection or in person?
24	Do you communicate with one person specific or with multiple per team?
25	What are the team sizes of the parties you are involved with regarding requirements?
26	Do you experience enough feedback in the process?
27	What level of verification exists for design/tests with originator of requirement?
28	What is the collaboration like?
29	Are people a constant factor? Do you work with the same colleagues for a long period?
30	Which communication media do you use? Are there enough options and tools available?
31	To what extent is cultural diversity a problem in communication?
32	Do you run into other problems in communication? If so, which?
33	What difference is there in the usage of the PRD between Globe & Synergy?
34	Is there overlap in development teams between Globe & Synergy?
	Exact's Transition
35	How long ago did Exact start to make the requirements engineering process more agile?
36	How did you experience the transition?

Table D.2: Mapping of SWEBOK entities onto interview questions - Part 1

Requirements Phase - Entities		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Process	Models	X		X															
	Actors		X																
	Support & Management				X				X		X								
	Quality & Improvement								X								X	X	
Elicitation	Requirements Sources		X																
	Elicitation Techniques					X	X												
Analysis	Requirements Classification						X					X	X						
	Conceptual Modeling												X						
	Architectural Design											X	X						
	Requirements Allocation											X	X						
	Requirements Negotiation											X	X						
Specification	System Definition Document	X																	
	Systems Requirements Specification	X													X				
	Software Requirements Specification	X						X						X	X	X			X
Validation	Requirements Reviews																		
	Prototyping																		
	Model Validation																		
	Acceptance Tests																		

Table D.3: Mapping of SWEBOK entities onto interview questions - Part 2

Requirements Phase - Entities		19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Process	Models																		
	Actors						X	X				X					X		
	Support & Management												X						
	Quality & Improvement				X	X				X									
Elicitation	Requirements Sources																		
	Elicitation Techniques	X	X	X															
Analysis	Requirements Classification	X	X																
	Conceptual Modeling			X															
	Architectural Design																		
	Requirements Allocation																		
	Requirements Negotiation	X	X																
Specification	System Definition Document																		
	Systems Requirements Specification																		
	Software Requirements Specification		X	X												X			
Validation	Requirements Reviews								X	X									
	Prototyping								X	X									
	Model Validation								X	X									
	Acceptance Tests								X	X									

Appendix E

General Recommendations from the Interview Series

From the results of the interview series, some general recommendations can be shared.

- Improving the quality of suggestions and requirements is recommended. When customers are encouraged more by people from Support or Consultancy to state *what* they want to accomplish instead of *how*, this would result in suggestions from a functional point of view. These functional suggestions would then be a better and clearer basis to create requirements from and would be easier to compare among the regions to look for similar functionality requests.

- Another recommendation is to create a better structure for suggestion management. Some sort of system should be developed in which suggestions can at least be linked, grouped and enriched with status and tags. Such a system would make cross-regional collaboration easier, as well as traceability of the requirements and feedback on released functionality to the originating customers.

- Furthermore, with respect to planning, it is suggested that the development organization focuses less on releases, yet more on an even distribution of the various projects at hand and creating buffers to overcome situations of waiting on each other. Spreading the workload and developing continuously would level the energy needed to run the projects; to start from scratch and initiate communication everytime takes more energy.

- It is also recommended to improve overall feedback and information channels; internal as well as towards the regions, Research, and Marketing. This would increase corporate knowledge of planning, releases, roadmaps and possible issues, which could improve corporate collaboration and could make the development process more efficient, because people can anticipate already on future issues.

- A final recommendation is to keep the focus for the development organization product-oriented — so for all customers world-wide — rather than single-customer-oriented.

Involving all regions when creating requirements from a suggestion would help creating functionality that suits all regions, instead of only the originating customer. In fact, this may improve customer experience, because more customers world-wide would be able to benefit from the new functionality.

Following one of the strengths of the organization, some of these recommendations had already been picked up in the course of the interviews and are as such part of the continuous process of refining the development process.

Appendix F

Evaluation Questions

This appendix shows the questions that were used to evaluate the testing in the proof of concept phase.

F.1 Questions for initiators

The initiators were asked to grade the items listed in Table F.1. The grades should be given from the range 1-10, with a higher grade being a better result.

#	Item
1	The usefulness of the feedback loop
2	Your feeling of the improvement in quality of the PRD/FD
3	Your feeling of the improvement in quality of the product
4	Your feeling of the improvement in efficiency of the development process
5	Your feeling of the increase in speed of PRD/FD creation
6	Your feeling of the increase in speed of the development process in total
7	The clearness of the generators' feedback
8	The fit of participating in the loop with your time schedule
9	The appropriateness of the forum as medium

Table F.1: Questions for initiators

F.2 Questions for generators

The generators were asked to grade the items listed in Table F.2. Again, the grades should be given from the range 1-10, with a higher grade being a better result.

#	Item
1	The usefulness of the feedback loop
2	Your feeling of involvement in the development process
3	Your feeling of being listened to
4	The clearness of the initiator's questions
5	The fit of participating in the loop with your time schedule
6	The appropriateness of the forum as medium

Table F.2: Questions for generators

Appendix G

Version Overview

This chapter shows which changes are done after version 1.

G.1 Version 2

- Added content in chapter Scoping
- Changed heading names in chapter Scoping

G.2 Version 3

- Added Challenges and references in chapter Introduction
- Reshuffled sections and added chapters Approach and Preliminary Research
- Added content to chapter Research
- Added Glossary
- Changed planning overview in chapter Approach
- Changed content in chapter Scoping
- Added interview appendices

G.3 Version 4

- Reorganized complete thesis
- Revised and added text throughout complete thesis
- Changed planning overview in appendix Approach (App. B)

G.4 Version 5

- Added mapping of RDP onto agile practices in Scoping (Ch. 3)
- Revised text in Constraints (section 1.4)
- Added note of customer term in Final Objective (section 3.5)
- Adapted images on RDP (sections 1.1 and 5.3)
- Added text to chapter Proof of Concept (Ch. 6)
- Added chapter Discussion (Ch. 7)
- Added references and text to chapter Theory (Ch. 4)
- Added formal title pages
- Added preface
- Added abstract

G.5 Version 6

- Added evaluation of second test to chapter Proof of Concept (Ch. 6)
- Changed logo of Exact in formal title pages
- Added text to chapter Discussion (Ch. 7)
- Changed some item lists into tables
- Added appendix with evaluation questions (App. F)
- Added references to section 4.2
- Added section Validity into chapter Discussion (Ch. 7)

G.6 Version 7

- Improved section concerning the requirements for customer involvement (Section 4.4)
- Improved section concerning the fit with the current process (Section 5.3)
- Improved section concerning first feedback loop test (Section 6.3)
- Improved section concerning the evaluation of the thesis assignment (Section 7.4.1)
- Improved section concerning vision on the future (Section 7.5)

G.7 Version 8

- Replaced ‘thesis project’ by ‘graduation project’
- Replaced ‘methodology’ by ‘concept’ (Section 1.2)
- Replaced ‘agile methods’ by ‘agile approaches’
- Replaced ‘division’ by ‘department’
- Removed subsection concerning the evaluation of working at Exact (Ch. 7)
- Added section Final Closure (Section 7.6)
- Applied consistent layout to tables
- Reorganized chapters 1 and 2 into chapter Introduction (Ch. 1)
- Moved research questions to section Final Objective (Section 3.5)