UNIVERSITY OF OSLO
Department of informatics

Challenges in Enterprise
Software Integration: An
Industrial Study using
Repertory Grids

Master thesis

60 credits

Heidi Jacobsen Rognerud

May 2009

Abstract

Identifying and systematizing software practitioners' perceptions of a problem in development projects is an important first step in searching for a solution to the problem. Moreover, different tiers in an organization might have diverging perceptions of the problem. Disregarding such diverging perceptions might lead to a situation where one loses commitment from some or all of the involved parties.

In this study we investigated practitioners' perceptions about challenges in integration projects at a major software vendor, Company X. When integrating their in-house products, Company X often faces major challenges, and representatives of the company observe that substantial resources are spent on the integrations. In order to investigate these challenges, the perceptions of practitioners working on the integrations were elicited using the semistructured interview technique Repertory Grid. A total of nine interviews from three organizational tiers were conducted. The three tiers were developers, QA managers and project managers.

The interviews were thoroughly analyzed, both on an individual and on an aggregated basis, to explore what the actual challenges were, as well as possible relations between the challenges, such as the challenges' relative importance. We found that perceptions differ substantially between the tiers. While the developers' main concerns are challenges related to technology and knowledge, the QA managers perceive challenges related to human resources and responsibilities for the integration tasks as most important. The product managers, on the other hand, are most concerned about human and organizational issues.

We also discovered internal variations within the tiers. The developers seem to be most united in their perceptions, while the QA managers and product managers have a higher level of agreement across the tiers than internally.

A connection between the interviewee's education and experience, and their perceptions was also discovered in addition to minor terminological differences.

The study also gave rise to an ontology for integration challenges based on a categorization of the challenges elicited from the practitioners. The ontology has provided the company with a structure of the challenges with three levels of abstraction, and might also be an initial step for a general ontology for challenges in integrations.

Acknowledgements

First of all, I would like to thank my supervisor Jo Hannay for excellent guidance and for always inspiring me. I appreciate our discussions and his engagement has been very motivating. Thanks also to my external supervisor, Dag Lorås, for quick responses and providing information whenever needed.

Thanks to the interviewees for participating. Without their contribution and cooperation it would not have been possible to do this study.

I am also grateful to Stein Grimstad, Hans Christian Benestad and Bent Foyn for their contributions.

I would also like to thank the students and employees at Simula Research Laboratory for the great social environment!

Last, but not least, I want to thank my family and friends for their tremendous support and for always believing in me.

Heidi Jacobsen Rognerud

Oslo, Mai 2009

Contents

Abstract.....	3
Acknowledgements.....	5
List of figures.....	9
List of tables	11
1 Introduction	13
1.1 Motivation.....	13
1.2 Integration	14
1.2.1 The need for integration.....	14
1.2.2 The technical aspect.....	15
1.3 Background information	18
1.4 Main hypothesis	19
1.5 Outline.....	19
2 Research Method	21
2.1 Lean research	21
2.2 Research questions.....	22
2.3 Repertory Grid	22
2.3.1 Introduction.....	22
2.3.2 Basic parts of the technique	24
2.3.3 The process of interviewing.....	25
2.3.4 Analysis	30
2.3.5 Application of technique.....	33
2.4 Summary	37
3 Related work.....	39
3.1 So you think you know other's goals.....	39
3.2 Practitioners Roles in Software Process Improvement	40
3.3 Summary	40
4 Analysis	41
4.1 Simple relationships.....	41
4.1.1 Elements	42
4.1.2 Constructs	42

4.1.3	Developers.....	43
4.1.4	Quality assurance	55
4.1.5	Product management	65
4.2	Content analysis.....	75
4.2.1	Elements	75
4.2.2	Constructs.....	78
4.2.3	Constructs – Honey’s method	81
4.3	Terminological interference.....	84
4.3.1	Conflicts	85
4.3.2	Correspondences	86
4.4	Feedback from interviewees	86
4.5	Summary.....	87
5	Discussion	89
5.1	Addressing the research questions.....	89
5.2	Ontology	96
5.3	Summary.....	98
6	Conclusion	99
6.1	Summary.....	99
6.2	Contributions	99
6.3	Critical assessments	100
6.4	Future work.....	101
7	Appendix A.....	103
7.1	Description of elements	103
7.1.1	Developers.....	103
7.1.2	QA managers.....	105
7.1.3	Product managers	108
7.2	Description of constructs	110
7.2.1	Developers.....	110
7.2.2	QA Managers	112
7.2.3	Product Managers.....	114
8	References	117

List of figures

Figure 1-1 One-to-one integration	16
Figure 1-2 Integration bus	16
Figure 2-1 Example of a grid.....	29
Figure 2-2 Example of cluster diagram	31
Figure 4-1 Cluster diagram for developer A.....	44
Figure 4-2 Relationships between supplied constructs for Developer A	47
Figure 4-3 Cluster diagram for Developer B	48
Figure 4-4 Relationships between supplied constructs for Developer B.....	51
Figure 4-5 Cluster diagram for Developer C	52
Figure 4-6 Relationships between supplied constructs for Developer C.....	54
Figure 4-7 Cluster diagram for QA manager A.....	55
Figure 4-8 Relationships between supplied constructs for QA manager A.....	57
Figure 4-9 Cluster diagram for QA manager B	58
Figure 4-10 Relationships between supplied constructs for QA manager B.....	60
Figure 4-11 Cluster diagram for QA manager C	62
Figure 4-12 Relationships between supplied constructs for QA manager C.....	64
Figure 4-13 Cluster diagram for Product manager A	65
Figure 4-14 Relationships between supplied constructs for Product manager A.....	67
Figure 4-15 Cluster diagram for Product manager B	68
Figure 4-16 Relationships between supplied constructs for Product manager B	70
Figure 4-17 Cluster diagram for Product manager C	71
Figure 4-18 Relationships between supplied constructs for Product manager C	73
Figure 4-19 Distribution of elements to categories	77
Figure 4-20 Distribution of constructs to categories	80
Figure 4-21 Average percentage similarity with supplied construct <i>Critical – Not critical</i>	82
Figure 4-22 Average percentage similarity with supplied construct <i>Easy to handle – Hard to handle</i>	83
Figure 4-23 Average percentage with supplied construct <i>Cause - Consequence</i>	84
Figure 4-24 Relationships between terminology and concepts (from Baddoo and Hall). 85	
Figure 5-1 Hierarchical ontology	97

List of tables

Table 4-1 Elements categories with descriptions and examples	76
Table 4-2 Constructs categories with descriptions and examples	79
Table 5-1 Agreement between the interviewees.....	92
Table 5-2 Interviewees' experience and education	94
Table 5-3 Ontology for Technology	97
Table 5-4 Ontology for Process	98

1 Introduction

Section 1.1 provides the motivation for the study, followed by basic information about integrations in section 1.2. Furthermore, section 1.3 provides background information about the company, while section 1.4 presents the main hypothesis. In section 1.5 the outline of the remainder of the thesis is described.

1.1 Motivation

Quite often, when discussing issues in an organization, people start out by speaking at cross purposes, and it may take several meetings before one has arrived at a more or less common understanding of an issue. The process of arriving at a common understanding is typically tacit and implicit. It might not even be clear to the participants that such a process has taken place, or even that there were diverging conceptions in the first place.

Software professionals discussing process improvement issues are no different. In a comprehensive study [1], Baddoo and Hall found that people in different organizational tiers in software development organizations have diverging views of the concepts and importance of software process improvement.

Arriving quickly at a common understanding of the essential concepts of a problem is important in a wide range of problem-solving techniques, such as the Productive Thinking Model [2], the Plan-Do-Check-Act cycle [3-5], and the Goal-Question-Metric approach [6]. It is reasonable to assume that clarification of concepts early in the improvement process will lead to more efficient problem solving.

In this thesis, findings from a study of software integration projects in a major software vendor, Company X, is presented. Company X develops large-scale enterprise software systems. Often, situations arise where their in-house products need to be integrated with each other, or their in-house products needs to be integrated with third-party software. These integrations were perceived as problematic from several perspectives within the company, but it was not clear what the exact pain points or challenges were, what causal relations existed between the challenges, or which challenges to prioritize. It was also suspected that people with different responsibilities in the company would have different perceptions on the challenges of integration.

Interviews of professionals involved in the company's integration work were conducted. The repertory grid technique was used for concept elicitation [7-9] on a total of nine professionals from three different organizational tiers: Developer, Quality assurance

(QA) Manager, Product Manager. The responses were analyzed both individually and on an aggregated basis within each tier. This gave a picture of the main challenges, their relative importance, and their causal relationships.

1.2 Integration

The IEEE Standard Computer Dictionary defines integration as “the process of combining software components, hardware components, or both into an overall system” [10]. The relevant integrations are concerned with “making applications work together that were never intended to work together by passing information through some form of interface” [11], i.e. making two or more independent systems able to communicate and share information. Hardware integration challenges fall outside the present scope of interest.

1.2.1 The need for integration

In today’s market there is a constant need to improve workflows. Shorter time to market and faster processing of requests from customers both exert pressures to improve the workflows in a company. To meet this need it is important that information is available whenever it is needed.

Moreover, it is often a prerequisite that the available data is correct and consistent. Various departments in the organization might need to access the same information, but might not have access to the same systems. As a result, the same information may end up being stored in several systems, thus introducing the risk of inconsistent data. If these systems are not integrated appropriately, further consequences may be poor business processes as well as unbudgeted costs when reconciling the data [12].

Users of a system require quick access to data, regardless of whether the data is spread out in several systems. McKeen and Smith [13] illustrate the importance of integration with an example from customer expectations in the banking sector. Customers usually expect summary information, for instance about savings, stocks and payments, to be presented on one screen. “That the transactions cross multiple business lines, require coordination among many applications/databases ... and must be done perfectly ... is of very little concern to [the customer]” [13].

Software vendors that develop enterprise systems therefore try to design their systems so as to minimize the need for post-installment integration in the customer organization. For example, in the 1990’s the Enterprise Resource Planning (ERP) system was introduced to solve this issue. The intention of the ERP system is to support activities such as finance, human resources, manufacturing, supply and distribution in one and the same system [14]. Enterprise systems such as ERP quickly became popular: according to Markus and Tanis [15] there was a 40 per cent increase in the ERP market between 1996 and 1997. The ERP system was predicted to solve various problems in companies.

The ERP system has to some degree succeeded in meeting the need for shared data across the enterprise. That is, ERP systems solve integration issues at a certain level. However, the “one system for all” solution introduces integration problems of its own.

Normally the ERP system will exist alongside other systems. Often a company has several legacy systems that are difficult to liquidate. This may again cause overlap in functionality and duplication of data. Themistocleous et.al. [14] report from a study carried out to identify the problems with ERP systems that “the most serious problems focus on the integration of the ERP solution with existing applications such as legacy systems” [14].

Since the ERP system cannot take over for all existing systems, it has turned out to be one of the most common systems to integrate with [16]. This gives evidence that the ERP system alone is not enough to support the needs of a company.

One of the reasons why it is difficult to succeed in developing one single business application is that it is almost impossible to meet all the different needs of a company. For instance, the economy department will have different needs for customer handling than the marketing department. By having one single business system, the possibility to select the best system for each purpose is compromised. In the survey performed by Themistocleous et.al. several of the companies reported that they “faced serious problems with their business strategies, as the ERP system imposed its own way of doing business” [14].

As a result, the focus has shifted to integrating the smaller systems of the organizations with enterprise software [16]. Several software vendors try to pre-integrate third-party software or their own software with their enterprise systems before shipping the integrated package to customers. This is the type of integration that was addressed in Company X in our study.

1.2.2 The technical aspect

There are two modes of integrating systems. One solution is to integrate all the systems directly to each other, i.e. one-to-one. As illustrated in Figure 1-1 this can result in up to n^2 connections when there are n systems integrated.

The other mode is to develop an integration bus that all systems are connected to. All systems will only communicate with the integration bus, and the integration bus will make sure that a message will reach the system it is intended for. Figure 1-2 shows an example of how an integration of the systems from Figure 1-1 will look like with an integration bus.

Figure 1-1 One-to-one integration

Figure 1-2 Integration bus

1.2.2.1 One-to-one integration

Company X uses a form of one-to-one integration based on file transfer. One application writes information to a file that another application later reads. The applications need to agree on the format of the file in advance of the communication, as well as where it is located and which of the applications that will master the file. Additionally, there needs to be a locking mechanism or an agreement on the timing for reading and writing to the file to keep the applications from accessing the same file simultaneously. Currently, XML is often used as a standard language for file-based integration.

The advantage of using file transfer is that it minimizes what the integration developer needs to know about the applications to integrate. The generation of files is normally developed by the developers of the separate applications. File transfer also allows the applications to be altered as long as they produce the same format of the files. This follows the recommended principle of loose coupling, which for integrations means to “reduce the assumptions two parties ... make about each other when they exchange information” [16].

The major issue with file transfer is that the data in the files might not always be the newest data. There is a huge cost of administrating the files if new files are produced every time the data in one of the application is altered. Usually new files are produced within certain intervals. However, if one of the applications alter their data and does not write the new data to a file immediately, the other system might read and process obsolete data.

1.2.2.2 Integration bus

The integration bus acts as a bus between the systems that messages can be posted to and transferred on. As the messages are transferred between the systems they can be transformed to a format the receiving application is able to read [16].

The advantage of the integration bus is that the systems are ensured to be loosely coupled. Additionally, the risk of reading old data is eliminated. Because messaging requires a lot less administration, it is possible to update the data whenever it is needed.

The integration bus also allows for many systems to connect to the same messaging bus. With one-to-one integration every system needs an interface for each of the other systems in the integration. In file transfer this applies to the agreement of file format and access times. When an integration bus is used for the integration, every system will only be concerned about the interface to the bus.

However, it is quite costly to set up an integration bus [11], and resources need to be allocated specifically for the purpose of developing the bus. It may be a challenge to convince senior management to allocate resources for such efforts since they are not tied specifically to the development of a sales product. This relates to the general problem of

defending capital expenditures (CAPEX) when operational expenditures (OPEX) are the main priority.

The integration bus solution is growing more and more popular in integrations. The company in this study is currently developing an integration bus for their integrations.

1.3 Background information

The research was initiated by Simula Research Laboratory. Simula is a research center focusing on research relevant for the IT industry. Contact with Company X was established and three employees in Company X were given the opportunity to present challenges they were experiencing in their development projects. The integration projects arose as the most challenging projects. They had observed that substantial resources were spent on integration, and postulated that the causes of the large amount of resources spent was probably due to managerial reasons rather than technical reasons. Also, they were of the opinion that very little relevant research on integrations seemed to be available.

Based on these observations, a systems-architect in Company X, who was our primary contact in the company, presented a hypothesis about the integration projects that he wanted to be tested. His hypothesis was that it is more important to emphasize the process than the technical aspect of the integration. This hypothesis has therefore been central in the research.

In several of Company X's integration projects, the need for integration is one-sided: The owners of System A have a need to integrate with System B, whereas the owners of System B do normally not want to make the necessary changes to accommodate the integration requests from System A. As a result, the integration is normally carried out by people from the system that initiated the integration. In Company X, the enterprise system with which other systems need to be integrated is most often the Customer Relationship Management (CRM) system.

Most integration projects are run according to a waterfall model, with specification, design, implementation, test, and pilot phases. However, most projects implement iterative aspects as well, because parts of the system are tested as soon as they are testable.

The perceptions on integration activities below were stated by three representatives for Company X in the initial meetings.

1. An unproportionate amount of time is spent on integration, both in absolute terms and relative to the amount of resources allocated to integration.
2. A lot of things are not specified well enough.
3. There is little time for specification.
4. Standardization is lacking.
5. Vague responsibilities may be the reason for delays and the large amount of time spent.
6. Too much time is spent on discussing interfaces, and perhaps the data semantics.

7. Little emphasis is put on project management in the integration projects.

The three representatives presenting the observations above are all from the same organizational tier. Research shows that people on different tiers and with different responsibilities often have diverging perceptions on issues in an organization [1]. We expected that this would apply to the practitioner's perception of challenges in integration projects as well. In order to include viewpoints from practitioners with different responsibilities, it was decided to collect data from three tiers in the company. The three main tiers are developers, QA managers and product managers. The developers are responsible for the design, the programming, unit testing and documentation of the system. The QA managers, on the other hand, have responsibilities related to specification, functional testing, user documentation, training and support. In some projects the QA manager is also the project manager, and in such projects they are also responsible for allocation of human resources. The product managers have an overall responsibility for the product and contact with the customers.

1.4 Main hypothesis

The main hypothesis, put forth by the contact in Company X, is

It is more important to emphasize the process than the technical tasks.

That is, it is more important to arrange for a development process where the participants in the integration can communicate effectively about the interface, than to emphasize the technical aspect. The rejection or acceptance of the hypothesis, in terms of software practitioners' perception of it, will be based on the results of the research questions presented in section 2.2.

1.5 Outline

In this chapter we have presented the motivation for the study and basic information about integrations as well as background information about the company. Moving on, in chapter 2 the research questions will be laid down and the research method will be thoroughly described, both generally and how it was applied in this thesis. Chapter 3 will present related work where the Repertory Grid technique was applied. Next, in chapter 4 the analysis will be presented and chapter 5 will address the research questions. Chapter 6 will conclude with a summary and suggestions for future work.

2 Research Method

2.1 Lean research

The research method used for this study followed three principles that are intended to apply relevant scientific research methods to highly relevant industrial challenges in a more “lean” manner than in traditional large-scale research programs. These principles are complementary to the principles of evidence-based software engineering [17, 18].

The following principles were followed:

1. Relevant, concrete and general interest: The research objective is initiated by a desire to solve an important concrete challenge in a software development company. The challenge is perceived to be of general interest to the software industry in general.
2. Practitioners’ theories: The research questions are defined and refined by consulting practitioners working on the problem.
3. Scientific-strength research methodology: The research questions are answered by applying scientific research methods, if necessary, with the help of researchers’ expertise on such methods.

Principle 1 is reflected in work by Houdek and Mathiassen [19, 20]. In this study, Principle 1 was implemented through the meetings with the representatives from the company. During these meetings the research objective was determined:

Research objective: To lay out and systematize knowledge and perceptions within Company X on what the main challenges of the company’s software integration activities are. The inquiry should be conducted on several organizational levels, since it was postulated that major challenges are managerial rather than technical. Results should be fed back into the company, since the information gained might help the company to solve some of the challenges. Results should also be disseminated to a wider audience, since the topic should be of general interest.

The strength of Principle 2 comes from work by, among others, Argyris, Gigerenzer, Jarvis, and Schön in [21-24], and is based on the view that practitioners hold valuable implicit and explicit knowledge of their work domain that should give input to scientific knowledge. Since practitioner knowledge is often tacit, various methods for eliciting tacit

knowledge may be employed. The concept of mental models [3, 25-28] gives a framework to define and elicit tacit knowledge. In this study, Principle 2 not only underlies the research objective itself, but also determined in what way the research objective was addressed.

Principle 3 was implemented by using the repertory grid technique (see section 2.3 below). This technique is a semistructured interview technique which originates from psychology, but which has had numerous applications in other domains of research [9], such as marketing, quality control, work training, and software engineering [1, 29]. The repertory grid technique seeks to elicit a person's present construct system. In this study, the purpose was to elicit a software professional's construct system regarding software integration.

2.2 Research questions

The research objective above forms the basis for the research questions (RQ). Based on interviews of project members of integration projects, the following questions will be investigated:

- RQ 1: What are the challenges with integrations according to the developers, quality assurance managers, and product managers, and which challenges should be prioritized?
- RQ 2: Do the challenges focus on technical aspects or process-related activities?
- RQ 3: Are there differences between the organizational tiers regarding opinions on what the challenges with integrations are?
- RQ 4: Are there agreements or disagreements internally on each of the tiers about what the challenges with integrations are?
- RQ 5: Are there differences between the tiers on *how* they perceive the challenges, and are these differences most evident internal or between the tiers?
- RQ 6: Are there any connections between the practitioners' experience and education, and what they regard as the challenges in the integrations?
- RQ 7: Are there any conflicts in terminology between the tiers?

The answer to these questions will determine the acceptance or rejection of the hypothesis.

2.3 Repertory Grid

2.3.1 Introduction

The name describes the intention of the technique. "Repertory" is the formal term of "repertoire", which means "all the things that a person is able to do" [30]. A grid is simply a matrix showing the connections between the parts in the grid. A grid of a person's personal repertoire will provide a picture of what a person is able to do and the connections between these actions. This will form the person's construct system, i.e. his view of the world.

2.3.1.1 Personal construct theory

The repertory grid was invented by the psychologist George Kelly and is based on his theory of personal constructs. Personal constructs are what we get a picture of using the repertory grid technique.

Kelly based this theory on one axiom: “Man is a scientist”. By this he implied that people try to understand the world and their place in it by constantly testing hypotheses about the world. Even from we are babies we develop hypotheses on the form “if A, then B”. Though these hypotheses will affect the way we see the world, they will also change over time. Given enough information or experiences that contradicts our construct system; we will adapt the constructs according to the new information.

That we base our view of the world on experience is not the only aspect of the personal construct theory. Furthermore, Kelly stated that our experience also influences our future. For instance will a person who got sick from eating in a restaurant probably avoid this restaurant in the future. He might also be more skeptical regarding other features of the restaurant, e.g. the sanitation in the toilets, because his experience with the restaurant is bad.

Based on the personal construct theory we can say that if you understand someone’s construct system, you will not only understand their previous experiences, but also be able to predict how they will act in future situations. By comparing two individual’s construct systems we are able to determine on which level they are like each other. If two people construe the same elements in a similar manner, they are more alike and are likely to understand each other and share the same views of the world.

2.3.1.2 Kelly’s problems

In the 1930’s Kelly experienced several problems in his practice of examining people, understanding them and helping them understand themselves better [9]. He decided to develop an interview technique to overcome these problems. The intention of the technique was to make it possible to map other people’s construct systems and he based it on the theory mentioned above.

The first problem Kelly experienced was the problem of making predictions about people. Psychology was a young science in the 1930’s and the psychologists wanted to be able to make laws about human behavior. Even though many people were studied, laws derived from these studies might not apply to all individuals. Therefore Kelly wanted to be able to measure the problems his patients had, treat them and then measure again after the treatment.

The repertory grid will solve this problem because it will allow a person to map another person’s view of the world. After a treatment it is possible to perform the same measure again to measure the difference.

Further, Kelly wanted to overcome the known problem of observer bias. When trying to understand someone else’s point of view, our experience and background affect us. We

might only see the things we find familiar and ignore the rest. Observer bias might also influence the interviewee if the interviewer forces the interviewee in a certain direction. The repertory grid will reduce the observer bias because the information will be provided by the interviewee without the interviewer directing in a certain way. The result of the interview is clear and difficult for the interviewer to bias.

The third problem Kelly encountered was the problem of the role of the expert. People often rely on an expert to see the problems and suggest solutions. Yet, the people experiencing the problem are often the best to find a solution once it is clear what the problem is. This issue may be solved because the interviewee is made aware of his problems. The repertory grid will make the person's construct system explicit and perhaps reveal things that the interviewee was not aware of himself.

All of these problems are relevant today as well. Especially the problem of observer bias will always be relevant when conducting interviews.

In the next section a description of the basic parts of the repertory grid will be given, in section 2.3.3 it will be described how these parts are used in the technique while section 2.3.4 will give an introduction on how to analyze a grid interview. The last section will give a description of how the technique was applied in this thesis.

2.3.2 Basic parts of the technique

2.3.2.1 Elements

An element is “an example of, exemplar of, instance of, sampling of or occurrence within a particular topic” [8]. The elements are mainly nouns or verbs and are the basic parts of a repertory grid interview. Combinations of the elements are used to discover the meaning of a person's construct system.

Stewart [9] suggests an analogy to a surveyor when defining what an element is. When mapping out a new piece of ground, the surveyor selects key points on that piece of ground. Based on these key points he can perform measurements to gain more information about the territory to be able to draw a map. In the repertory grid the elements are these key points. When eliciting constructs we are looking at the relationship between elements and mapping the territory of a person's personal construct system.

2.3.2.2 Constructs

The essential and most important part of the repertory grid is the constructs, and these are used to describe the elements. The word construct means “an idea or a belief that is based on various pieces of evidence which are not always true” [30]. In the repertory grid this idea or belief is based on our experience and affects the way we view the world.

Kelly has himself given several definitions of a construct. One is saying that a construct is “a way in which two or more things are alike and thereby different from a third or more things” [7]. This definition is directly applied in the repertory grid when comparing elements. The constructs will then represent the dimensions that are used when

separating elements from each other. Another definition is that “a construct is a way of transcending the obvious”.

Still, the essence in constructs is that they are bipolar. One of the basic ideas in the personal construct theory is that our understanding of the world is built up of contrasts, not absolutes [8]. In order to make sense of a construct, both poles need to be defined. For instance, we cannot fully understand what a person means by describing someone as “nice” if we do not know what the person means is the opposite of nice. As Fransella et.al. [7] states, the bipolarity lies in the construct itself, not in the elements that are contrasted by the construct. This statement shows the dynamics in a person’s construct. An element might give one meaning for one construct and another for another construct.

It is not necessarily the case that the poles are the dictionary opposites of each other. In fact, Stewart [9] states that research shows that there is more meaning behind words that are not the direct opposite of another word. For instance will the construct *Happy – Furious* give more meaning than the construct *Happy – Unhappy*.

Additionally, all constructs have a range of convenience. This means that construct needs to be seen within a context. The construct can be applied to elements within that context, and to apply a construct to elements outside the context would be nonsense.

It is important to remember that a person’s construct system is a complex system, and the constructs elicited in an interview will only provide a small picture of this system. One pole may have many different opposites and the constructs may vary with the context. This will result in a network of constructs and it is not possible to get a picture of the whole network.

2.3.3 The process of interviewing

The basic parts of the repertory grid described above will together form the interview technique. A repertory grid interview can be carried out in several different ways. In the next sections a basic way of conducting a repertory grid interview, with some possible variations, will be described.

2.3.3.1 Choosing elements

The process of interviewing starts by settling the topic for the interview. The topic needs to be clear for both the interviewer and the interviewee. Based on the topic, elements are chosen.

2.3.3.1.1 Guidelines for elements

The elements can be almost anything as long as they are related to the topic. Still, there are some guidelines on how to ensure good elements. It is important to emphasize that these are just guidelines and do not need to be followed strictly.

Elements should be concrete. Abstract terms should be avoided as they possibly give more than one meaning, which may cause problems when comparing the elements to find constructs. Additionally they should not be evaluative and represent qualities.

Next, the elements should be homogeneous. The elements will later be used for construct elicitation by comparing elements, and this comparison will be a challenge with elements from different classes. It would for instance be impossible to compare *Change management* with *Project manager*, simply because the latter is a person or role and the other is an activity. However, if *Project manager* is changed to *Project management*, the activity the project manager is performing, we are comparing two activities instead.

Another guideline suggested by Stewart [9] is that elements should not be sub-sets of other elements. The reason for this is that it may be difficult to compare and contrast elements that share a lot of the same features.

Last, the elements should not include constructs. If it is possible to make an opposite of the element, it is likely to be a construct. Such elements should be avoided as they belong among the constructs.

2.3.3.1.2 Elements elicitation

The choosing of elements can be done in several ways. The least controlling way of choosing elements is to let the interviewee come up with the elements without any direction from the interviewer. The advantage of this approach is that it is ensured that the elements are representing the interviewee's view of the topic. However, this might not be what the interviewer is interested in.

Another possibility is to elicit the elements from the interviewee, for instance by asking questions or giving the interviewee different categories to choose elements from. An example of this is to ask the interviewee to mention one person she likes, one she finds interesting and so on. Such elicitation might also be done by asking questions such as "what activities in the development process to you find most time consuming?"

This way of eliciting elements has the advantage of letting the interviewee choose the elements, thus their meaning is clear to the interviewee, while it is ensured that they cover the area of interest for the interviewer. The interviewer will also be able to apply the same categories for all interviewees if several interviews are conducted. This may be a benefit in the analysis of the interviews as well. However, controlling the categories of the elements might result in important elements being missed out because they did not belong in any of the provided categories.

A third way of choosing elements is to settle on the elements in collaboration between the interviewer and interviewee. One way of doing this is to let the interviewer provide a certain number of elements and let the interviewee supply additional elements. As with the previous method of choosing elements, this will ensure that the elements represent the interviewee's view of the topic while the interests of the interviewer are still looked after.

The last and most controlling way of choosing elements is to let the interviewer provide the elements. The elements will then be based on the interviewer's view of the topic which might not correspond with the interviewee's. Elements that are important to the

interviewee might be missed out. Still, the elements will represent what the interviewer believe is most important for him or her to answer the questions the interview was intended to answer.

2.3.3.2 Construct elicitation

After the elements are chosen, the interviewer will group the elements in groups of three and ask the interviewee to compare them. The comparison is the process of eliciting constructs. The triadic elicitation method is based on Kelly's theory of how constructs are made.

To ensure that all constructs are elicited, all possible triads should be compared. However, this will probably not be possible if the number of elements exceed five or six. For a grid with ten elements there are 120 possible triads. As Fransella et.al. [7] states, the number of triads may be prohibitive. The number of triads will often be determined by the time available, or the elicitation can be completed when the interviewee does not come up with any new constructs.

The triads could be composed in a random manner or the interviewer could compose them with regard to what he or she wants to investigate.

One of the main ideas behind the repertory grid technique is that it should be free from observer bias. Still, such interviews are normally carried out for a purpose and it is the interviewer's responsibility to guide the interviewee to come up with constructs that relate to the purpose of the interview. Otherwise the interview might not give a result that is satisfactory for the interviewer.

A possible way of guiding the interviewee is to introduce the phrase "in terms of" when comparing elements. For instance, if the interviewer wants to know how the interviewee thinks about his friends when they behave at work, he or she might ask "what separates person A from the persons B and C in terms of how they behave at work?" This will guide the interviewee to come up with relevant constructs without actually suggesting constructs, hence without introducing observer bias.

The constructs are normally provided by the interviewee, but the interviewer could add constructs if they are important for the research, i.e. supplied constructs.

2.3.3.3 Laddering

The technique of laddering is also related to the area of the relevance of constructs. Stewart suggests thinking of the construct system as "a series of interlocking ladders" [9]. The ladders are getting fewer, but stronger in influence as one reaches the top. The purpose of laddering is to be able to go from general constructs to specific constructs and back again, according to what is interesting for the interviewer.

For a given construct one can go up the ladder by asking questions starting with "why". For example "why is it that *Ann* and *Eve* are effective in their work?" Approaching the top of the ladder, the constructs will be of greater importance and personal relevance to

the interviewee. On the top of the ladder is the so-called core constructs. In the Personal Construct theory the core constructs are described as fundamental constructs that a person uses to interpret the world. An example of a core construct is *Good – Bad*.

As these constructs are very personal it is recommended for inexperienced interviewers to stop the laddering before reaching the core constructs. The reason for this is that it may exist contradictions in one's core construct system. A person might cope with these contradictions simply by ignoring the fact that they contradict. Revealing the contradiction may be hard to handle for the interviewee. The core constructs might also contradict to the person's self-image.

To go down the ladder will be relevant if the construct is too global and needs to be more specific in order to give meaning relative to the research objective. This is done by asking "how" questions. For example "how is it that *Ann* and *Eve* are effective in their work?" It is important to emphasize that it is not examples of the construct that is desired, but rather new constructs that together form the global construct.

2.3.3.4 Rating

The last phase in the Repertory Grid interview is the rating. Like the other steps in the Repertory Grid, this can be done in many ways. The main idea is to let the interviewee rate all the elements in relation to each of the constructs.

The range of the scale used in the ratings can be chosen by the interviewer. Kelly usually applied a two-point scale. This will make the interviewee place the elements to one pole or the other, there is no in-between. Though a two-point scale will make the analysis easier, it will provide a black and white view of the world and one may miss the nuances.

Normally one uses a 5- or 7-point scale [8]. These scales will provide enough points to capture the nuances. Still, they will not give too wide range. The interviewee might find it difficult to differentiate between the elements if the range is too wide. In such cases the rating might not be consistent throughout the grid.

The use of a scale with an uneven number of points will allow the interviewee to place elements medial to the poles. Sometimes it is simply not possible to determine which of the poles the element is closest to. In such cases the grid would not provide the correct picture of the interviewee's construct system if he or she was forced to choose one of the poles. It is important to emphasize that the mid-point of the scale should not be used when the interviewee is unsure where to put the element or if the element is not possible to rate on the given construct.

It is sometimes the case that it is not possible to apply an element on a given construct. Such elements should be left out of the rating. To force a rating of an element that does not really apply to the construct, will be nonsense. For instance, it is impossible to apply the element *Car* on the construct *Happy – Furious*.

2.3.3.5 The grid

The result of the interview is a matrix called the grid, which usually looks similar to Figure 2-1. Each column represents an element, while the constructs are the rows. The pole corresponding to the rating value '1' is written on the left side, and the pole corresponding to '5' is written on the right side. Each number in the matrix represents the element's rating on the corresponding construct. For instance, Ann is rated as a senior on the construct *Trainee – Senior*.

Figure 2-1 Example of a grid

2.3.3.6 Depth or breadth first

If more than one person is to be interviewed, the three steps, choosing elements, construct elicitation and rating, can be performed following two different approaches.

2.3.3.6.1 Depth first

One possibility is to follow the “depth-first” approach. All the three steps are then carried out in one session for each interviewee. The advantage of this approach is that it is the interviewee’s own elements and constructs that are used. It is therefore ensured that it is clear to the interviewee what is meant by the elements and constructs because they will arise from his or her personal construct system.

Another advantage of performing the interviews depth-first is that the whole interview is carried out in one session and it is more likely that the interviewee remember what he or she meant by the different elements and constructs. Still, if the elements and constructs are clear and well-defined this should not be a problem in any case.

However, it might be difficult to compare and analyze the interviews since it is likely that the interviewees have used different elements. It is therefore difficult to perform any statistical calculations directly on the grids.

2.3.3.6.2 Breadth first

Following the other approach, “breadth-first”, the steps will be carried out in more than one session. The idea of this approach is to first elicit elements from all the interviewees, and based on these elements choose a set of elements to use for the construct elicitation. Furthermore, it is also possible to do the same with the constructs, i.e. to elicit constructs from all interviewees and choose a set of constructs that all interviewees should perform ratings on.

The “breadth-first” approach will make it easier to compare and analyze the interviews. The grids can be compared directly and statistical analyses will be easier to apply.

Furthermore, it is easier to perform a quality check of the elements and constructs. Since not all of the elements and constructs are to be used it is possible to choose the most appropriate or even alter them to make them better.

The disadvantage of the breadth-first approach is that it is time-consuming to choose which elements and constructs to eliminate from the first interview. It also needs to be ensured that the chosen elements and constructs are clear to the interviewee. Even though they are clear and give meaning to the interviewee, this meaning might not correspond to what the interviewer interprets them to be. This might lead to misunderstandings and false interpretations of the result.

2.3.4 Analysis

There are several ways of analyzing the result of repertory grid interviews. Stewart [9] suggests five main methods for analyzing: frequency count, visual focusing, principal components analysis, cluster analysis and content analysis. All methods can be used in analyzing a single grid or multiple grids. In analysis of multiple grids, the method can be applied to each interview independently and the results of these analyses can be compared to give the final result, or, as in content analysis, the analysis is applied to all interviews at once.

A brief description of the methods is given below. The methods used in this study, namely cluster analysis and content analysis, are explained in more detail. How the methods were applied and adapted in this study will be described in chapter 4 Analysis.

2.3.4.1 Frequency count

Using the frequency count method one simply counts the constructs and elements. This is especially useful when eliciting elements by using categories in more than one interview. For instance, if one category is “challenges related to process” and the other is “challenges related to technology”, it could be interesting to count the occurrence of each task for the different categories.

It is also possible to count the constructs, though it is more difficult. Different people do rarely produce the exact same constructs. The constructs may not have the same level of detail and they may not have the same poles. One possibility is to group the similar constructs, for instance through content analysis, and examine their use in this way.

2.3.4.2 Visual focusing

Visual focusing is about comparing the elements and constructs to compute an agreement score. This method is normally used when a two-point scale has been applied. Still, it is possible to apply this analysis on grids with a wider range of scale as well. Calculating the agreement score will then be more complicated.

To calculate an agreement score for the elements, the elements are compared pair wise. If the two elements belong to the same pole for a given construct, the total agreement score is increased. If two elements have a high agreement score they are more alike each other than two elements with a low agreement score.

Computing the agreement score for constructs can be done in a similar manner. An agreement score is calculated for all the constructs based on how many elements that belong to the same pole on the constructs. The constructs may be reversed, i.e. the left pole will be the right pole and vice versa, in order to achieve a higher similarity between the constructs.

The result of the agreement calculations will be a matrix. Based on this matrix it is possible to see which of the constructs or elements that is most similar. This may reveal interesting connections between the elements or constructs.

2.3.4.3 Principal components analysis

Another method of analyzing grids is the principal components analysis. It is based on the cluster analysis and displays the result in a diagram. The basic idea is to count the independent variables needed to describe the relationships within the grid and display them in a dimensional diagram. Several measures can be done on the diagram; the shape of the lines representing the constructs can be investigated as well as the groupings of constructs relative to the principal components.

2.3.4.4 Cluster analysis

The cluster analysis is an extension of the visual focusing method which is normally performed by a computer. Instead for using simple calculations, a cluster analysis program will make correlations between the elements and constructs which will give a more detailed analysis and make the calculations for grids using scales of more than two points easier.

The result of the analysis is a diagram showing the correlations, which will indicate the similarity between the elements and constructs. For the elements, the program starts by looking at the two most similar elements. These two are merged together at the point indicating their similarity score on the vertical axis on the right side. Further, the program searches for the second most similar elements and now the previous pair are treated as one element. This process is carried out for all the elements and the result is shown in the tree-structure above the grid. The same approach is used for the constructs, for which the result is the tree-structure on the right side.

The cluster diagram for the example in section 2.3.3.5 is shown in Figure 2-2. In the trees structure above the grid, we see that the two most similar elements are *Bob* and *David*. They have a similarity score of about 85%. The second most similar elements are *Ann* and *Eve*. As for the constructs, the two most similar constructs are *Effective – Slow* and *Senior –*

Figure 2-2 Example of cluster diagram

Trainee. The tree structure shows that the similarity is 80%, which might indicate that the interviewee perceive experienced people as effective and trainees as slower.

2.3.4.5 Content analysis

Content analysis is similar to the frequency count method. The difference is that the elements and constructs are not counted separately, they are assigned to categories.

In content analysis it is important to define the content unit, the basic unit to be analyzed, and the context unit, where to locate the content unit. In content analysis of repertory grids the elements or constructs are both the content unit and context unit.

The categories can either be derived from the content unit (often referred to as bootstrapping technique) or from a standard category system. A standard category system might be derived from the literature or from another theory. In the former approach, the categories are established by indentifying similarities and differences between the content units. If two units have a similar characteristic, they are placed together, thus creating an initial category. The next unit is compared to the newly established category. If it has similar characteristics as the units allocated to this category, it is joined with these. Otherwise, a new category is established.

All units are compared with the existing categories. If a new category is required, it might be necessary to redefine the established categories.

Krippendorff [31] defines two requirements for the categories, they should be mutually exclusive, that is, no units are possible to allocate to more than one category, and exhaustive, meaning that no units are left out because there is no appropriate category.

After the category system is established, the reliability should be ensured. Assigning constructs to categories is really about construing someone else's constructs [9], thus reliability is important. The threats to reliability should be avoided by introducing efforts such as performing the categorization more than once, and by involving a co-researcher.

Analysis based on the categories can, for instance, be counting of the number of units allocated to each of the categories, thus performing a frequency count. Differential analysis may be performed by indentifying subgroups and looking at the distribution of the subgroup's units to the categories.

2.3.4.5.1 Honey's method

The approach to content analysis described above does not take the ratings on the constructs into account. Thus, valuable information might be lost. Honey's method makes use of the information in the ratings and treats the constructs as more or less related to the overall topic. This is possible by supplying an overall construct to sum up the interviewees perceptions of the topic. For instance, in an interview about employees in a company the interviewer might supply the construct *Overall, good employee – Overall, bad employee*.

All constructs from all interviewees are matched against the ratings on the overall construct. Two indices are used to reflect the match, namely percent similarity score and H(high)-I(intermediate)-L(low) index. The latter index is intended to take personal variances in the construct system into account.

The procedure is similar to the procedure of the general content analysis. First the calculations are made and the constructs are labeled with their indices. Next, the constructs are allocated to categories. Within the categories, constructs on which there is a consensus in the group are investigated further. Based on the H-I-L indices it is possible to assess how the constructs relate to the overall topic.

2.3.5 Application of technique

The interviewees were selected by the contact in Company X. The criterion for selection was that they had participated in at least two integration projects. The interviewees have also been involved in different integrations, which ensured that the results are based on a diversity of integrations.

All the interviews were conducted in Company X's premises and took place during the period from October 2008 to January 2009. The developers were interviewed first, followed by the three QA managers and last the product managers were interviewed. All interviews lasted from 1.5 to 2.5 hours, and all interviews were audio recorded for the purpose of subsequent clarifications.

Before the process of interviewing was started in Company X, two researchers at Simula Research Laboratory were interviewed to test the interview technique. The purpose of these interviews was to ensure that we mastered the technique and to get experience using it.

An interview of the contact in the company was also carried out. The intention of this interview was to assure that the technique would provide suitable material and be adequate for the purpose of the study. Naturally, the topic for the interview was challenges in integration projects. Performing an interview with the same topic that the actual interviews would have was an informative exercise. It provided an idea of what to expect from the interviews with the employees in Company X.

Based on the experiences from the interview with the contact, minor changes were made to the interviewing process. These changes were the introduction of standard constructs and the decision to perform the interviews "depth first", rather than "breadth-first". The basis for these decisions will be given in the following sections where the choice of technique and its adaption will be justified.

2.3.5.1 Choice of technique

The main hypothesis in this study is that it is more important to emphasize the process than the technical issues when integrating systems. One of the questions of research is whether the different levels in the organization use different terminology. To be able to answer this question we need to know how people on the different levels use the terms.

Niu and Easterbrook have applied the repertory grid technique to discover mismatches in the vocabulary of stakeholders, and state that “analysis of terminological interference is possible only if we can discover relationships between the stakeholder’s mental models and the terms they use to describe them” [29]. Using a general interview technique it is difficult to get a picture of the interviewee’s mental models. This is, however, exactly the strength of the Repertory Grid technique.

Furthermore, the grid interview has the benefit of a low level of observer bias. When interviewing about challenges in integration projects, as in any kind of interviews, it is crucial not to bias the answers from the interviewee. Since the topic of the interviews is directed at challenges it might be tempting to make the problems bigger and more serious than the interviewee actually perceive them to be. In a general interview the questions could have been directed by the interviewer and focusing more on the problems than what the interviewee really want. Applying the grid technique will minimize this effect because it is the interviewee that provides the information with only minor directions from the interviewer.

Another benefit the grid interview provides is that information might emerge during a grid interview that the interviewee would not have thought of in an ordinary interview. When comparing elements, their qualities are put up against each other and it is often easier to express our thoughts when we can apply them on examples. Because it is rather vague what the problems in integrations are, it is possible that a triadic comparison will help this issue.

Since the process of eliciting constructs is searching and exhaustive, it is quite difficult to fake a grid interview. The grid will eventually show a part of the person’s personal construct system, and telling lies during the interview will most likely lead to contradictions in the constructs. The interviewer is therefore likely to get the truth from interviewees. Since the topic of the interview is directed at challenges and the interviewees are previous practitioners in these projects, it might be tempting to cover up their problems. In a general interview contradictions will not be as obvious as in a grid interview.

The result of a grid interview is a grid with elements, constructs and numbers. The grid will be on the same form for all the interviews, even though the elements and constructs will probably be different. Since the results of the interviews are on the same form they are easier to compare than an ordinary interview. A total of nine interviews were carried out and the fact that they had the same form made them easier to analyze.

2.3.5.2 Depth or breadth first

The interviews were conducted following the “depth-first” approach. This approach was chosen because three persons from three tiers in the organization were interviewed. Dependent on which level the person belonged to, there would probably be different views on what are the most problematic activities in integration projects, thus using the same elements for all levels will not show these differences. A developer might focus on

the technical parts, while the product manager might be more interested in the whole process. Since the result of the interview should represent their personal constructs about integration projects, it was important that they used their own terms.

2.3.5.3 Elements

The elements were pain points or challenging activities in integration efforts in the company. One might argue that to focus the elements on the problematic activities is to introduce observer bias because this might lead the interviewee into thinking that there is problems, even though he or she did not think so in the first place. The choice of focusing on the problems was made to ensure that the interviews would be useful and provide results relevant to the topic. If the elements had been any kind of activities, one might have missed the relevant elements simply because the interviewee did not think of them or thought they were interesting in this context.

In order to facilitate the element elicitation as well as the subsequent rating, the interviewee was provided with a set of cue cards (7cm x 10cm laminated cards) with a white-board pen. They were asked to write down what they considered to be the most central pain points or challenges regarding integration in the company. The interviewee was free to write down as many elements as he or she liked.

After the interviewee seemed to have no further elements to present, he or she was asked to pick out the nine most salient elements. In case of nine or less than nine initial elements, all elements were kept for further use.

The purpose of choosing the nine most salient elements was to be able to perform a quick quality check of the elements. It often turned out to be several overlapping and hierarchical elements, and the interviewee was thus given the opportunity to merge, split, or rethink elements. Although hierarchical or overlapping elements are not wrong, it is recommended to avoid this since the subsequent rating becomes more difficult for the interviewee [9].

Choosing the most salient elements also limits the number of elements to the recommended number. Jankowicz [8] suggests between five and 12 elements in most cases. It would be possible to limit the number of elements at the outset by simply asking for the nine most salient elements. However, this would require the interviewee to hold a large number of elements in working memory simultaneously, which is cognitively taxing, e.g., [32]. Writing an unspecified number of elements on cue cards, and then reducing elements afterward, is one solution to this issue.

2.3.5.4 Constructs

After the elements had been established, the nine elements were put together in groups of three to elicit constructs. The groups were composed randomly, but it was ensured that all elements were used approximately the same number of times. If there were any combinations of elements that were particularly interesting, these were put together.

Three cards with the name of the elements were placed in front of the interviewee and he or she was asked to move them around to find which of the two elements that had a common characteristic that distinguished them from the last element. Moving elements around physically is meant to facilitate the mental processes that are involved in finding similarities and differences [33]. The grouping of two of the elements generated a bipolar pair of characteristics, i.e. a construct. For example, given the elements *Lack of knowledge*, *Who does what*, and *Documentation*, the interviewee might group the first two and contrast them to the third, by characterizing the first two as *Resource related*, and the third as *Information related*. This gives the bipolar construct *Resource related – Information related*. Direct opposites, such as *Resource related – Not resource related* was discouraged, since such constructs lack meaning at one of the poles.

Laddering was performed if clarification of the constructs was needed.

The process of construct elicitation was repeated with a new set of three elements until the desired number of construct have been elicited or the interviewee did not come up with any new constructs. Note that the same set of three elements may give rise to several constructs.

In some applications of the repertory grid technique, it may be relevant to provide predefined elements and/or constructs [7-9]. In this study, three constructs were supplied for all the interviewees:

1. *Cause – Consequence*
2. *Critical – Not critical*
3. *Easy to solve – Hard to solve*

These constructs were provided in order to elicit information that may be useful for improving the challenges. The first construct was applied in order to give an indication on where to try to solve the problems; attacking the cause is, generally, better than improving the consequence. The intention of the next construct, *Critical – Not critical*, was to make the interviewee prioritize the problems according to the level of criticality. A possible problem with this construct is that the problems might be rated according to their scheduled priority. For instance, a problem that is critical, but will require a lot of work to solve may be rated as *Not critical* because other problems that are easier to solve will be prioritized. To solve this issue a third construct was provided, namely *Easy to solve – Hard to solve*. Progress is an important motivational factor in process improvement [34]. Therefore, prioritizing the challenges that are easy to handle might facilitate motivation among the practitioners.

2.3.5.5 Rating

When the elicitation of constructs was finished, the interview proceeded with rating of the elements according to each construct.

The nine cards with the name of the elements were placed in front of the interviewee. The two poles for a specific construct were written on two cards to illustrate the scale.

The interviewee was then asked to place the elements relatively to each other between the two construct poles, according to how close the interviewee felt an element associated to one or the other pole, relative to the other elements. This process was inspired by the mental model elicitation approach in [33]. Elements that were not possible to rate according to the given construct were left out.

A 5-point scale was chosen for the rating because a two-point scale would not have shown the nuances. For instance, when rating the problems according to the construct *Easy to solve – Hard to solve* one would probably not end up with two piles, but rather a rating based on a comparison of the elements.

After the rating the interviewee was asked to explain why he or she had placed each of the elements on the specific rank on the scale. This explanation resulted in two benefits. First, the interviewee was given the opportunity to verbalize their decisions. Such verbalization that is intended to reason or explain action (type 3 verbalization in Ericsson and Simon's terminology [35]), has been shown to increase performance, e.g. [36]. In all cases, this resulted in the interviewee making changes to the rating. The other benefit is that it was helpful for our understanding of the often tacit choices made by the interviewee. It was also useful to be able to listen to the explanation subsequently.

2.4 Summary

This chapter has described the principles followed to ensure lean research and how these were applied in the study. As we recall, the research objective was presented by the company, while we provided the research method. The Repertory Grid, with its elements and constructs, has been described in detail. In addition to a general description of the technique, how the technique was applied in the study was also explained. In the next chapter, similar studies where the Repertory Grid was applied will be presented.

3 Related work

A search for studies that used Repertory Grids within the context of software engineering was carried out. The results show a wide range of applications of the Repertory Grid technique. Applications range from usability of web sites [37] to knowledge acquisition [38]. This chapter will present the studies that are most relevant to this study.

3.1 So you think you know other's goals

Niu and Easterbrook applied the Repertory Grid technique to “detect stakeholders’ inconsistent use of terminology when stating goal-oriented requirements” [29]. They report a pilot study that shows how to compare stakeholders’ terminology.

In order to determine terminological interference, they investigate how the stakeholders construe the elements, which are tasks that contribute to the satisfaction of goals. To be able to make such a comparison all interviewees use the same set of elements. The constructs are softgoals, where the left pole is breaking the softgoal and the right pole is making the softgoal. The grids are compared by looking at how the tasks are rated on the constructs, hence making or breaking the softgoal.

The authors suggest four categories for relationships between the stakeholders’ terminology and concepts. If the elements are rated on different softgoals in a similar manner, they are potential correspondences, i.e. the stakeholders use different terminology for the same concepts. If two stakeholders use the same term for a softgoal, but the elements are rated in a very different manner, they are potential conflicts. A conflict is a difference in both terminology and concepts.

The constructs were compared by generating one grid with elements and constructs from all interviewees, and then performing cluster analysis. Further, differences and similarities were identified and the tasks were categorized according to the relationships between the terminologies and concepts. The results were discussed with the interviewees, who confirmed their findings on terminological interferences.

The findings of the study show that Repertory Grid is suitable for detecting differences in terminology. Still, their results did not verify that all such differences are revealed. The authors have initiated further investigations of possible differences that did not emerge in the Repertory Grid interviews.

3.2 Practitioners Roles in Software Process Improvement

Baddoo and Hall investigated “the different perceptions that different staff groups have regarding their role in [Software Process Improvement (SPI)]” [1]. They collected data from nearly 200 practitioners in groups of four to six. The elements were the three staff groups developers, project managers and senior managers. The constructs were characteristics derived from the practitioners that distinguished one of the staff groups from the other two.

Grids containing all elements and constructs for each of the staff groups were constructed. Their perceptions were analyzed using visual focusing. Each element was compared with the other elements and an agreement score was calculated. A high score would indicate a common understanding between the two staff groups. For instance, according to the developers, the senior managers and project managers have most in common. Based on how the staff groups rate the elements on the constructs, they draw conclusions about how the groups perceive the other groups.

Their findings show that “all staff groups appear to be defining their SPI roles in relation to what they already do in software development” [1]. They identified a perception among the developers that SPI is the management’s responsibility. This delegation of responsibility violates the recommended empowering of developers in SPI processes. Additionally, they showed that all groups agree that the developers and senior managers have least in common, while the project managers and senior managers have most in common.

3.3 Summary

Niu and Easterbrook has provided a method for how to detect differences in stakeholder’s terminology, and a similar approach will be used in exploring terminological interference among the interviewees in our study.

The findings from the study of Baddoo and Hall will be used to compare our findings related to similarities and differences between the tiers regarding perceptions of the challenges.

The topics of the previous chapters, namely the background information, the research method and related work, form the basis of the study. In the next chapter, the analysis of the interviews will be presented.

4 Analysis

The analysis of the individual interviews is organized in the following manner: for each interviewee the elements and the constructs are analyzed respectively followed by a summary of the most important results from the analysis of the interviewee. First the developers are analyzed, followed by the QA managers and last the product managers are analyzed. A summary of the tier is given after all interviewees for the specific tier are presented.

Section 4.2 presents the content analysis of both the elements and the constructs.

First, a description of how the individual analysis was carried out is presented.

4.1 Simple relationships

The relationships between, respectively, elements and constructs were analyzed using cluster analysis. Figure 4-1 shows a cluster diagram for Developer A. The trees on the top and right side of the diagram are called dendrograms and show the relationships between the constructs and elements in a visual manner. The grid is rearranged and the most similar constructs and elements are placed next to each other. The cluster diagram is useful for visualizing the correlations and to draw conclusions based on more than two constructs. The cluster diagrams were generated in Web Grid of the centre for Pearson-Computer studies.

The disadvantage of a cluster diagram is that it groups the two most similar constructs (or elements) and treats them as one in the subsequent comparisons. Relationships with a few, specific constructs are difficult to emphasize in cluster analysis and one might miss out on findings where two interesting constructs are grouped in separate clusters. The analysis of relationships between constructs has therefore been supplied with a second metric, namely matching score. The matching score is based on single comparisons between each of the elements and constructs and will therefore cover the correlations missed out by the cluster analysis. This is useful when there are specific relationships of interest, for instance relationships between supplied constructs and the interviewee's constructs.

The matching score is calculated using the city block metric formula. The city block metric represents the absolute distance between two constructs when all elements are considered. The formula for the city block metric is:

$$CB_{xy} = \sum_{i=1}^{\epsilon} |G_{(x)} - G_{(y)}|$$

Formula 4-1 City block metric

The formula above is from the help manual of the analysis tool Idiogrid, which uses this metric in its calculations.

When using matching scores for the analysis it should be determined what to consider as a high correlation. However, such a limit is difficult to set universally because of personal differences in typical similarity scores [8]. Some people might have 80% as the highest similarity score; while others have 80% as the lowest similarity score and a 100% match as the highest. To take personal variances into account, the percent limit was not fixed; rather the 3-4 constructs with the highest similarity score for each supplied construct were analyzed. In case of low similarity scores among the selected constructs, the ratings on the constructs were looked into to ensure that it was possible to draw conclusions based on the correlations.

The cluster analysis and analysis using matching score will not always give the exact same result, but used together, they will cover the most interesting relationships within the grid.

4.1.1 Elements

The elements indicate *what* the interviewees think about the topic. We are interested in the relationships between the elements because this might identify groups of elements and their common characteristics. Such groups might hold valuable information when improvement actions are initiated. For instance, two challenges with common characteristics might be possible to solve together.

Clustering of the elements might also give an indication of the diversity of the interviewee's elements, whether he or she has chosen a lot of different elements or has more narrow opinions.

4.1.2 Constructs

The constructs indicate *how* the interviewee thinks about the elements. When looking at the relationship between constructs, we are looking for similarities in how the interviewee thinks about the elements. A similar rating on two constructs might indicate that

1. there is a connection between the two constructs,
2. they have similar meaning, or
3. the correlation is accidental

An example of a connection might be that all elements on the construct *Effective - Slow* have the same rating as on the construct *Senior - Trainee*, which might indicate that the person from whom the constructs was elicited perceives senior managers as more effective, and trainees as slower. An example of the second type of similarity might be a

correlation between *Trainee – Senior* and *Beginner – Experienced*. This relationship is most likely due to the interviewee perceiving these constructs as having a similar meaning.

In this study, the relationships between the constructs will give information about relationships on how the interviewee perceives the challenges. Such relationships might be valuable for identifying characteristics of the challenges.

The supplied constructs are used by all interviewees, and will therefore make the comparison of grids more interesting. Another reason is that they are in fact supplied because they are believed to be interesting for the research. It is therefore natural to emphasize them in the analysis.

In the subsequent sections, the analysis will be presented per interviewee. For each interviewee, the cluster analysis of the elements will be presented first, followed by the cluster analysis of the constructs. Last, the matching scores will be used to emphasize the relationships to the supplied constructs. Together, these analyses will give a thorough description of the relationships between both the constructs and the elements. There will also be given a summary of each interviewee and an interpretation of the results. A diagram illustrating the elements' rating on the supplied constructs is presented in the summary. These diagrams were a part of the feedback to the interviewees, see section 4.4. Due to limited space, the element's names are shortened. The full names are displayed in the cluster diagrams.

Elements and constructs that are not intuitive are briefly described in the analysis. For a complete explanation of elements and constructs, see appendix A.

4.1.3 Developers

4.1.3.1 Developer A

4.1.3.1.1 Cluster elements

The dendrogram has two main groups with two elements outside these groups, as shown in Figure 4-1.

The first group contains the elements *What information* and *Meaning*, which is related to what information to exchange and the meaning of the concepts in the systems. They are rated maximum one point in difference on the constructs. Because they are not rated on three of the constructs they end up with a similarity score of 65% (web grid performs pairwise deletion when calculating scores). The elements in this group are critical and active actions. They are also causes and related to the information *in* the system rather than information *about* the system.

The second group consists of the three most similar elements *Documentation*, *Lack of knowledge* and *Who does what*, which have a similarity score of 82.5%. *Missing flow of information* and *Low amount of recycling* are also included in this cluster. All these elements are challenges the company can affect and they represent usage of resources.

Figure 4-1 Cluster diagram for developer A

The challenges in this group are, as opposed to the first group, related to information *about* the system, rather than information *in* the system, and they are quite easy to handle.

The two outsiders, *Errors in external components* and *Waiting for technology*, are closer to the second group than the first, but show no significant correlation.

4.1.3.1.2 Cluster constructs

As seen in Figure 4-1 the constructs *Critical* – *Not critical* and *Active action* – *Passive action* are the two most similar constructs. Together with the constructs *Something we can affect* – *Something we cannot affect* and *Easy to handle* – *Hard to handle* they form a cluster. This clustering might indicate that the critical challenges are easy to handle. The correlation between *Easy to handle* – *Hard to handle* and *Something we can affect* – *Something we cannot affect* might be that the constructs have similar meaning to the interviewee or that there is a connection between them, that is, the reason why the challenges are hard to handle is that they cannot affect them.

The constructs *Information about the system* – *Information in the system* and *Usage of resources* – *Agreement* form a small cluster. This clustering might indicate that challenges related to information about the system are related to usage of resources, for instance that they use resources on gaining information about the system, while challenges associated to information in the system are just matters they have to agree on.

The two clusters described above are grouped together in one larger cluster. Within this cluster we see that information about the system is easier to handle than information in the system. The reason for this might be that the information *in* the system is unchangeable (because the systems are huge and difficult to change) and they simply have to agree on how to use the information in the best possible way. Information *about* the system, on the other hand, might be seen as easier to handle because it is not fixed in any way. The way the knowledge about the systems is shared might be more available for change than information in the system.

Next, the three constructs *Information - Technology*, *Information related – Resource related*, and *Information gaining – Information spread* are joined with this larger cluster.

Cause – Consequence is the construct with least similarity with the other constructs. This would indicate that whether the challenge is a cause or a consequence has little to do with the other constructs from the interviewee's perspective.

4.1.3.1.3 Matching score

4.1.3.1.3.1 Critical – Not critical

The construct with the highest score of similarity with the supplied construct *Critical – Not critical*, is *Active action – Passive action*. The similarity score for this pair of constructs is 80.6%. The developer defines a passive action as absence of a specific action and an active action as actually performing a specific action. The active actions are challenging actions that are carried out in the integration projects, but they are challenging in order to succeed with the integration task. This might make the active challenges critical. The passive actions might be actions that would make the integration projects less challenging if they were carried out, but since they currently are passive, i.e. not carried out, they are still able to execute the integrations, and therefore they are not critical. The correlation between criticality and activeness of the challenges does not apply to the challenges *Missing flow of information* and *Lack of knowledge* which is seen as passive, critical actions.

The constructs *Critical – Not critical* and *Easy to handle – Hard to handle* have a similarity score of 77.8%. This similarity indicates that the challenges that are most critical to overcome in order to succeed with the integration are the ones that are easiest to handle. However, it should be noticed that the two most critical elements, *Meaning* and *What information*, are rated midpoint on the *Easy to handle – Hard to handle* construct. As for the rest of the elements, the correlation indicating that the critical challenges are easier to handle than the not critical ones, is strong.

Next, the construct *Something we can affect – Something we cannot affect* is given a similarity score of 69.4% to the supplied construct. This suggests that the most critical challenges are the ones that the company can affect.

4.1.3.1.3.2 *Easy to handle – Hard to handle*

The construct *Something we can affect – Something we cannot affect* have a high similarity score with *Easy to handle – Hard to handle*. The correlation between these constructs might be because the challenges that are easy to handle are in fact easy because they can be affected by the company. For each element the developer has rated the elements maximum one point difference between these constructs.

Passive action – Active action has a score of 63.9%. The interviewee perceives the passive actions as easier to handle than the active actions. Since the interviewee defines passive actions as not performing a specific action, this might indicate that the interviewee finds it easier to activate passive actions than to alter the active actions they currently carry out.

However, this correlation says the opposite of the cluster analysis, where the active actions were said to be easier to handle than the passive actions. This contradiction is an example of the two analysis methods not always giving the same results. Because the analysis gives a contradictory result, it is most likely not a strong correlation. To solve this contradiction it is possible to look into the ratings on the two constructs. In this case it shows that half of the elements indicate that the active actions are easier to handle, while the other half indicate that the passive actions are easier to handle. It is therefore not possible to draw any conclusions from this correlation.

Further, it is a high correlation between the constructs *Technology - Information* and *Hard to handle – Easy to handle*. According to the developer, the challenges concerning information about the systems are easier to handle than the technological challenges. The technological challenges, such as *Waiting for technology*, would require a lot of resources to solve. Other technological challenges, such as *Errors in external components*, are external and therefore difficult to handle for the company because they do not have the power of influence, and they are not the ones in control of the resources needed to solve the challenges.

On the other hand, challenges related to information about the system seem to be easier to handle from the point of view of the developer. The developer also states that the challenges related to information are more critical than the technical challenges.

4.1.3.1.3.3 *Cause – Consequence*

None of the constructs have a high similarity score with the construct *Cause – Consequence*. The most similar construct is *Easy to handle – Hard to handle* with a similarity score of 52.8%. This similarity would indicate that the challenges that are the causes for difficult integrations are easier to handle than the challenges being the consequences of challenging integrations. The correlation applies to most of the elements, except *Waiting for technology* and *Lack of knowledge*.

Figure 4-2 Relationships between supplied constructs for Developer A

4.1.3.1.4 Summary

The main results in the analysis for Developer A are that the interviewee experience that lots of resources are used on acquiring knowledge about the system to integrate with. Fortunately, the challenges that are associated with information about the systems are easier to handle than the technological challenges.

Furthermore, the interviewee perceives the critical challenges are easier to handle than the non-critical challenges. These challenges are related to information, both about the system and in the system. Therefore, it seems that the interviewee finds the challenges associated to information as the most important issues to deal with. This might be a good starting point for an improvement process.

Figure 4-2 shows the elements' rating on the supplied constructs. The vertical and horizontal axis illustrates, respectively, *Easy to handle – Hard to handle* and *Critical – Not critical*, while the point color tone indicates the rating on *Cause – Consequence*.

As seen in Figure 4-2 the challenges pointed out to be easiest to handle and still critical are *Missing flow of information*, *Low amount of recycling*, and *Lack of knowledge*. The former challenge is also a cause, which might make this a challenge that is natural to prioritize. *Missing flow of information* is related to communication regarding the technical solution between the participants in the projects.

4.1.3.2 Developer B

4.1.3.2.1 Elements

There are three main groups in the dendrogram in Figure 4-3. The cluster with the highest similarity score is *Integration interface*, *Responsibility for the integration is with another system*, *Dependencies of versions* and *Missing integration module, low amount of recycling*. All the elements in this cluster have the same rating on four constructs.

They are all related to the content of the solution and the logical and semantical aspects. The challenges are easy to handle, but still critical.

All elements in this cluster, except *Integration interface*, are about design of the solution, whereas *Integration interface* is associated to the integration itself. All elements are rated as a cause, except *Dependencies of versions* which is a consequence.

The next cluster consists of the two elements *Have to be running all the time* and *Different technology platform*, none of which are critical. On three constructs, the elements are related to technology.

The last cluster, which is least similar to the other clusters, consists of *Synchronization* and *Different data models, different semantics*. Both challenges are hard to handle. They are related to exchange of data. At constructs where one of the poles is technical, they are both rated at the non-technical pole of the constructs.

The element *Documentation, technical and usage* is not clustered with the other elements at all. This is because the analysis cutoff is set to 25%, meaning that elements that have a similarity score of less than 25% with any of the other elements are not clustered. In this case, the low similarity score is probably due to all the constructs the element is not rated on.

Figure 4-3 Cluster diagram for Developer B

4.1.3.2.2 Cluster constructs

The two most similar constructs are *Contents of the solution – How the solution is technologically* and *Logical/semantics – Solution*. This correlation is probably due to the constructs having similar meaning to the interviewee. Challenges related to logic and semantics is probably also related to contents of the solution, while the solution itself is the technology used in the integration, i.e. how the solution is technological.

Added to this pair of constructs is *Hard to handle – Easy to handle* and *Not critical – Critical*, together forming a cluster. The clustering of these constructs would indicate that the interviewee sees the challenges that are related to the solution itself as critical, thus easy to handle. Contrary, the challenges that are associated with the semantics and contents of the integration are hard to handle, but not critical.

Three constructs form another cluster, namely *Knowledge about the systems – Exchange of data*, *Ability at the receiving system – Interaction between systems*, and *Responsibility - Technology*. The two first constructs have one similar pole, that is, exchange of data is related to interaction between systems. Therefore, the correlation between these constructs is probably caused by the similar meaning of these poles. Two of the elements also indicate that the interviewee sees a connection between knowledge about the systems and ability at the receiving system. Ability to integrate is about the services the system has to offer for an integration. The correlation with knowledge might be that the developers have to have knowledge about the services the other system offer in order to integrate with them.

The correlation with the third construct in this cluster, *Responsibility – Technology*, shows that the interaction between the systems and exchange of data is related to the technology. On the other hand, the responsibility for arranging for the integration has a connection to the system's ability to integrate. This might indicate that the interviewee thinks that if the responsibility for the integration is delegated to the participating systems and they all offer relevant services for integration, the integration task would be less challenging.

The constructs that have the least similarity with the other constructs are *Cause – Consequence* and *The gate of the system – Knowledge about the systems*, which would indicate that the interviewee sees no similarity between these constructs and the other constructs from the grid.

4.1.3.2.3 Matching score

4.1.3.2.3.1 Critical – Not critical

The developer perceives the critical challenges are easier to handle than the less critical ones. The only element this correlation does not apply to is *Synchronization*, that is, the challenge regarding who is the owner of the data and what to do in case of simultaneous modification of data. *Synchronization* is seen as both difficult to handle and critical.

There is also a correlation indicating that the challenges regarding the solution are more critical than the logical and semantical challenges.

4.1.3.2.3.2 Easy to handle – Hard to handle

The constructs *Solution – Logic/semantics* and *Easy to handle – Hard to handle* have a strong correlation for both poles with a similarity score of 75%. *Solution* is defined as the technologies chosen for the integration and the integration itself, seen from a technical aspect. The other pole, *Logic/semantics*, is about the information and data in the integration, and rules for how the systems should interact. This indicates that the interviewee sees the challenges related to the technical aspect are easier to handle than the semantics.

Supporting the previous section, there is a correlation between *How the solution is technological - Contents in the solution* and *Easy to handle – Hard to handle*, indicating that the technological challenges are easier to handle than the challenges related to the contents of the integration. The only element this correlation does not apply to is *Different technological platform*, which is both technological and difficult to handle.

4.1.3.2.3.3 Cause – Consequence

None of the constructs show a strong correlation with *Cause – Consequence*. The construct with the highest similarity score is *Ability at the receiving system – Interaction between systems* with a score of 50%. The low similarity score is due to three of the elements not being rated on this supplied construct. The rest of the elements show a medium correlation indicating that the causes are associated with the systems' ability to integrate. The interviewee sees the challenges related to the interaction between the systems as the consequences of challenging integrations.

4.1.3.2.4 Summary

The technological challenges are the most important issues, according to the interviewee. They are perceived as easy to handle as well as critical, and are therefore natural to prioritize in an improvement process. Contrary, the logical aspect of the integration is not as important.

As seen in Figure 4-4 challenges related to the responsibility for the integration and the interface the systems offer for an integration, are critical, but easy to handle. These challenges are therefore possible starting points for an improvement process.

Figure 4-4 Relationships between supplied constructs for Developer B

4.1.3.3 Developer C

4.1.3.3.1 Elements

In the cluster diagram of the elements for this developer, there are two main clusters. The cluster with the highest similarity score consists of the elements *Challenges related to similar look and feel as the target system*, *Challenges related to authentication against different ERP systems*, *United installation of integrated systems* and *Handling different properties when integrating with more than one system*. All the elements in this cluster are closer to the technical poles on constructs where one of the poles is technical. They are all related to user experience rather than development process, and none of the elements are important or critical. This indicates that the technical challenges and those related to user experience are perceived as non-critical.

The other cluster consists of *More testing when integrating with more than one system*, *Problems related to handling errors in target system*, *Changes in target system*, *Missing documentation and examples of new target system*, and *Challenges to find people with knowledge about the target system*. The challenges in this cluster are mostly related to the development process, and they are more critical than the other cluster. They are also more related to quality assurance than the technical aspect.

4.1.3.3.2 Cluster constructs

The dendrogram in Figure 4-5 shows that the constructs for this interviewee have an overall high correlation. All constructs have a similarity score over 70%.

The two most similar constructs are *Important – Not important* and *Critical – Not critical*. This correlation is probably due to the interviewee perceiving their meaning as similar. Another pair of constructs with a high similarity score is *Quality assurance – Technical* and *Test - Development*. Test and quality assurance is most likely related to the same aspects, i.e. testing is a form of quality assurance. Technical and development is probably correlated because the development is about developing the technical solution.

Figure 4-5 Cluster diagram for Developer C

Together with the previous two constructs, *Understanding of the system – Technical* and *Organization and human resources – Technical* forms a cluster. All the right poles of the constructs are either development or technical. The correlation between the left poles is not as strong as for the right poles because a lot of the elements are rated on the middle of the scale. It is therefore not possible to draw any conclusions based on these correlations.

Easy to handle – Hard to handle, *Cause – Consequence*, and *Independent of number of systems – Consequence of more than one system* are the constructs that have the lowest similarity score with the rest of the constructs.

4.1.3.3.3 Matching score

4.1.3.3.3.1 Critical – Not critical

Important – Not important and *Critical – Not critical* have a similarity score of 91.7%. They probably have a similar meaning to the interviewee. Three of the elements are rated with one point in difference on the two constructs. One reason for the differences might be that the interviewee does not see the constructs as identical, only very similar. The differences might also be due to natural inconsistencies in the construct system.

Further, the user experience is regarded as less critical than the development process. The reason for this might be that the interviewee does not see it as required that the user experience is good in order to achieve a successful integration. Because he is a developer he might also see the challenges that are directly related to him and his tasks as more critical.

The developer perceives the technical challenges are less critical than the challenges related to quality assurance. This correlation is especially applicable to the technical elements *Challenges related to authentication against different ERP systems* and *Challenges related to similar look and feel as the target system*, which are seen as less critical. The elements related to quality assurance, *More testing when integrating with more than one system* and *Problems related to handling errors in target system*, are more critical. Note that a lot of the elements are placed in the middle of the scale, i.e. given the ratings 2, 3 or 4, which will give a weaker relationship than if they were closer to the poles.

4.1.3.3.2 *Easy to handle – Hard to handle*

The challenges related to test seem to be more difficult to handle than those related to development, indicated by a similarity score of 83.3% with the construct *Development – Test*. This relationship is also supported by a correlation with *Technical – Quality assurance*.

Easy to handle – Hard to handle and *Technical – Understanding of the system* have a similarity score of 80.6%. The developer has four constructs where one of the poles is *technical*. In three of them the technical challenges are seen as easy to handle. Access to information is easier to handle than technology, according to the developer.

Besides indicating that the technical challenges are easier to handle, the construct *Technical – Organization and human resources* also shows that the interviewee sees the organizational challenges as harder to handle.

4.1.3.3.3 *Cause – Consequence*

As opposed to the other developers, this developer has a quite high similarity score between *Cause – Consequence* and two of the other constructs. One of these constructs is *Access to information – Technical* with a similarity score of 72.2%. This similarity would indicate that the interviewee see the challenges related to access to information as the causes of the challenging integrations, while the technical challenges are consequences.

The developer is, unlike the other developers, working on integrations where more than two systems are integrated. One of his constructs is *Independent of number of systems – Consequence of more than one system*, which means whether the challenge is a challenge only because they are integrating with more than one system, or it would be a challenge independent of how many systems they are integrating with, hence also only one system. This construct has a similarity score of 72.2% with the supplied construct *Cause – Consequence*. From this correlation one might draw the conclusion that the causes are independent of the number of systems in the integration, that is, they would be challenging even if they were integrating with only one system. The consequences, on the other hand, are challenges because they are integrating with more than one system at a time.

Figure 4-6 Relationships between supplied constructs for Developer C

4.1.3.3.4 Summary

Developer C sees access to information as the cause of challenging integrations. This perception applies both to documentation and finding people with knowledge about the system they are integrating with. The technological challenges, on the other hand, are consequences and quite easy to handle.

The interviewee perceives the technical challenges as less critical than the challenges related to test and quality assurance. Testing and quality assurance as well as organizational challenges are also harder to handle.

The challenges related to documentation of the system they are integration with are easy to handle, and still quite critical. As seen in Figure 4-6 they are also causes and are therefore possible starting points for an improvement process.

4.1.3.4 Summary developers

From the individual analysis it is evident that two of the developers tend to agree on an issue while the third disagrees. It is not the same developer who always disagrees with the others; rather it varies from case to case who agrees and who disagrees.

Regarding the technical challenges, two of the developers see them as easy to handle. One of these developers has several constructs where one of the poles is technical. Most of the constructs indicate that technical challenges are easy to handle, but challenges related to information seems to be even easier. The latter statement is supported by the developer who disagrees with the two first developers. He thinks that technology is hard to handle, while information is easy.

Technical challenges are seen as critical by one of the developer, while the other developers do not think it is critical.

Information or knowledge about the system also seems to be important to the developers. Two of the developers find information as a challenge that should be prioritized.

Frequent constructs among the developers are technical against various other poles, such as testing, organization and resources. Every developer has at least two constructs where one of the poles is technical. A frequent aspect to include in the constructs is resources. Two of the developers presented a construct that is related to resources and the usage of resources.

4.1.4 Quality assurance

4.1.4.1 QA manager A

4.1.4.1.1 Elements

As seen in Figure 4-7, the dendrogram of this grid shows three distinct clusters. The first cluster consists of the two most similar elements, namely *Resources (shared and timeboxed)* and *Time*, which have a similarity score of 86.1%. Added to this cluster is *Silo*, which is related to protecting their own system, at a score of 55.6%. These challenges are causes and they are rated midpoint on the criticality scale. They are all hard to handle and are seen as organizational rather than technical challenges.

The next cluster consists of the elements *Owner of the integration*, *Person (chemistry)*, and *Creative*. These elements are all about communication, organization and resources rather than the technical aspect.

The last cluster in this grid contains the elements *DB (data model)*, *Architecture api*, and *Requirements*. These elements are technical challenges and they are placed on the middle of the scale for both criticality and how easy they are to handle. They are about ability to integrate rather than will to integrate. The two first elements in this cluster are causes, while the third is placed in the middle of the construct *Cause – Consequence*.

Figure 4-7 Cluster diagram for QA manager A

4.1.4.1.2 Cluster constructs

As shown in the dendrogram in Figure 4-7, the two most similar constructs are *Technical – Resources* and *Technical – Organizational*, with a similarity score of 90%. Added to this cluster is *Technical – Communication*. The similarity between the left poles of these constructs is rather obvious as they are all technical. The right poles are more about the non-technical aspects of the integrations.

The second most similar constructs are the supplied constructs *Easy to handle – Hard to handle* and *Critical – Not critical*. This would indicate that the interviewee sees the critical challenges as easier to handle than the non-critical challenges. The constructs *Cause – Consequence* and *Ability to integrate – Will to integrate* are added to this cluster, indicating that one of the causes of integrations being challenging is the system's ability to integrate. The causes are critical and easy to handle. On the other hand, there is no correlation indicating that the system's will to integrate is a consequence of challenging integrations. There is only one element rated as *Will to integrate*, namely *Silo*, which is a cause and hard to handle.

The constructs *Execution – Communication* and *Availability of resources – Usage of resources* have a similarity score of 50%. The relatively low score is probably due to the three elements not being rated on these constructs. The elements that are rated on these constructs receive quite similar ratings. This might indicate that the execution is dependent of the availability of resources, in other words, that the challenges associated with the execution are challenging because they might not have the resources they need. On the other hand, the challenges associated with the communication are more about how they use the resources they have available, hence the challenge related to communication is not due to lack of resources.

4.1.4.1.3 Matching score

4.1.4.1.3.1 *Critical – Not critical*

Easy to handle – Hard to handle and *Critical – Not critical* have a strong correlation, i.e. the critical challenges are easier to handle than the less critical ones. This correlation applies to every challenge except the challenge of chemistry between people which is both quite critical and quite hard to handle.

There is a correlation between the constructs *Cause – Consequence* and *Critical – Not critical*. However, most of the elements that are causes are rated midpoint on the criticality scale. None of the consequences are critical. From the previous correlation (between *Easy to handle – Hard to handle* and *Critical – Not critical*) we can derive that the most critical challenges are also the ones that are hardest to handle, thus the causes are most critical and hardest to handle, while the consequences are easier to handle, but also less critical.

4.1.4.1.3.2 *Easy to handle – Hard to handle*

The technical challenges are easier to handle than the organizational challenges, except for the challenge of having a clear owner of the integration.

Further, the usage of resources is easier to handle than the availability of resources. More detailed, it is more difficult to allocate the resources needed than to actually use the resources that are available. This might indicate that there is a lack of resources in the integration project, and a struggle to allocate the resources needed. This correlation does not apply to the challenge of allowing the developers to be creative.

4.1.4.1.3.3 Cause – Consequence

None of the constructs have a significant correlation with the supplied construct *Cause – Consequence*.

4.1.4.1.4 Summary

The cause of challenging integrations is the system’s ability to integrate according to the interviewee. Thus, it is not lack of will to integrate that is the cause, but rather technical obstacles.

The technological challenges are perceived as easier to handle than the organizational challenges. Still, the technological challenges are the most critical. It would therefore be advisable to emphasize the technical challenges when initiating improvement actions.

As illustrated in Figure 4-8 the interviewee also sees the challenge of having a clear owner of the integration as the most important issue. Fortunately, this is also the easiest challenge to handle and would be natural to prioritize in an improvement process.

Figure 4-8 Relationships between supplied constructs for QA manager A

4.1.4.2 QA manager B

4.1.4.2.1 Elements

As shown in Figure 4-9, this grid has three main clusters in addition to one single element with no significant correlation to the other elements. The cluster with the highest similarity is *Responsibility*, *Communication*, *Delivery*, and *Test*, where the first two elements have an identical rating on all constructs. The lowest similarity score in the cluster is 93.8%, which is remarkably high.

The two first elements, related to responsibility for the integration and communication among the practitioners, are seen as hard to handle, while all elements in the cluster are critical and internal. The challenges are related to the process and are consequences. They are also seen as continuous activities rather than technical issues.

The next cluster consists of two elements, *Competence* and *Semantical differences in applications*. They have a similarity score of 56.2%, and are both causes and internal. On the construct *Easy to handle – hard to handle*, the elements are placed in the middle.

The elements *Dependencies of versions* and *Differences in architecture* make up the next cluster. They have the same ratings on all constructs, but because they are not rated on one of the constructs they receive a similarity score of 87.5%. They are both related to the technical aspect rather than to process. Both are causes, as well as critical and hard to handle. The elements are also internal challenges that are functional rather than technical.

The element *Requests from customer* are not clustered with any of the other elements. The challenge is easy to handle, not critical and a consequence.

Figure 4-9 Cluster diagram for QA manager B

4.1.4.2.2 Cluster constructs

The dendrogram in Figure 4-9 shows two distinct clusters. The two most similar constructs are *Not critical – Critical* and *External – Internal*, the similarity score being 94.4%. This would indicate that the interviewee sees the external challenges, for instance those involving the customer, as less critical than the challenges internal in the company. It should be noted that the interview presented only one external challenge.

Easy to handle – Hard to handle and *Functional – Technical* are added to the previous constructs and forms the first main cluster. Based on the clustering it is evident that the interviewee sees the technical challenges as harder to handle than the functional challenges, but still more critical. The technical challenges are internal, as opposed to the functional challenges which involve external parties.

The other cluster consists of the constructs *Cause – Consequence*, *Technical – Process*, *Resources – Process*, and *Technical – Activity*. The clustering would indicate that the interviewee perceives the technical challenges as the causes of the challenges experienced in integration projects, and the challenges associated to the process as the consequences.

4.1.4.2.3 Matching scores

4.1.4.2.3.1 Critical – Not critical

The critical challenges are internal, indicated by the correlation with *Internal – External*. The one challenge that is external is not critical, but it should be noted that the interviewee has rated all elements, except two, as critical.

The less critical challenges are easy to handle, and the challenges that are hard to handle are critical.

The correlation with the construct *Technical – Functional* shows that the technical challenges are critical while the functional are not critical.

4.1.4.2.3.2 Easy to handle – Hard to handle

The functional challenges are easy to handle and the technical are hard to handle.

The external challenges are easy to solve and the internal are hard to solve. Still, since there is only one external challenge, there should not be put much emphasis on this relationship.

4.1.4.2.3.3 Cause – Consequence

The technical challenges are the causes of challenging integrations and the process elements are the consequences, according to the QA manager. This indicates that if the technical challenges were solved, the process might have been less challenging as well. However, this correlation does not apply to the challenges related to lack of competence.

Another pair of constructs supporting the previous correlation, is *Technical – Activity* and *Cause – Consequence*. Two constructs indicating that the technical issues are the causes

of challenging integrations, provide a broader basis for concluding that the technical challenges are the causes. Two constructs make the conclusion stronger because they have different opposite poles and still the technical pole is the cause.

There is also a correlation indicating that challenges related to resources are the cause, and the challenges related to the process are the consequences. Hence, two constructs are indicating that the consequences are the challenges related to process. Seen together with the previous correlation it is possible to conclude that the technical challenges and the challenges related to resources are the causes of challenging integrations, while the process related challenges are the consequence.

4.1.4.2.4 Summary

One of the main results from the analysis above is that the interviewee sees the challenges related to either the resources or the technical aspect as the causes, while the challenges related to the process is the consequence. In other words, if the challenges related to resources or the technical aspects are solved, the process would also be less challenging.

The technical challenges are perceived as both hard to handle and critical. The functional challenges are, on the other hand, easy to handle, but not critical.

As seen in Figure 4-10 the interviewee does not see any of the critical challenges as easy to handle. The critical challenge that is least difficult to handle is lack of competence. This challenge is also a cause, and is therefore a possible starting point for an improvement process.

Figure 4-10 Relationships between supplied constructs for QA manager B

4.1.4.3 QA manager C

4.1.4.3.1 Elements

The grid has four clusters with two elements in each cluster in addition to a single element with low correlation to the other elements. The cluster with the highest similarity score is *Knowledge about the system one is going to integrate with* and *Developers' knowledge about the system to integrate with (technical/functional)*. This pair of elements has the same rating on all constructs, except one where they are not rated at all, and the similarity score is 88.9%. They are quite easy to handle, critical and they are both causes. The challenges are related to knowledge, but are rather technical activities than management activities. Both challenges are necessary to overcome in order to achieve a good integration and they are related to the interaction between systems.

Similar to the previous cluster, the next cluster, *Scope of the integration* and *Who has the best premises to make the requirements for the integration*, is also necessary to overcome in order to achieve a good integration and is about the interaction between systems. The elements are also critical causes that are related to knowledge. The challenges are rather management activities than technical activities.

The third cluster consists of *Protectionism* and *Understanding for each other's purpose of the integration*, and is related to personal qualities. The challenges are seen as hard to handle and placed in the middle of the criticality scale. On the constructs where knowledge is one of the poles, these elements are placed close to this pole.

Focus from the management and *Coordination between teams in R&D* forms the fourth cluster. These challenges are seen as management activities and are easy to handle. On the criticality scale they are closer to the critical pole, and they are related to work allocation and time.

The element *Different development cycles and different versions* does not relate particularly well to the other elements and have a similarity score of 38.9%. The fact that it is a technical challenge seems to be the factor that is most distinguishing; the other elements are either related to knowledge or not rated on this particular construct. The challenge is also the only challenge that is not critical.

4.1.4.3.2 Cluster constructs

The constructs *Necessary to achieve a good integration – Not necessary to achieve a good integration* and *Critical – Not critical* have the highest similarity score, 94.4%. It is reasonable to assume that they have similar meaning to the interviewee. The challenges that are necessary to overcome in order to achieve a good integration are the ones that are critical to overcome, and vice versa. Together with *Cause – Consequence* and *Interaction between systems – Only one system* they form a cluster. This clustering might indicate that the challenges related to the interaction between systems are critical to overcome in order to succeed with the integration and they are the causes of challenging integrations. The correlation in the cluster does not apply to the opposite poles due to some of the elements not being rated on the constructs.

Figure 4-11 Cluster diagram for QA manager C

The constructs *Knowledge – Work allocation and time* and *Knowledge – Technical* have a high similarity. This similarity is probably due to the left poles having the same meaning. As for the right poles there is no significant correlation because some elements are not rated on the constructs. This pair of constructs is joined with the previous cluster forming a larger cluster at the score of 72.2%. A result of this clustering is a correlation between knowledge and cause, indicating that lack of knowledge is a cause to why integrations are challenging. *Lack of knowledge* is also a critical challenge.

Time – Personal qualities and *Easy to handle – Hard to handle* make a pair. The correlation between these constructs might indicate that the challenges that are related to the aspect of time, for instance coordination in time, are easier to handle than the challenges related to personal qualities. By personal qualities the interviewee meant the people's ability to understand what other people want to achieve with the integration. Hence, the interviewee perceives people's attitudes are more difficult to change than the challenges related to the more technical aspect.

The construct *Management activity – Technical activity* have no significant relation with the other constructs.

4.1.4.3.3 Matching score

4.1.4.3.3.1 Critical – Not critical

The interviewee has rated only one challenge as not critical. Therefore, all the correlations are stronger for the critical pole, than for the not critical pole.

The critical challenges are necessary to achieve a good integration, while the not critical challenges are not necessary to achieve a good integration. The reason for this correlation is probably that they have a similar meaning.

Further, the critical challenges are causes, but because the interviewee has not rated any of the challenges as consequences, it is not possible to draw any conclusions about the correlation between *Not critical* and *Consequence*.

Between the supplied construct *Critical – Not critical* and *Interaction between systems – Only one system* there is a correlation with a similarity score of 75%. The critical challenges are related to interaction between systems, but it is not possible to draw any conclusions based on the correlation between the other poles.

According to the QA manager, the critical challenges are related to knowledge, and the technical challenges are not critical. This would indicate that the interviewee perceives it as most important to overcome the challenges related to lack of knowledge.

4.1.4.3.3.2 *Easy to handle – Hard to handle*

The correlation between *Easy to handle – Hard to handle* and *Work allocation and time – Knowledge* gives a similarity score of 69.4%. This might indicate that the interviewee sees the challenges associated with the aspect of time as easier to handle than the challenges related to acquiring knowledge about the system they are integrating with. However, some of the challenges related to knowledge are easy to handle, while all the elements associated with work allocation and time are easy to handle.

Time – Personal qualities has a similarity score of 63.9% with the supplied construct. This would indicate that the interviewee finds it difficult to handle the challenges that relate to personal qualities.

4.1.4.3.3.3 *Cause – Consequence*

The interviewee has not rated any of the challenges as consequences. All the correlations are therefore only valid for the left poles.

The challenges that are necessary to overcome in order to achieve a good integration are the causes of challenging integrations.

Additionally, the challenges that are related to the interaction between systems are the causes. This relation might indicate that the interviewee thinks that the systems are not made for integration, for instance that they do not provide an interface for integration, and therefore the interaction is the cause of challenging integrations.

The challenges related to knowledge are causes, indicated by two constructs where one of the poles is *Knowledge*. These two correlations give a strong indication that the interviewee perceives lack of knowledge as a cause of why integrations are challenging.

Figure 4-12 Relationships between supplied constructs for QA manager C

4.1.4.3.4 Summary

From the analysis above it is evident that the interviewee perceive lack of knowledge as the main challenge. Lack of knowledge is both the most critical challenge and the reason why integrations are challenging. Some of the challenges related to knowledge are easy to handle, while others are hard to handle.

The most critical challenge that is easy to handle, is *Scope of the integration*. This challenge is related to deciding what information and functionality that should be included in the integration. As seen in Figure 4-12 the challenges related to lack of knowledge is also critical and quite easy to handle. It is therefore advisable to prioritize these challenges in an improvement process.

4.1.4.4 Summary QA

Two of the QA managers have identified the causes as more critical than the consequences. One QA manager sees the causes are harder to handle than the consequences. One says that technical challenges are causes, while another identifies lack of knowledge as a cause.

Frequent constructs among the QA managers are technology and organizational aspects, such as resources and communication.

The QA managers are not as united as the developers. QA managers A and B seem to be quite united, but QA manager C has quite different perceptions. The two first QA managers find the technological challenges as the most important challenges and the causes why integrations are challenging. The third QA manager, on the other hand, does not emphasize these challenges and perceive challenges related to knowledge as more important.

4.1.5 Product management

4.1.5.1 Product manager A

4.1.5.1.1 Elements

In the grid, three clusters can be identified. The cluster with the highest similarity score is *Functional requirements*, *Too early releases*, and *Release cycle*. They are all critical and quite easy to handle. The three challenges are related to the development process and expectation in the market, and are also related to a specific project rather than being project independent.

The next cluster contains five elements, *Organization*, *Who decides*, *Competence technology*, *Competence business understanding* and *Right competence, few people*. They are seen as quite easy to handle, but still quite critical. On the construct *System – Organization* the elements are more related to the organization. The challenges are mostly about execution of integrations.

Poor technical documentation does not have a strong correlation to any of the other elements, and have a similarity score of 53.1% with the other elements.

4.1.5.1.2 Cluster constructs

The two most similar constructs, receiving a similarity score of 91.7%, are *Expectation in the market – Personnel administration* and *Project related – Project independent*. This indicates that the challenges related to personnel administration, are independent of the projects. On the other hand, the challenges the interviewee relate to the expectation in the market, are related to a specific project.

Figure 4-13 Cluster diagram for Product manager A

Added to these constructs are *Development process – Personnel policy*, *Expectation in the market – Execution*, *Critical – Not critical*, and *System – Organization*. Together they form a cluster with a similarity score of about 80%. The challenges that are related to the system and to a specific project are more critical than the challenges related to the organization and personnel.

Hard to handle – Easy to handle and *Cause – Consequence* shows no significant correlation with the other constructs.

4.1.5.1.3 Matching scores

4.1.5.1.3.1 *Critical – Not critical*

The construct with the highest similarity score with *Critical – Not critical*, is *Expectation in the market – Personnel administration*, with a score of 83.3%. This suggests that the product manager finds the challenges related to the market as more critical than the personnel administration internal in the company. This might be a consequence of his responsibility for looking after the customer's needs.

According to the project manager the project related challenges are critical, while the challenges independent of a specific project are not critical.

The correlation between *Critical – Not critical* and *Development process – Personnel policy* indicates that the challenges related to the development process are critical, while the personnel policy is not critical. It is reasonable to assume that the development process is related to specific projects while the personnel policy is more an organizational challenge, hence project independent. This correlation will support the previous correlation which indicates that the project related challenges are critical while the challenges independent of projects are less critical.

4.1.5.1.3.2 *Easy to handle – Hard to handle*

The execution is easier to handle than expectation in the market, which is hard to handle. This correlation does not apply to the challenges *Too early releases* and *Functional requirements*. The interviewee's definition of execution is challenges that are internal in the company. Contrary, expectation in the market is external. This suggests that the product manager sees the internal challenges as easier to handle than the external challenges.

The product manager also perceives the development process as easier to handle than the personnel policy. This perception would indicate that the interviewee sees it as more difficult to administrate the human resources than to handle the challenges related to how the work is carried out in the integration projects.

Further, the organization is easier to handle than the system. This correlation does not apply to the challenge *Functional requirements*.

4.1.5.1.3.3 Cause - Consequence

There are no strong correlations between the construct *Cause – Consequence* and the other constructs. The strongest is with the construct *Hard to handle – Easy to handle*, which has a similarity score of 61%. There is also a weak correlation indicating that the causes are related to the system, while the consequences are related to the organization.

4.1.5.1.4 Summary

The interviewee finds the challenges related to the expectation in the market as critical, which reflects the interviewee's responsibility for communication with the customers. The challenges that are directly related to the integration projects are also seen as critical, while the challenges related to the personnel are not critical.

Matters concerning the execution of the integration projects and the organization are easy to handle, perhaps because the product manager feels that he has more influence on these issues.

The challenge that is perceived as most important to overcome is the prioritization of what functionality to include in the integration. This challenge is illustrated by the black point in the lower left part of Figure 4-14. It is advisable to emphasize this challenge in an improvement process.

Figure 4-14 Relationships between supplied constructs for Product manager A

4.1.5.2 Product manager B

4.1.5.2.1 Elements

As we can see in Figure 4-15 there are no obvious clusters in this grid. The elements *Reflecting the business logic* and *Synchronization* have identical rating on all constructs, and therefore receive a similarity score of 100%. Together with *Internal integration, assets* and *Security problems when integrating with 3.party* they form a cluster. All four elements are critical causes. On the construct *Easy to handle – Hard to handle* they are placed in the middle of the scale, and they are all technical challenges.

Another cluster is *Heavy implementation of standard*, which is related to installing the integration, and *Not working*, which is challenges they do not understand why occur. The challenges are quite easy to handle and can be fixed during the project, and do not necessarily have to be done right from the beginning. On the three constructs where one of the poles is *Technical*, they are all closer to the technical pole than to *Planning*, *Organization* and *Cosmetically*.

With a similarity score of 68.8% to the previous clusters is the element *Common user interface*. Like the previous cluster the challenge is technical rather than related to organization and planning. It is also a cause, but contrary to the previous challenges it should be done right from the beginning.

The last two elements form a cluster connecting with the other clusters at 65.6%. These challenges are causes that can be affected, and are quite critical. They should be done right from the beginning, and are also more related to organization and planning than to technical.

Figure 4-15 Cluster diagram for Product manager B

4.1.5.2.2 Cluster constructs

The dendrogram in Figure 4-15 shows that all constructs except one are clustered together in one larger cluster. There are two pairs of constructs having the highest similarity score (77.8%), namely *Technical – Organization* and *Technical – Planning*, and *Cause – Consequence* and *We can affect – We cannot affect*. The first pair probably receive the high score because both left poles are *Technical*, and for the right pole, planning is more related to the organizational aspect than to the technical. As for the other pair, *Cause – Consequence* and *We can affect – We cannot affect*, the correlation only applies to the left poles, which would indicate that the interviewee sees the causes of challenging integrations as something they can affect.

The technical challenges are seen as more critical and something they can affect, a correlation that is supported by three constructs. These challenges are possible to fix during the project, contrary to the organizational challenges which should be done right from the beginning.

The supplied construct *Easy to handle – Hard to handle* have the least similarity with the other constructs.

4.1.5.2.3 Matching score

4.1.5.2.3.1 Critical – Not critical

Three constructs indicate that the technical challenges are more critical than the other challenges. Still, the non-technical challenges are not non-critical, they are just less critical than the technical challenges.

The correlation with *Something we can affect – Something we cannot affect* indicates that the critical challenges are something they can affect. However, all challenges can to some extent be affected.

4.1.5.2.3.2 Easy to handle – Hard to handle

Between the supplied construct *Easy to handle – Hard to handle* and *Technical – Organization* there is a correlation of 66.7%. This correlation indicates that the interviewee see the technical challenges as easier to handle than the organizational challenges. Handling the technical challenges is probably not the product manager's responsibility and the product manager might see them as easy to handle because he underestimates their complexity.

The challenges they can affect are easier to handle than the challenges they cannot affect, according to the interviewee. The challenges that are easy to handle could be easy to handle simply because they can affect them. The one exception is *Organizational resistance*, which is a challenge they can affect, but is hard to handle.

4.1.5.2.3.3 Cause – Consequence

There is only one challenge rated as a consequence. The poles correlating with *Consequence* will therefore not be considered in the following analysis. The only

consequence is *Not working*, which is defined as problems that occur that they do not understand.

According to the product manager the causes are something they can affect. This is fortunate as the causes should be improved in order to improve the consequences.

The correlation between *Cause – Consequence* and *Critical – Not critical* indicates that the causes are critical. The causes might be seen as critical because in order to solve the consequences the causes have to be solved first.

4.1.5.2.4 Summary

According to the interviewee the technical challenges are the most important issues to solve. Some of the technical challenges the interviewee finds it pressing to solve are challenges related to the synchronization of data and problems they encounter when integrating with 3rd parties. Fortunately, the technical challenges are quite easy to handle, according to the product manager.

The organizational challenges are perceived as less critical than the technical challenges. Examples of organizational challenges are issues related to organizational resistance, i.e. people do not want to adapt their systems for the integration. Though these challenges are not seen as non-critical, they are still less critical than the technical challenges. Contrary to the technical challenges, the organizational challenges are quite hard to handle.

The interviewee has presented several possible starting points for an improvement process, indicated by the challenges located at the lower left part of Figure 4-16. The challenges related to reflecting the business logic of the systems in the integration, synchronization of data, as well as the challenges they experience with the internal integrations are all critical and neither hard nor easy to handle.

Translating the market needs into a technical specification is a challenge that is easy to handle, though not as critical as the previous described.

Figure 4-16 Relationships between supplied constructs for Product manager B

4.1.5.3 Product manager C

4.1.5.3.1 Elements

There are two main clusters of elements in Figure 4-17. The smaller one is *Exchange* and *Web service*. These elements are concrete technical challenges and have the same rating on all constructs except two constructs where they are not rated. Both elements are critical and hard to handle. They are related to the market and are technical challenges. Both challenges are causes. The similarity score with the other cluster is remarkable low, at 27.5%.

The larger cluster contains the rest of the elements, namely *Relations*, *Communication*, *Same direction*, *Information*, *Mutual understanding of useful functionality/needs*, and *Time*. All challenges are more related to human resources than to the technical aspect. They are also easier to handle than the other challenges, and are more about common platform in the company than the market.

4.1.5.3.2 Cluster constructs

The dendrogram in Figure 4-17 shows a cluster of five constructs having approximately the same similarity score. These constructs are *Market – Common platform*, *Technical – Prioritization*, *Hard to handle – Easy to handle*, *Technical – Human resources* and *Cause – Consequence*. The cluster shows that the interviewee sees the technical challenges as hard to handle and as the causes of challenging integrations. As for the more organizational aspects, i.e. human resources and prioritization, the interviewee perceives these as the consequences and also easier to handle.

Figure 4-17 Cluster diagram for Product manager C

Together with the previous cluster, the constructs *Critical – Not critical* and *Technological platform – Market communication internal/external* form the larger cluster in the dendrogram. This clustering shows that the technical challenges are more critical than the other challenges, according to this product manager.

The constructs *Active action – Passive action*, *Necessary tools – Collaboration*, and *Information internal/external – Respect for each other* have less relation to the other constructs.

4.1.5.3.3 Matching score

4.1.5.3.3.1 *Critical – Not critical*

For the construct *Critical – Not critical* none of the elements are rated as not critical. Therefore, in the following correlations the emphasis will be on correlations with *Critical*, and elements that are not critical will be denoted *Less critical*.

There are three constructs where one of the poles is related to technology. In two of the constructs, the technical pole is the most critical, while human resources and market communication internal/external are less critical. As for the last construct, both the technical pole and prioritization of what to include in the integration, are critical. From this correlation we might conclude that the interviewee sees prioritization and the technical challenges as more critical than challenges related to human resources and market communication.

The less critical challenges are seen as easy to handle, while the critical challenges are seen as hard to handle, which is indicated by a similarity score of 68.8%. This correlation does not apply to the challenge *Mutual understanding of useful functionality/needs*.

There is a correlation for the left poles between *Critical – Not critical* and *Market – Common platform*, with a similarity score of 75%. *Market* is defined as Company X's ability to "hit" the market, and these are external challenges. The correlation between *Critical – Not critical* and *Market – Common platform* would indicate that the interviewee sees the external challenges related to the market as critical. This perception might be due to the interviewee's position as a product manager with responsibility for the communication with the customers.

4.1.5.3.3.2 *Easy to handle – Hard to handle*

All three constructs with technical as one of the poles indicate that the technical challenges are hard to handle. The human resources as well as prioritization are seen as easy to handle.

The similarity score of 87.5% between *Easy to handle – Hard to handle* and *Common platform – Market* suggests that the challenges related to common platform in the company are easy to handle, while the market is hard to handle. Since the common platform is internal in Company X, and the market is external, it is reasonable to draw the conclusion that the internal challenges are easier to handle than the external, according to the product manager.

4.1.5.3.3 Cause – Consequence

All the technical challenges are regarded as causes, while market communication and human resources are perceived as consequences.

Further, the causes are hard to handle, and the consequences are easy to handle. This correlation does not apply to the challenge *Time*, by which the interviewee meant that there is always lack of time and one should have respect for each other's time. *Time* is a cause which is easy to handle.

The causes are critical, while the consequences are not critical. This correlation is stronger for the left poles. All the causes are critical, but among the consequences there are both critical and less critical challenges. The causes might be seen as critical because the interviewee thinks it is more effective to solve the causes than to improve the consequences.

4.1.5.3.4 Summary

The main results in the analysis are that the technical challenges are hard to handle and are seen as the reasons why integrations are challenging. Two of the technical challenges mentioned by the product manager are concrete challenges, but they are hard to handle. On the other hand, the challenge regarding mutual understanding of what functionality is needed in the integration, is easier to handle and still critical. As we can see in Figure 4-18 the challenge is also a cause, and therefore, this is an advisable starting point for an improvement process.

The organizational challenges are consequences and are easier to handle. They should not be prioritized in an improvement process.

Figure 4-18 Relationships between supplied constructs for Product manager C

4.1.5.4 Summary product managers

The product managers agree on several issues, but their perceptions are not united in all cases. One of the main points they agree on is the challenges related to the functionality in the integration. Still, they have diverging approaches to the problem. One of the product managers finds the prioritization of what functionality to include in the integration as challenging, while another perceive it as challenging to convert the market needs into a technical specification. Yet, the third product manager is more focused on the importance of a mutual understanding of each other's needs regarding functionality in the integration.

All product managers have included both technical and organizational aspects. However, they disagree on their criticality and tractability.

The product managers agree that the market is hard to handle and critical, which probably reflects their position as product managers with responsibility for the contact with the market.

4.2 Content analysis

Moving on from the individual analysis, we will now present the analysis on an aggregated level, using content analysis.

As described in section 2.3.4, content analysis is a way of categorizing the elements and constructs.

The categories can either be a standard category system or be derived from the constructs or elements, as a grounded theory approach [8]. In the first alternative one might use category systems from a theory or former studies. In the latter alternative, the categories emerge by looking systematically at the elements or constructs and identifying similarities. As we did not find any standard category system appropriate for our study, we chose the latter approach.

4.2.1 Elements

The content analysis of elements was carried out in an exploratory manner following guidelines in [8].

The element cue cards were spread out on a table. If two elements seemed to be related, they were grouped, thus forming an initial category. Other elements were either added to this category, or they formed a new category. This aggregation process continued until all element cards were allocated to a category.

To ensure reliability this process was also performed independently by another researcher. Reliability in this case concerns which categories that arise and then how elements are allocated to categories. Prior to computing agreement scores, we consulted each other's categories and decided which ones corresponded semantically to the other's categories. A two dimensional matrix with our respective categories was made. All elements were placed in the cell corresponding to, respectively, the first and the second researcher. Elements located on the diagonal had been placed in the same category by the two researchers.

As expected, the agreement score was low on the first pass: 51.25% when considering both corresponding and non-corresponding categories and 74.5% when considering only corresponding categories.

Then, the non-corresponding categories were negotiated upon until a full category set, with common names, was obtained. Afterwards, all elements were recategorized independently using only the agreed categories. Based on the final categorizations, a new reliability matrix and agreement scores was computed. Jankowicz [8] suggest to aim at 90% agreement. Our reliability matrix gave an agreement score of 92.5%, which is acceptable. The remaining discrepancies were discussed and resolved.

Table 4-1 shows the resulting categories with descriptions and an example of each category. The distribution of elements to the categories, both overall and by organizational tier, are given in Figure 4-19.

	Category	Description	Example
1	Unstable/new technology	When new technology is implemented in an integration the new technology is sometimes experienced as unstable and might cause problems	<i>Web service</i>
2	Spurious errors	Errors that are not caused in their systems or errors that occur without a reasonable explanation	<i>Errors in external components</i>
3	Integration with 3 rd party products	Integration with products developed outside the company where the company does not have control over changes made to the 3 rd party product and they do not have influence to make changes.	<i>Security problems when integrating with 3rd party</i>
4	Coordination of data	Synchronization of data and how to map data concepts	<i>Synchronization</i>
5	Coordination of functionality	Reflecting the business logic and ensure that the functionality of the systems are mapped	<i>Semantical differences in applications</i>
6	Coordination of platform	Challenges due to the technologies the systems are built upon	<i>Differences in architecture</i>
7	Coordination in time	Dependencies between the systems causes release cycles and deliveries to be coordinated for the integration to work	<i>Dependencies of versions</i>
8	Difficulties with on-site installation of integration	Challenges that arises when installing the integration at the customers'	<i>United installation of integrated systems</i>
9	Difficult/lack of will to reuse code	Reuse of code from other integrations, both the technical obstacles and the practitioners' interest in reusing	<i>Low amount of recycling</i>
10	Lack of communication on technical issues	Communication between the teams from the systems involved in the integration about technical issues that is relevant for the integration	<i>Missing flow of information</i>
11	Lack of documentation	The quality and quantum of the technical and functional documentation of the system they are integrating with	<i>Poor technical documentation</i>
12	Difficult communication between stakeholders regarding integration requirements	There are different views on what is important to include in the integration and the integration should reflect the market's needs	<i>Understanding of each other's purpose of the integration</i>
13	Lack of ownership of integration tasks	The company is organized according to their products and in an integration it is not always clear which team is responsible for the integration tasks	<i>Owner of the integration</i>
14	Allocating/requiring the right personnel/knowledge/skills	The practitioners of the integration have to acquire knowledge about the system they are integrating with. People with essential knowledge about the systems are often not able to work on the integration	<i>Challenge to find people with knowledge about the target-system</i>
15	Interpersonal issues	Issues related to relations between people	<i>Relations</i>
16	Too product focused/resistance to integration	The teams working on the individual systems are not interested in making changes to their own system in order to facilitate the integration.	<i>Protectionism</i>
17	Management lack of priority	How the management prioritizes the integration project with regards to resources and time	<i>Resources (shared, timebox)</i>
18	Other	Challenges that did not belong in any of the other categories	<i>Waiting for technology</i>

Table 4-1 Elements categories with descriptions and examples

Figure 4-19 Distribution of elements to categories

The categories to which most elements were allocated, were (12) *Difficult communication between stakeholders regarding integration requirements* and (14) *Allocating/acquiring the right personnel/knowledge/skills*. In the latter category, the three organizational tiers are represented evenly (via their respective elements). This would indicate that this category summarizes challenges that concern all three tiers, and that there is agreement that allocating the appropriate human resources is indeed a challenge.

In contrast, the former category, (12), only contains elements put forth by QA managers and product managers. Though there are elements from only two tiers, for both tiers the elements are put forth by several interviewees. This indicates that it is consensus within these tiers that communication regarding requirements is challenging.

4.2.1.1 Developers

The categories that contain most elements put forth by developers are the technically focused (5) *Coordination of functionality* and (6) *Coordination of platform*. In the former category all developers have placed at least one of their elements, which would indicate that coordination of functionality is a challenge the developers agree is challenging.

The developers have not placed any elements in the categories (12) *Difficult communication between stakeholders regarding integration requirements*, (13) *Lack of ownership of integration tasks*, (16) *Too product focused/resistance to integration*, and (17) *Management lack of priority*. These four categories are all popular amongst the QA managers and product managers. The absence of developer's elements indicates that the

more organizational challenges, such as communication and delegation of responsibility, are not the main priority of the developers.

4.2.1.2 QA managers

QA managers place most challenges in (13) *Lack of ownership of integration tasks* and (14) *Difficulties in allocating/acquiring the right personnel/knowledge/skills*, and seem to be less focused on technical challenges and more concerned about allocating human resources and responsibilities.

(8) *Difficulties with on-site installation of integration* and (11) *Lack of documentation* is not the main concern of the QA managers and none of them have put their elements in these categories (which do contain elements from both developers and product managers). The categories are more technically, which seem to be less important to the QA managers.

4.2.1.3 Product managers

The product managers are the tier that spread their elements over the most categories. Every category, except two, contains at least one element put forth by a product manager. They place most elements in (16) *Too product focused/Resistance to integration* and (12) *Difficult communication between stakeholders regarding integration requirements*, which indicates that product management is concerned with challenges that are related to human and organizational issues. In the latter category, all three product managers are represented with at least one element, which would indicate that there is a consensus that communication associated to requirements is indeed a challenge.

There are only two categories the product managers have not placed any elements in, namely (9) *Difficult/lack of will to reuse code* and (13) *Lack of ownership of integration tasks*. Both these categories seem to be “tied” to the other tiers; the former category consists of elements from only developers while the latter category is dominated by QA managers.

4.2.2 Constructs

The content analysis of the constructs was carried out in a similar manner as for the elements. The agreement score was 82% on the first pass, and 87.5% on the second pass. However, three categories were abstracted into a single category, which gave a final agreement score of 96.4%. The remaining discrepancies were discussed and resolved. The resulting categories with descriptions together with an example of each category are given in Table 4-2. Figure 4-20 shows the distribution of constructs to categories.

	Category	Description	Example
1	Need resources to handle – Need only to agree what to do or to allocate resources	Balancing issues they have to use resources to handle and issues that only need to be agreed on	<i>Communication – Execution</i>
2	Ability to do integration – Will to do integration	What qualifications a system has to integrate with another system, contrary to how willing the developers are to accommodate the integration	<i>Ability to integrate – Will to integrate</i>
3	Challenges associated to a specific integration project – General challenges	Whether the challenges are related to a project or is project independent	<i>Project related – Project independent</i>
4	Challenges that demand involvement – Challenges that do not demand action	The degree of involvement required of the practitioner	<i>Active action – Passive action</i>
5	Challenges within Company X – Challenges that involve 3 rd parties, including customers	Whether the challenges are internal in the company or involving external parties	<i>Internal – External</i>
6	Organizational challenges – Technical challenges	Organizational challenges are related to management activities and process, while technical challenges are directly related to the technology in the integration	<i>Responsibility – Technology</i>
7	Challenges due to multiple integrations – Challenges when integrating with one system	Whether the challenges are due to integration with multiple systems or would be challenging when integrating with one system as well	<i>Consequence of more than one system – Independent of number of systems</i>
8	Integration implementation and technology – Integration requirements, design, and interface	Balancing the technological challenges and the more planning-related phases	<i>Ability at the receiving system – Interaction between systems</i>
9	Challenges associated to development – Challenges associated to quality assurance	Balancing the development phases and the testing phases	<i>Test – Development</i>
10	Challenges associated to development process – Challenges associated to human resources	Challenges related to the development itself, and the allocation and treatment of the human resources	<i>Development process – Personnel policy</i>
11	Critical challenges – Non-critical challenges	The criticality of overcoming the challenges in order to succeed with the integration	<i>Can be fixed during the project – Should be done right from the beginning</i>
12	Challenges that are possible to solve – Challenges that are outside handling	Their abilities to affect the challenges	<i>We can affect – We cannot affect</i>
13	Other	Constructs that did not belong in the other categories	<i>Information spread – Information gaining</i>

Table 4-2 Constructs categories with descriptions and examples

Figure 4-20 Distribution of constructs to categories

The categories that contain most constructs are (6) *Organizational challenges – Technical challenges* and (8) *Integration implementation and technology – Integration requirements, design, and interfaces*. In the former category, all three organizational tiers are represented, but with more constructs from product managers and QA managers. This would indicate that there is a consensus across the tiers that the balance between the organizational and technical challenges is important. The latter category (8) is dominated by developers' constructs.

4.2.2.1 Developers

The developers spread their constructs over the most categories. This tier also placed the most constructs in one category, i.e. (8) *Integration implementation and technology – Integration requirements, design, and interfaces*, which may indicate a more or less joint focus on the dilemmas of prioritizing the technical parts of integration engineering in contrast to the more planning-related phases. All developers have placed several of their constructs in this category.

The developers are not represented in the categories (2) *Ability to do integration – Will to do integration* and (3) *Challenges associated to a specific integration project – General challenges*. However, these categories contain only one construct each, and it is therefore not reasonable to draw any conclusions from this.

4.2.2.2 QA managers

The QA managers are represented in fewest categories, although their constructs are evenly distributed in the categories in which they are represented. Most constructs from QA managers are placed in the categories (1) *Need resources to handle – Need (only) to agree what to do or to allocate resources* and (6) *Organizational challenges – Technical challenges*, which may indicate a main concern among QA managers toward tradeoffs

between human resources and organization on the one hand, and tasks to be done and technology on the other hand.

The QA managers are not represented in four of the categories. However, these categories contain either constructs from only one tier or one construct from each of the other tiers. It is therefore not possible to draw any conclusions about QA managers based on the categories they have not placed constructs in.

4.2.2.3 Product managers

Product managers put most of their constructs in (5) *Challenges within Company X – Challenges that involve 3rd parties, including customers* and (6) *Organizational challenges – Technical challenges*. This may indicate a view among product managers that integration challenges primarily concern balancing internal procedures with external customer-driven demands, as well as with balancing organizational and technical issues. All product managers are represented in both categories, which indicate a consensus within the tier that these are important issues.

The categories the product managers are not represented in are (2) *Ability to do integration – Will to do integration*, (7) *Challenges due to multiple integrations – Challenges when integrating with one system*, and (9) *Challenges associated to development – Challenges associated to quality assurance*. Since these categories contain constructs from only one or two interviewees, the product manager's absence from these categories should not be emphasized.

4.2.3 Constructs – Honey's method

Following Honey's method [8, 39] outlined in section 2.3.4.5.1, we computed the similarity scores of each construct against each of the three supplied constructs: *Cause – Consequence*, *Critical – Not critical*, *Easy to handle – Hard to handle*. If an elicited construct receives a high similarity score with one of the supplied constructs, this means that the elements that were ranked on the elicited construct were ranked similarly on the supplied construct, and that the respective poles of the two constructs describe each other.

In order to summarize these relationships, one can do the same for construct categories. Thus, based on the similarity scores for individual constructs, average scores for each of the 13 construct categories were computed, see Figure 4-21, Figure 4-22 and Figure 4-23.

Jankowicz suggests to “identify personally salient constructs on which there is a consensus in the [category]” [8] and to look at the H-I-L indices for the selected constructs. However, the method described in [8] is intended for larger samples of constructs, and in this study we therefore looked at the category as a whole.

The correspondence with the supplied constructs is summarized below. Note that categories that contain few constructs or constructs from only one interviewee (e.g., categories 4, 9 and 12) are not commented upon since they provide a weak basis for generalization. These categories are indicated by a lighter color tone in the diagram. Of

the remaining categories, we analyzed those where the majority of the constructs had high similarity scores.

4.2.3.1 Critical – Not critical

In category (5) *Challenges within Company X – Challenges that involve 3rd parties, including customers* five of the six constructs allocated to this category have a high similarity score with the supplied construct *Critical – Not critical*. Half of the constructs indicate that the internal challenges are the most critical, while the other half indicates that the external challenges are the most critical. In the latter case, all the constructs are from product managers, while in the former case there is one construct from each organizational tier. From this we can conclude that there is a consensus between the developers and the QA managers that the internal challenges are most critical, while the product managers do not have a united perception on this issue.

In the category (6) *Organizational challenges – Technical challenges* most constructs indicate that the organizational challenges are the most critical. The constructs have mixed similarity scores, with an emphasis on intermediate scores. All organizational tiers are represented evenly, which indicates that the majority see the organizational challenges are more critical than the technical challenges.

The product managers have the highest similarity score to the supplied construct *Critical – Not critical* within category 6, which might indicate that the product managers perceive the organizational challenges as critical, and the technical challenges as not critical.

Within the category (1) *Need resources to handle – Need (only) to agree what to do or to allocate resources*, the constructs from QA managers are the ones with the highest similarity score to the supplied construct *Critical – Not critical*. This would indicate that the QA managers perceive the challenges that they need resources to handle as more critical than the challenges where they only need to agree upon an issue.

Figure 4-21 Average percentage similarity with supplied construct *Critical – Not critical*

4.2.3.2 *Easy to handle – Hard to handle*

In (5) *Challenges within Company X – Challenges that involve 3rd parties, including customers*, half of the constructs indicates that the internal challenges are easier to handle than the external challenges. These constructs are put forth by product managers. In contrast, the other half of the constructs in this category indicates that the external challenges are easier to handle than the internal challenges. These constructs constitute one from each tier. This indicates that there is not a consensus on whether the external or internal challenges are easiest to handle, but that most of the product managers see the internal challenges as easier to handle. Also, there is a consensus between the developers and the QA managers that the external challenges are easier to handle.

Similarly to the previous category, there seems to be differing opinions about the organizational and technical challenges. Half of the constructs in the category (6) *Organizational challenges – Technical challenges* indicates that the organizational challenges are easier to handle than the technical challenges. These constructs are put forth by product managers and QA managers. The other half indicates the opposite, i.e. that the technical challenges are easier to handle than the organizational challenges. In this latter group, all three tiers are represented. However, the constructs from QA managers and product managers are put forth by only one person from the each tier. In the former group, each tier is represented by at least two persons, and therefore there is a stronger correlation. It is therefore reasonable to assume that most product managers and QA managers perceive the organizational challenges as easier to handle than technical challenges.

There is a correlation indicating that the critical challenges are easier to handle than the non-critical challenges. All tiers agree on this correlation and they all have high similarity scores.

Figure 4-22 Average percentage similarity with supplied construct *Easy to handle – Hard to handle*

Figure 4-23 Average percentage with supplied construct *Cause - Consequence*

4.2.3.3 *Cause - Consequence*

Though the correlation between the construct *Cause - Consequence* and *Easy to handle - Hard to handle* is not very strong, most constructs with intermediate and high similarity scores indicate that the causes are easier to handle than the consequences.

Overall, *Cause - Consequence* is the category with the least similarity with the constructs in the categories.

In the category (6) *Organizational challenges - Technical challenges* there seems to be an agreement among the QA managers and the product managers that the technical challenges are the causes. Two developers agree on the opposite, that the organizational challenges are the causes. Still, the former correlation is stronger than the latter.

In (8) *Integration implementation and technology - Integration requirements, design, and interfaces* most constructs indicate that the challenges related to *Integration requirements, design, and interfaces* are the causes. The developers and one of the QA managers have placed constructs in this category supporting this conclusion.

4.3 Terminological interference

Terminological interference was investigated because conflicts in terminology might be a reason why people experience challenges in the first place. If there are differences in the practitioner's terminology, the communication about the integration will be more challenging. Such differences are often difficult to unveil, but are important to identify before improvement actions are initiated.

		Terminology	
		Same	Different
Concepts	Different	Consensus Terminology and concepts are used in the same way	Correspondence Different terminology are used for the same concepts
	Same	Conflict Same terminology are used for different concepts	Contrast Different terminology are used for different concepts

Figure 4-24 Relationships between terminology and concepts (from Baddoo and Hall)

elements, the relationships could be identified by looking at the element's rating on the constructs. For instance, an element that is rated as technical by one interviewee and organizational by another, would probably be an example of the interviewees using the same terminology for different concepts, hence a conflict.

If two elements with different terminology have identical rating in two grids, they are most likely an example of correspondence. The interviewees are probably thinking of the same concept, but use different terminology for the concept.

4.3.1 Conflicts

In this study we chose to elicit the elements from the interviewees. As expected, this resulted in a diversity of elements. It is therefore not straight-forward to compare the elements and identify conflicts in terminology. Still, some of the interviewees did use the same terms and these cases were further investigated to identify possible conflicts. Elements with the same terminology were identified and their description and ratings were compared.

The results for the condition conflict are:

- *Time*: this element is elicited from a QA manager and a product manager. They have slightly different descriptions of the element, but there is no doubt that there is consensus between the terminology and concept.
- *Synchronization*: this element is elicited from a developer and a product manager. Their description and rating on the constructs are concurrent, and this is certainly a consensus.
- *Dependencies of versions*: this element is elicited from a developer and a product manager. Their description and rating on the constructs are concurrent, and this is certainly a consensus.
- *Low amount of recycling*: this element is elicited from two developers. Their description of the element is concurrent, though their ratings of the element on the constructs are diverging. One of the developers has rated it as related to information, while the other developer rated the element as a technological issue.

Still, it is likely that there is consensus about the concept, but that the differences in the rating are due to their explanation of why the element is a challenge.

- *Documentation*: this element is elicited from two developers and one product manager. Their description and rating on the constructs are concurrent, and this is certainly a consensus.

The analysis of the elicited elements with equal terms indicated that there are not terminological conflicts in the elements from the interviews.

4.3.2 Correspondences

Another form of difference in terminology, namely what Badoo and Hall call correspondence, could be identified by looking at the description of the elements. Elements that have similar description, but different terms are possible correspondences. This analysis was carried out by searching for concurrent descriptions and comparing the term used for the element.

This investigation resulted in the following:

- *Silo and Protectionism*: both elements are elicited from QA managers and have similar description. They are about protecting their own system, resistance to changes to their system, and not helping out in the integration.
- *Semantics*: challenges related to different semantics in the systems, are elicited from several interviewees. Terms like *Meaning*, *Different datamodels – different semantics*, *DB (datamodel)*, *Semantical differences in application* and *Handling different properties when integrating with more than one system* are all about the semantics. Still, the interviewees used different terminology for this concept, which may be a sign of correspondence.
- *Requirements*: different terms are used for requirements. *Requirements*, *Scope of the integration*, *Functional requirements* and *Requirements specification* are elements that are elicited from the interviewees. They all have similar descriptions and might be cases of correspondences.

As we see in the results of the analysis, there are a few cases where different terms have been used for the same concepts. Though there are not many occurrences, they could cause misunderstandings.

4.4 Feedback from interviewees

At the end of each interview, the interviewee was asked to give an immediate assessment of whether he or she had been given sufficient opportunity to present his or her perceptions. The majority felt that their perceptions had been covered, while a minority stated that they felt that the elements they came up with in the beginning of the interview acted as limitations for the rest of the interview.

A summary of the analysis of simple relationships was fed back to the interviewees after the analysis was carried out. As we recall from section 2.1, feeding the results back to the

company is an important factor in lean research. As well as giving them an initial insight into the results, this also gave them the opportunity to comment on the analysis.

The summary included a short introduction to the analysis methods used to arrive at the results along with the summary of the analysis of constructs and elements. A diagram showing the relationship between the supplied constructs was also attached. This was intended to point out possible starting points for an improvement process.

Having read the analysis, the interviewees were asked to answer the following questions:

1. Do you feel that the analysis agrees with your perceptions of challenges in integrations?
2. Are there connections described in the analysis that you were not aware of earlier?
3. Are there viewpoints that are not pointed out in the analysis?
4. Other comments?

Seven of the nine interviewees responded to the feedback. They all felt that the analysis covered their perceptions of the challenges in integrations. It was also salient that they thought the interviews made their viewpoint more explicit, and felt it was valuable to have their viewpoints written down. This would be helpful when talking to others working on other integrations.

However, most interviewees did not find the analysis surprising or felt that it had uncovered unknown relationships between the challenges.

4.5 Summary

In this chapter we have presented detailed analyses of the grids. The analyses have explored what the actual challenges are and connections between the challenges. In the content analysis, both the elements and constructs were analyzed on an aggregated basis. By comparing the elements and their description, terminological differences were uncovered. The chapter concluded by presenting the feedback from the interviewees. The results of the analyses will form the basis for the discussion of the research questions and main hypothesis in the next chapter.

5 Discussion

This chapter will address and discuss the research questions outlined in section 2.1 based on the analysis in chapter 4. Each of the research questions will be addressed before the main hypothesis is discussed. In section 5.2 an ontology for challenges in integrations will be presented.

5.1 Addressing the research questions

RQ 1: What are the challenges with integrations according to the developers, quality assurance managers and product managers, and which challenges should be prioritized?

During the interviews, several challenges in integration projects emerged and were made explicit through the elements. The content analysis of the elements, as well as the summary of the individual analyses, will be used as the basis for addressing this question.

The developers perceive technical challenges related to coordination as the main challenges with integrations, especially coordination of functionality and technological platform. The challenges associated with coordination of the functionality are for instance related to semantics, i.e. that the concepts have different meanings in the systems, or overlapping functionality. The coordination of the technological platform is about the technologies the systems are built upon, and how to make these work together.

Challenges related to information and knowledge about the system they are integrating with, are also important to the developers. This is both about documentation and people's knowledge, for instance locating people that have knowledge about the system and including them in the integration tasks.

The QA managers, on the other hand, are more concerned about challenges related to the human resources and delegation of the responsibility of integration tasks. The challenges associated with human resources are about the allocation of the resources. Some of the QA managers find the technological challenges important as well.

Last, the primarily concerns among the product managers are related to human and organizational issues. They perceive communication about requirements as a central challenge, as well as people's resistance to integration. The communication about the

requirements are challenging because people from the integrating systems have different backgrounds and viewpoints on what is important functionality. It is also seen as difficult to translate the market needs into a technical specification.

In the individual analyses, possible starting points for an improvement process were pointed out. The criteria were that the challenges are critical and easy to handle. The challenges that are easy to handle should be prioritized because progress is an important motivational factor [34]. Further, the causes should be prioritized because it is generally more effective to attack the cause than to improve the consequence. The following summary is based on the prioritized challenges from the individual analysis.

There are three main areas identified as possible starting points for an improvement process. These are the most salient challenges and all tiers are represented in at least one of the areas:

- Responsibility
- Requirements
- Knowledge

The first main area is related to the placement of responsibility for the integration. As we recall from the introductory chapters, the responsibility for the integration is often placed with one of the systems, namely the system initiating the integration. The challenges experienced with responsibilities relates both to the development of the integration and responsibilities for maintenance tasks. For instance, it might be problematic to locate the cause of an error in case of fault reports from customers with integrated systems. None of the systems want to take responsibility for the fault, and usually it is the developers of the integration who have to solve the fault, even if the error is in the other system.

Second, several of the interviewees pointed out requirements for the integration and what functionality to include as a challenging activity. Though they have different perceptions of what the exact problem is, they agree that the specification of requirements is challenging.

Last, there is consensus that knowledge is a challenge. The challenges related to knowledge are both about access to documentation as well as people's knowledge about the systems. Documentation seems to be either not easily accessible or not good enough. Regarding people's knowledge, some of the interviewees state that it is difficult to find people that have knowledge about both systems, which would be desirable in an integration project. Others do not perceive it as difficult to find people to talk to, but rather to allocate people with knowledge to work on the integration.

RQ 2: Do the challenges focus on technical aspects or process-related activities?

First, the distinction between a technical challenge and a process-related challenge should be settled. *Process* is a general concept, and in this study we have used a rather

rough division between *Technology* and *Process*. The challenges directly related to the technology of the systems, for instance the platform and synchronization of data, are the technical challenges. On the other hand, the process-related challenges are associated with the “softer” part of the development, i.e. human aspects, and are not directly related to the technologies.

Secondly, the conclusion is based on the content analysis of the elements. As only a minority of the interviewees provided a construct *Technical – Process*, the individual constructs will not be used as a basis for categorizing the challenges. The categories from the content analysis of the elements were separated into two piles, technical and process related. The categories 1 – 9 (see Table 4-1) belong to the technical pile, while the categories 10 – 17 are process related. The challenges in the category *Other* are not included because they are difficult to categorize.

Overall, there is approximately the same number of technical challenges as challenges related to process, respectively 44% and 56% of the total number of challenges. This suggests that both aspects are perceived as challenging.

More interesting is the distribution of technical and process related challenges on the tiers. As expected, the developers provided more technical challenges than process-related challenges. About 70% of the developer’s challenges are related to technology.

The majority of the challenges provided by QA managers are related to process. 73% of their challenges are related to process, while the remaining are technological challenges.

The most surprising result is perhaps the distribution of the challenges provided by product managers. About 36% of their challenges are technical, while 64% are process related. As they have a high-level responsibility for the products and integrations, one might expect their challenges to be more related to process than the other tiers’ challenges. The inclusion of the technical challenges could be due to some of the product managers’ education and experience, discussed in more detail under RQ 6.

We can conclude that the developers focus on the technical challenges, the QA managers on the challenges related to process, while the product managers focus on both aspects, but with emphasis on the process related challenges. Overall, there is focus on both the technical and process related challenges.

RQ 3: Are there differences between the organizational tiers regarding opinions on what the challenges with integrations are?

As expected, there are different foci on what the main challenges with integration projects are. In the discussion of the previous research questions it is evident that the organizational tiers have diverging perceptions of the challenges.

The most salient difference between the tiers is the topic addressed in the previous research question, i.e. whether the focus is on the technical or process-related challenges.

The developers' focus on the technical challenges is as anticipated. The QA managers and product managers are more united in their emphasis on process related challenges.

In the content analysis of the elements we see that there are no categories in which only developers and QA managers have placed their challenges (when excluding categories where all tiers are represented). The product managers, on the other hand, have several categories in common with the other tiers, which suggests that the developers and the QA managers have the most diverging perceptions.

The results of our study regarding this issue are to some extent concurrent with the findings in [1]. Though Baddoo and Hall's classification of organizational tiers is not identical with the tiers in this study, some similarities can be identified. Their project management tier can correspond to our tier of QA managers, while senior managers in their study correspond to product managers in our study. Baddoo and Hall found that the two manager tiers have the greatest similarity, which supports our finding that QA managers and product managers are most united. Still, Baddoo and Hall report that developers and senior managers have the most diverging perceptions, while in our study, the QA managers and developers seem to have least in common. This difference might be due to the technical experience of the majority of the product managers in our study. In Baddoo and Hall's study, it was the project managers that had technical experience.

We can therefore conclude that there are indeed differences between the organizational tiers, and that the QA managers and the product managers have the highest level of agreement, while the developers and the QA managers differ most in their perceptions.

RQ 4: Are there agreements or disagreements internally on each of the tiers about what the challenges with integrations are?

The figure shows the agreement between the nine interviewees by counting the number of elements placed in the same categories (based on the content analysis of the elements). The highest agreement between two persons is underlined.

		Developers			QA			Prod. M.		
		A	B	C	A	B	C	A	B	C
Dev.	A	X	3	3	1	3	2	3	3	1
	B		X	<u>6</u>	2	3	1	2	<u>4</u>	0
	C			X	1	<u>4</u>	2	3	3	0
QA	A				X	2	3	3	<u>4</u>	3
	B					X	2	2	2	1
	C						X	<u>6</u>	2	<u>4</u>
Prod. M.	A							X	1	2
	B								X	3
	C									X

Table 5-1 Agreement between the interviewees

The developers have the highest agreement internally on their tier. The QA managers are not as united as the developers, but still they agree more than the product managers. As seen in the figure, all developers agree most with a person on their own tier, while all QA managers and product managers agree most with a person on another tier. This finding supports the result from the previous section. A conclusion to this question is therefore that the developers have the highest level of agreement internally on the tier, while the QA managers and product managers agree more

with the other tiers than internally on the tier.

RQ 5: Are there differences in *how* the interviewees perceive the challenges, and are these differences most evident between the tiers or internal?

As seen in the content analysis of the constructs there are indeed differences both between the tiers and internally on the tiers regarding how the interviewees perceive the challenges.

The developers perceive the dilemmas of prioritization of the technical aspect of the integration in contrast to the more planning-related phases as the most important issue regarding the challenges. They are also united in the perception that design and interface are the cause of challenging integrations. The consensus among the developers regarding these issues separates them from the other tiers.

There are some disagreements among the developers as well. Issues they disagree on are whether the technical challenges are easy or hard to handle and whether the cause of challenging integrations are related to the organization or technology.

The tier the developers agree most with, regarding how they perceive the challenges, is that of QA managers. They agree that the internal challenges are most critical, while the external challenges are less critical.

QA managers agree internally about a main concern regarding the tradeoffs between human resources and organization on the one hand, and tasks to be done and technology on the other. There is also a consensus that the technical challenges are the causes.

The differences among the QA managers are related to whether the technical or organizational challenges are easiest to handle, though most QA managers perceive the organizational challenges as easiest to handle. Overall, the QA managers have a higher level of agreement with the developers than with the product managers.

The product managers have a united perception that integration challenges primarily concern balancing internal and external challenges, as well as dilemmas between organizational and technical challenges. There is also a consensus that the technical challenges are the causes.

However, they disagree on the issue regarding whether the internal or external challenges are most critical, as well as whether the external challenges are easy to handle or not. Furthermore, they have diverging perceptions on whether the technology or organization is easiest to handle, though most product managers think the organizational are easiest to handle.

The tier which differs most from the other tiers is the product managers, while the developers and QA have more similar perceptions. This result is contradictory to the result from the similar analysis of the elements. In the analysis of the agreement between the tiers about *what* they think of the challenges, the QA managers and product managers

had the highest level of agreement. Regarding *how* they perceive the challenges, the developers and QA managers seem to agree more with each other than with the product managers.

Internally, the developers are the most united tier, while the QA managers are more united about *how* they perceive the challenges than *what* they perceive are the challenges. The product managers seem to differ in their perceptions in both cases.

RQ 6: Are there any connections between the practitioners' experience and education, and what they regard as the challenges in the integrations?

Table 5-2 gives an overview of the education and experience of the interviewees. As we can see, the QA managers and product managers in this study have similar educations and experiences, and they also present a lot of the same challenges. They are the two tiers that have most in common regarding what they perceive as the challenges in integrations. Still, the QA managers have less technical experience, which is evident in their emphasis on the process-related challenges. The technical experience of two of the product managers is not shown in the elements, as only one of them has put forth several technical challenges.

The developers have only technical education and experience, and they also emphasize the technical challenges. Developer A is the developer with most process related challenges, and he is also the developer with the least experience. The majority of the other developer's elements are technical challenges.

The QA managers have presented mostly process-related challenges. However, one of the QA managers has provided several technical challenges, and he is the only QA with technical experience, though he has no technical education. On the other hand, the only QA with technical education has only one technical challenge. This might indicate that for these QA managers the experience is more influential than the education.

For the majority of the interviewees, the education and experience they bring to their roles seem to have an impact on their perceptions of challenges. It also seems like experience is a more influential factor than education for the interviewees in this study.

Name	Education	Experience (prior to their current position)
Developer A	IT	Technical
Developer B	IT	Technical
Developer C	IT	Technical
QA manager A	Economy	Sales and economy
QA manager B	Economy	Development and implementation
QA manager C	Economy and IT	Implementation of products at customer
Product manager A	Economy	Development and QA
Product manager B	High school	Technical and sales
Product manager C	High school	Sales

Table 5-2 Interviewees' experience and education

RQ 7: Are there any conflicts in terminology between the tiers?

The results from the terminological analysis in section 4.3 indicate that there are minor terminological differences between the interviewees, and that the differences are not necessarily between the tiers. Regarding the condition *conflict*, i.e. the interviewees use the same terminology for different concepts, the analysis indicated that such differences does not exist. In cases where two or more interviewees have used the same term for an element, the descriptions are also concurrent.

As for the condition *correspondence*, that the interviewees use different terminology for the same concepts, the analysis pointed out *Silo* and *Protectionism* as two terms for the same concepts. Both elements are elicited from QA managers, thus the difference is not due to differences between the tiers.

The analysis also identified several terms for the challenges related to requirements, with concurrent descriptions. All the terms that were used are well known terms for this concept, but still such differences could cause misunderstandings. The differences were identified both between the tiers and internal on a tier.

Further, different terms for challenges related to semantics were also identified. The elements were elicited from interviewees from all tiers, and this may be a source of terminological interference.

Some of the correspondences are between the tiers, but there also seem to be correspondences in terminology internally on the tiers. We can therefore conclude that there are some terminological differences between the interviewees, both internally on the tier and between the tiers.

The main hypothesis: “It is more important to emphasize the process than the technical tasks”.

As all the research questions have been addressed, they will form the basis for answering the main hypothesis.

As we recall from the first chapter, our contact in Company X presented a hypothesis saying that it is more important to emphasize the process than the technical aspects in integrations. He postulated that one should arrange for effective communication between the practitioners regarding interface, semantical differences and other issues that should be agreed upon in order to develop an integration.

Through the analyses that have been carried out on the grids, it is evident that there is a clear distinction between the technically related and the process-related challenges. All interviewees have presented at least one challenge of each kind, and the balance between the challenges related to technology and those related to process seem to interest all tiers.

There are diverging perceptions on whether the technical or the process-related challenges should be emphasized, but the majority sees the process-related challenges as

most critical. The hypothesis can therefore be said to be supported by several of the practitioners that were interviewed in this study. The question of whether the hypothesis is supported by objective measures such as long-term efficiency or reduced costs is a matter for further research.

As indicated by the results of this study, however, most developers would probably state it is most important to emphasize the technology in an integration. Still, developers also point at the process through their frustration over lack of documentation and accessible knowledge.

The importance of effective communication is supported by Hohmann who states that “communicating effectively is perhaps the single most important factor influencing success or failure” [40]. Davis agrees in [41] and one of his principles is that “communication skills are essential” [41]. Both Hohmann and Davis consider the success of the outcome of the development to be dependent of good communication. Such communication is probably even more important in integrations than in ordinary development projects because the communication is across teams and systems. One of the observations made by the contact in Company X that is presented in section 1.3 was that too much time is spent on discussing interfaces. Challenges related to communication were also one of the frequently mentioned challenges elicited from the interviewees. These findings support the acceptance of the hypothesis.

In addition to effective communication, a successful process is also related to good management. Another of Davis’ principles is “good management is more important than good technology” [41]. His opinion is that even with few resources, good management can produce good results. One of the observations made by the contact in Company X was that little emphasis is put on management in the integrations. Challenges related to vague responsibilities and allocation of resources were salient in the interviews, and there is therefore good reason to believe that management is indeed important in the integrations.

The awareness of the importance of the process, and especially effective communication, is vital when initiating improvement actions.

5.2 Ontology

In short, an ontology is a taxonomy, i.e. a system for classification. As stated by Corcho et. al. [42], there are many definitions of ontology, and the definitions are continuously evolving. The most quoted definition is proposed by Gruber who defined an ontology as “an explicit specification of a conceptualization” [43]. The specification can be of both systems’ and human’s conceptualizations. The ontology in this study is intended for the latter case and is a type 1 theory in [44], a theory for analyzing.

The ontology can be used as guidance in other studies investigating challenges in integration projects. It can be helpful in identifying the challenges, as well as what kind of challenges to look for. The ontology is based on the practitioner’s perceptions and is therefore a grounded theory [45].

Figure 5-1 Hierarchical ontology

The preparation of the ontology was inspired by a study of Pareto et.al. [46] where an ontology for the implementation of Model Driven Architecture (MDA) was established. The authors identified categories on four levels of abstraction, central categories, categories, subcategories and codes, from the data sources. In our ontology we applied the top three levels with the elements from the interviews as the data source, thus forming a hierarchical ontology as illustrated in Figure 5-1.

The categories from the content analysis of the elements were used as the subcategories. We identified similarities between the categories from the content analysis, and established the categories for the ontology. The categories from the study of Pareto et.al. [46] were used as an inspiration and frequently their categories were concurrent with our categories. In such cases the terminology from the MDA study was applied. The central categories followed the division suggested under RQ 2, respectively *Technology* and *Process*.

The resulting ontology is shown in the tables below, where Table 5-3 illustrates the categories and subcategories in the *Technology*, while Table 5-4 shows the categories and subcategories in *Process*.

Technology								
External relations	Coordination					Reuse	Error handling	
Integration with 3 rd party products	Difficulties with on-site installation of integration	Coordination of platform	Coordination of data	Coordination of functionality	Coordination in time	Difficult /lack of will to reuse code	Unstable/new technology	Spurious errors

Table 5-3 Ontology for Technology

Process							
Communication		Knowledge		Organization		Commitment	
Lack of communication on technical issues	Difficult communication between stakeholders regarding integration requirements	Lack of documentation	Allocating/ requiring the right personnel/ knowledge / skills	Lack of ownership of integration tasks	Management lack of priority	Inter-personal issues	Too product focused/ resistance to integration

Table 5-4 Ontology for Process

A description of the subcategories is given in Table 4-1, while the categories are described in the sections below.

Technology:

- *External relations*: challenges that involve external parties and technological challenges one might experience that are not due to their in-house products. In the study of Pareto et al., this category is located under the central category *Management*, which would correspond to the central category *Process* in our ontology. However, in our context, the concern was more with the technical issues resulted from the relations, and not necessarily the communication with the external parties; thus it is placed under *Technology*.
- *Coordination*: issues that have to be coordinated because two or more systems are communicating, that is, issues that would not necessarily be challenging if the systems were not integrated.
- *Reuse*: the ability of the systems and will of the practitioners to reuse code from other integrations.
- *Error handling*: how errors are handled in the systems.

Process:

- *Communication*: the exchange of information and opinions between the people that are in some way involved in the integration.
- *Knowledge*: the information that is needed in an integration and knowledge transfer.
- *Organization*: issues that are caused by how the company or project is organized.
- *Commitment*: the practitioners' attitude and personal qualities, and their accommodation of the integration.

5.3 Summary

In this chapter we have addressed each of the research questions and the main hypothesis. The most important challenges were pointed out and both similarities and differences were identified between and internally on the tiers. We concluded that the main hypothesis is supported by the majority of the interviewees. An ontology based on the challenges experienced by the interviewees was presented. The next chapter will summarize the study and its contributions, point at some critical assessment, and suggest future work.

6 Conclusion

6.1 Summary

To identify challenges in Company X, we conducted Repertory Grid interviews of three practitioners from three tiers in the organization. The interviews were conducted using the semi-structured technique Repertory Grid. Experienced challenges were elicited and rated on the interviewee's constructs and on supplied constructs.

Extensive analysis was performed on the data from the interviews, both individually and on aggregated basis. The analysis revealed perceptions regarding what the challenges in integrations are and their characteristics, as well as relationships among the challenges.

The results were fed back to the interviewees and starting points for improvement actions were suggested. This feedback gave the interviewees an initial insight into the results as well as an opportunity to comment upon the analysis. All interviewees felt the analysis had covered their perceptions of challenges.

6.2 Contributions

For company X, this study is the first step towards an understanding and overcoming of integration challenges. They are now provided with concrete formulations of the challenges experienced by the practitioners of the company's integration projects. As well as having precise formulations, the challenges are also prioritized according to three criteria that are important in improvement processes, i.e. the rating on the supplied constructs.

The interviewees were chosen from several integration projects, giving a diversity of situations in which integration challenges were experienced. Also, viewpoints from several organizational tiers were elicited, which is important to emphasize in process improvement.

Both differences and similarities in perceived challenges were uncovered between the tiers. A mutual awareness and understanding of differences in perceived challenges is crucial for efficient problem solving. On the other hand, the fact that there is consensus across all tiers on certain issues might give the company valuable information when initiating process improvement. For instance, challenges concerning the allocation of personnel and knowledge were mentioned by all three tiers.

The study has also contributed with an ontology for integration challenges. The ontology has provided the company with a structure of the challenges with three levels of abstraction, which could be useful when communicating about the challenges and when assessing improvement actions. The ontology is the first step toward a practitioner-based scientific theory. Other companies involved in software integration that see a need to systematize the company's knowledge on this topic may find the ontology useful as well.

In addition to the ontology, the research method used in this study might be used as a guideline for other companies with a need to identify challenges in integrations, and perhaps also in other development projects.

6.3 Critical assessments

The main threat to the validity of the study is that the data is collected on only nine practitioners. This might increase the risk of challenges being left out because none of the interviewees happened to have experienced them. Nine interviewees will also give a weaker basis for generalizations. Still, given the timeframe of the study, interviewing more than nine practitioners would cause the interviews to be less in-depth; hence our knowledge about the challenges might have been more superficial, had we interviewed more people.

Possible misunderstandings of the elements or constructs could act as a threat to the validity of the study. Though the majority of both the elements and constructs were thoroughly described by the interviewees during the interviews, we might have misunderstood the meaning an interviewee placed in an element or a construct. Since the content analysis is not carried out by the interviewees themselves, the categorization of the elements and constructs might be a source of uncertainty. Still, this uncertainty is reduced by two researchers performing the categorization independently.

Another threat to the validity of the study is the research method itself and whether it has achieved to capture the entirety of the practitioner's perceptions. This issue raises the question of the appropriateness of the technique. In the feedback to the interviewees this issue was addressed and the response indicated that the technique was adequate. All interviewees felt that their perceptions were presented in a satisfactory way. As we recall from section 4.4 a minority of the interviewees felt, immediately after the interview, that the elicitation of elements limited the rest of the interview. Still, none of the interviewees pointed this out in the response to the feedback.

If identifying terminological differences had been a more important focus of the study, supplied elements should have been a part of the study. The elements with identical terminology were too few to provide a solid basis for conclusions about conflicts in terminology. However, as the main purpose of this study was to identify the practitioner's perceptions of challenges, the choice of eliciting all the elements was the most appropriate.

6.4 Future work

Following this study, further discussions with the interviewees, with other representatives of the three tiers, and with senior management should be initiated. Other representatives of the tiers should be involved in order to assure that the perceptions of the interviewees represent the majority of all practitioners' perceptions. Though the interviewees are from a diversity of projects there could be important issues which are missed. During these discussions a course of action for the ongoing integration projects and for future projects should be laid down.

This study has been focusing on identifying the problems. A similar approach could be used to identify improvement actions by having a solution-directed, rather than problem-directed, focus in the interviews. The challenges have now been pointed out, and in order to suggest improvement actions the current ways of action must be further investigated.

Further elaborations of this study should also be considered, for instance by including other sources of information. The challenges identified in the interviews are subjective perceptions elicited from the practitioners themselves. Objective data sources, for instance data logs, change request logs, and project reports, could add a new perspective to the study. To achieve triangulation, observation of how the practitioners carry out their integration tasks could also be possible.

7 Appendix A

7.1 Description of elements

7.1.1 Developers

7.1.1.1 Developer A

1.1 Missing flow of information: The developers in the integration project work on their own areas and do not discuss their solutions with each other. The interviewee thinks that this might get one-tracked and that the result of the integration could have been better if there were more communication regarding the solution between the practitioners in the projects.

1.2 Meaning: The challenge of meaning is about agreeing on what things in the systems mean. It might be a challenge to find out what a concept is called and mean in the system they are integrating with.

1.3 Waiting for technology: Currently an integration bus is in development, and the interviewee thinks that a lot of challenges would have been easier if they could have used the integration bus in the integrations.

1.4 Low amount of recycling: Several integration projects require some of the same work, and this is done all over again for every integration. The developers are not interested in recycling unless there is a 100% match.

1.5 What information: The systems involved in the integration have to agree on what information to exchange. It is usually neither possible nor necessary to exchange all the information in the systems. Therefore they have to agree on what information to exchange in order to fulfill the requirements.

1.6 Errors in external components: Unexpected errors caused by external components might cause a lot of problems because the developers do not understand what has happened and what the problem is.

1.7 Lack of knowledge: When integrating with another system the project participants have to acquire knowledge about the other system. This is knowledge they will only use in the current integration. The interviewee thinks that it usually results in a lot of nagging on the people with knowledge about the system. It might also be that these people are not accessible.

1.8 Who does what: When integrating with another system the developers have to acquire knowledge about the other system. This is time consuming and developers from the other system might have done the same tasks a lot faster because they have more knowledge about the system. The interviewee says that the system he is working on is a lot smaller than the other system, and therefore easier to acquire knowledge about. It all comes down to resources and the use of these.

1.9 Documentation: The documentation of the systems is not good enough. According to the interviewee, this is due to the routines of documentation not working as they should and the documentation is hard to find when they need it.

7.1.1.2 Developer B

2.1 Missing integration module, low amount of recycling: the integration code is not optimized and when integrating with a new system they have to refactor the old code. A lot of the code is the same for all integrations.

2.2 Responsibility for the integration is with another system: When integrating with another system, the developers of the integration have to acquire knowledge about this system. The other system is passive and therefore they cannot take advantage of the functionality in this system.

2.3 Documentation, technical and usage: when integrating with another system they have to understand how this system work, both technically and usage. The documentation is not always good enough.

2.4 Integration interface: different systems offer different interfaces for integration. Therefore, the integration has to be solved in different ways.

2.5 Different technology platform: some of the systems are based on old platforms. Some of these technologies do not work well together.

2.6 Synchronization: Integration raises challenges regarding which of the systems that owns the data and what actions to take if the same data is altered in both systems at the same time.

2.7 Dependencies of versions: changes in the version of one of the systems might cause problems in the integration.

2.8 Different data models, different semantics: concepts have different meanings in the different systems.

2.9 Have to be running all the time: All the systems in the integration have to be running all the time because the data from one system might have to be used in the other system. For instance, if a new customer should be created, the next vacant identification number has to be identified, which might require access to both systems.

7.1.1.3 Developer C

3.1 Changes in target system: the systems they are integrating with are not finished and therefore changes are made that affects the integration.

3.2 Missing documentation and examples for new target system: Similar to the previous challenge, the system they are integrating with is not finished and therefore documentation for this system does not exist.

3.3 Challenges related to authentication against different ERP systems: different technologies make it difficult to find a reasonable way to authenticate in the integration.

3.4 Handling different properties when integrating with more than one system: the systems are different and the more systems that are involved the more complicated the integration becomes. Concepts in one system might not be present in the other systems or they might have different meanings.

3.5 Challenges related to similar look and feel as the target system: there is a desire that the GUI should be similar to the GUI of the ERP-system (the system they are integrating with). This could be a challenge if the systems are built on different platforms.

3.6 More testing when integrating with more than one system: because more systems are involved it will be a lot more to test. A test might pass with one of the systems, but not the other.

3.7 Problems related to handling errors in target system: error situations are handled in different ways in different systems. Not all systems notify the integrated systems when errors occur.

3.8 Challenge to find people with knowledge about the target system: one has to have knowledge about the systems one is integrating with. The ideal is a person with knowledge about both the target-system and the integrating system. It is not a problem to find people to talk to, but to allocate people to be involved in the integration project.

3.9 United installation of integrated systems: the customer wants to have the integrated system ready to be used when the ERP-system is installed.

7.1.2 QA managers

7.1.2.1 QA manager A

4.1 Architecture api: the architecture of some of the systems has an architecture that makes it easy to integrate with the system, while others' architecture makes the integration more difficult.

4.2 Resources (shared, timebox): the resources are not controlled by the project manager of the integration. Therefore the integration manager has to adjust to how the owner of the resources has planned the resources. Changes in the plan will most likely cause the resources to be taken away from the integration project.

4.3 Creative: depending on how the project is managed, it might be room for the practitioners to be creative or not. If there exists a mutual understanding of the direction of the project, it is easier to come up with creative solutions. Creative solutions will make the integration better and not just satisfy the technical requirements.

4.4 Time: In integration projects there is usually a lack of time, causing the development to end too early. The developers are allowed to only develop a minimum of the functionality. It is believed that the integration would be better with only a little more time to develop.

4.5 Owner of the integration: In order to have a successful integration, it has to be clear who the owner of the integration is.

4.6 Requirements: it is important that the requirements are correct and specified good enough. Further, they should be communicated to the right people. The practitioners need to know what and how.

4.7 DB (data model): the database and data model might be different in the integrating systems. The fields and concepts might be different; they might be complementary or not supported in both systems.

4.8 Person (chemistry): usually people communicate better with some people than with others. This might be crucial for the progress of the integration project.

4.9 Silo: People are often only interested in their own project, and are not interested in integrating their system with another system. There is no commitment from them.

7.1.2.2 QA manager B

5.1 Dependencies of versions: the systems are dependable of the version of the other system. It is often a hierarchy of versions.

5.2 Semantical differences in applications: differences between the systems in the functional part of the integration.

5.3 Requests from customer: requests from customers or support could be difficult to route correct. Who should you contact when the systems are integrated? The requests could be desires or errors (one might think that the error is located in one of the products, but actually it is located in the other system).

5.4 Delivery: the delivery should be synchronized. Because of the dependencies of versions the phases in the different projects has to be synchronized, hence the delivery has to be synchronized as well.

5.5 Communication: communication between the practitioners of the project.

5.6 Test: It is more difficult to test. Who has the responsibility to test? Something might be developed one place and is going to be used another place. It is the planning of the testing that is difficult, not the testing itself.

5.7 Competence: the competence is often located in more than one unit. This has to be coordinated. The competence might be located in another department than the responsibility for the integration.

5.8 Differences in architecture: the systems are built on different platforms which caused technical challenges.

5.9 Responsibility: the responsibility is distributed on many departments.

7.1.2.3 QA manager C

6.1 Who has the best premises to make the requirements specification for the integration: when writing the requirements specification for the integration, one needs knowledge about the systems to be integrated, both technical and high-level knowledge. It might be a challenge to find the person best suited for this job, i.e. the one who has best knowledge about the systems.

6.2 Coordination between teams in R&D: every product is organized as a team in R&D. These teams have to be coordinated with regard to time and resources.

6.3 Different development cycles and different versions: the different systems have different development cycles and different releases, which may cause problems. This is due to the teams working differently and the more technical challenges, for instance that two platforms are dependent of each other's versions.

6.4 Protectionism: the system initiating the integration has to make sure that the data that is sent to the other system is correct. The other system does not want to make changes to their own system to accommodate their systems to the integration.

6.5 Knowledge about the system one is going to integrate with: the people working on the integration have to obtain knowledge about the other system. This does not only apply to the developers, but also to the other tiers. They have to obtain more superior knowledge.

6.6 Scope of the integration: What should be covered by the integration and what should not be covered, all within the time available.

6.7 Focus from the management: if the management does not have a focus on the integration it might be difficult to allocate resources and allocate the right people to work on the integration.

6.8 Developer's knowledge about the system to integrate with (technical/functional): the developers have to have knowledge about the system they are going to integrate with, both technical and functional knowledge.

6.9 Understanding of each other's purpose of the integration: understanding between the parties involved in the integration about the other's purpose of the integration.

7.1.3 Product managers

7.1.3.1 *Product manager A*

7.1 Organization: The company is organized in a way that is optimized for delivering sales results and to deliver new versions of existing products. It is difficult to make room for innovation in a business model like this, which makes new products and new business areas difficult to introduce. This also applies to integrations.

7.2 Release cycle: their system is web-based, which imposes a requirement to look modern and be up-to-date. The system is shown to the customer frequently. On the other hand, the ERP-system they are integrating with has a release cycle of one year, whereas the web-system has to release at least every 3rd month. This is difficult to coordinate.

7.3 Competence business understanding: technologists know a lot about technology, but they do not easily understand how a customer uses the system.

7.4 Competence technology: the system they are working on should be able to work together with several ERP-systems. They have to know a lot about the systems that are going to be integrated with their system and the technologies they are built on. Therefore they need a wide range of knowledge and the organization has to be willing to invest in competence.

7.5 The right competence, few people: there are few people that have good knowledge about the core systems and therefore these people become bottlenecks. In order to continue with the integration they need the right competence and to prioritize the resources right.

7.6 Poor technical documentation: the documentation of the systems they are going to integrate with is often poor, the API is not correct and the documentation does not concur with the reality. As a consequence one has to talk to people from the other system to understand the system, which is a less practical way of working.

7.7 Too early releases: the management might take away the resources working on the integration project which makes it difficult to deliver the integration on time. They are forced to stop the development and unfinished products will give them bad will in the market.

7.8 Functional requirements: they have to prioritize what functionality to include in the integration. They try to use the functionality from the system they are integrating with, but they have to consider what is easiest to manage and what is most relevant.

7.9 Who decides: is it the developers or the market that control the development? The developers often have an idea about what they want to do with the integration, but this might not correspond to what the market wants. The development department is located one place, while there are many product managers in several countries. The different countries often have different needs and might have semantical differences.

7.1.3.2 Product manager B

8.1 Organizational resistance: people that have worked on a product for a long time might be very dedicated to the product and they do not want to make changes to the product or stop the development. These attitudes are not a good foundation for a successful integration.

8.2 Not working: the integration might not work for reasons unknown to the integration team. This might cause a lot of fumbling.

8.3 Requirements specification: the company is good at identifying market needs in an integration, but they miss when translating these needs into a technical specification.

8.4 Security problems when integrating with 3. Party: the 3.party is normally not very interested in the integration and can make changes due to security-holes without notifying the company that is integrated with them. This may cause the integration to not work, which again makes the integration unstable.

8.5 Heavy implementation of standard: the integration might be complicated to implement with the customer. The integration might look good when it is developed in Company X, but be difficult to implement when released to the customer.

8.6 Reflecting the business logic: it is often a challenge to reflect the business logic from one system in the other system. A request should give the same answer in both systems.

8.7 Synchronization: if the data is stored in more than one system, it might be a challenge to synchronize the data. The question is when to update the data and which of the systems is the master.

8.8 Common user interface: Ideally, when two systems are integrated they should be perceived as one system. This might be a challenge to achieve, especially when integrating with 3.party products.

8.9 Internal integration, assets: a lot of the functionality is common for several systems. This functionality is separated in assets to be used as building blocks. When integrating with several of these assets the communication might be a challenge. Each asset is developed by an individual team and as a product manager he does not want to communicate separately with all these team, but rather have one group to communicate with.

7.1.3.3 Product manager C

9.1 Relations: it is important to establish good relations between people. Good relations are a prerequisite for good communication. Because a lot of the communication is across different teams this is even more important, thus more challenging, in integrations projects.

9.2 Same direction: it is important to work against a common target. This might be a challenge in integration projects because there are more interests to look after and there needs to be a mutual agreement on what direction to take.

9.3 Information: it might be a challenge to get a hold of the relevant information about other systems in integration projects. For instance, information about changes in one of the systems might affect the integration. Such information is not always available or given to the right people.

9.4 Time: the resources and their time are always limited and it is important to have respect for each other's time. Everything needs to be prioritized.

9.5 Communication: communication between the teams. People in the different teams have different background and different views on what is important. When the communication between the teams is not good enough, the integration will not be good enough. The sales people are organized in another company than the developers. The sales people tell the developers the needs and wishes of the market, but the developers sometimes decide to make something else.

9.6 Web service: web service is the new technology for their integrations. This is a challenge because it is not stable yet. They need stability and to get a hold of the functionality in the other system.

9.7 Exchange: integration with external systems or components. It is difficult to integrate with Microsoft, because errors occur that they do not have control over. Therefore they cannot guarantee that the information integrated with a component from Microsoft is correct, which might cause the customers to not want to buy the product.

9.8 Mutual understanding of useful functionality/needs: there need to be an agreement on what functionality is useful in the integration. Because there are different needs in the integrating systems this agreement might be a challenge. Understand each other's need in the integration.

7.2 Description of constructs

7.2.1 Developers

7.2.1.1 Developer A

1.1 Usage of resources – Agreement: some of the challenges require resources. The reason why they are challenges is because it takes time and resources to solve them. They might be internal resources or external. Other challenges might only require discussion to be solved.

1.2 Something we can affect – Something we cannot affect: There are different levels of how easy it is to overcome a challenge. Some they can affect, while other they just have to evade. The things they can affect are often internal, while the things they cannot affect are external.

1.3 Resource related – Information related: resource related is about what people are told to do and what they really should do. Information is about the structure and maintenance of documents with information about the system and how easy accessible these documents are.

1.4 Information spread – Information gaining: information spread is about making their knowledge visible for others. Information gaining is about obtaining information for them self.

1.5 Technology – Information: technology is about the technical aspect of the integration. Information is about documentation and knowledge about the system.

1.6 Information about the system – Information in the system: information about the system is about knowledge about the system and how to integrate with it. Information in the system is about what information to exchange and what information is stored in the system.

1.7 Active action – Passive action: a passive action is something a person do not do, hence absence of doing something or not acting at all. An active action is actually performing a specific action.

7.2.1.2 Developer B

2.1 Ability at the receiving system – Interaction between systems: ability is about the receiving system's ability to integrate, a service they have to offer. The interaction is about how to make the rules for how the systems are going to collaborate.

2.2 Solution – Logical/semantics: The solution is about what technologies that has been chosen and the integration itself, while the semantical pole is about the contents in the solution and rules for how the systems are going to interact.

2.3 Responsibility – Technology: responsibility is about the responsibility for arranging for the integration. If this responsibility is located at each system, the problems with the integration will be reduced. Technology is about the technological aspect of the integration.

2.4 Knowledge about the systems – Exchange of data: Knowledge is about knowledge about how the system is built up technically and how to use the system. The exchange is about the interaction and synchronization of data, and the importance of correct data.

2.5 The integration itself – Design of the solution: The integration itself is about the exchange of data, while the design is about how the solution is built up and the technologies chosen for the integration.

2.6 The gate of the system – Knowledge about the system: the gate of the system is the integration interface of the system, what abilities the system has to integrate with other systems and the service they offer in an integration. The knowledge is about the use of the knowledge about the systems.

2.7 How the solution is technologically – Contents of the solution: The technical pole is about what technologies that is chosen for the integration and any dependencies with other technologies. The contents of the solution are about the content itself, the rules for how the interaction should work and how the solution should work.

7.2.1.3 Developer C

3.1 Understanding of the system – Technical: the understanding is about understanding how the target system works and how to use it. There is often missing information about the system. The technical pole is about the technical aspect and the coding.

3.2 Technical – Quality assurance: Technical is about the technical aspect of the integration, while the quality assurance is about testing and assurance of the quality.

3.3 Important – Not so important: the degree of how important it is to overcome the challenge in order to have a successful integration.

3.4 Technical – Organization and human resources: technical is about the technical aspect of the integration, while the other pole is about the organization and how the human resources are managed.

3.5 Consequence of more than one system – Independent of number of systems: the left pole is about whether the challenge is a consequence of the system integrating with more than one system. These are challenges that the interviewee thinks that would not be a challenge if the system would only integrate with one other system. Other challenges are independent of how many systems that are integrated (the right pole).

3.6 Access to information – Technical: Access to information is about acquiring knowledge about how things work in the other system. Technical is about the technical aspect of the integration.

3.7 User experience – Development process: the user experience is about how the customer experiences the integration and about making the integration as seamless as possible, both for the customer and for the people that is installing the integration. Development is about the things that happen in the development.

3.8 Test – Development: Test is about the challenges the developers meet when testing the integration and the systems, while the development is about the development itself (development of the integration and the systems).

7.2.2 QA Managers

7.2.2.1 QA manager A

4.1 Resources – Technical: the resources that are available determine how the technical part turns out.

4.2 Availability of resources – Usage of resources: what resources are available and how the resources available are used.

4.3 Communication – Technical: communication is about how people act and what possibilities they have of talking to each other. The technical pole is about the technical aspect of the integration.

4.4 Communication – Execution: Communication is about how people talk about things, while execution is how they do what they talk about.

4.5 Ability to integrate – Will to integrate: the ability to integrate is about what qualifications a system has to integrate with another system. Contrary, the will to integrate is about how willing the developers in the other system are to integrate their system with another, e.g. make changes to the API and so on.

4.6 Organizational – Technical: the organizational pole is about planning, how the organization is and what the organization encourages. The technical pole is about the technical aspect of the integration.

7.2.2.2 QA Manager B

5.1 Internal – External: Internal is about challenges that occur in Company X, while external challenges involve something from outside Company X, for instance the customers.

5.2 Process – Technical: process is about challenges that are not directly related to the development, but rather about ongoing activities in the company. The technical challenges are activities that are directly related to the development.

5.3 Technical – Functional: technical is about the technical aspect, while the functional is about the functionality in the integration and systems.

5.4 Resources – Process: resources are about the competence that is involved and available. The challenges related to the process are about the project management and ongoing activities.

5.5 Technical – Activity: technical challenges are related to something concrete technical, while an activity is an ongoing task that is carried out.

7.2.2.3 QA Manager C

6.1 Necessary to achieve a good integration – Not necessary to achieve a good integration: the left pole is about challenges that have to be solved in order to succeed with the integration, it is not possible to continue with the integration and succeed without solving them. While other challenges makes the integration easier if they are solved, but it is possible to succeed with the integration if they are not solved as well.

6.2 Only one system – Interaction between systems: only one system is about challenges that are not critical when there is only one system (in the integration), while the other pole is about challenges that have to be solved in order to achieve good interaction between the systems.

6.3 Knowledge – Work allocation and time: knowledge is about the knowledge that is required to develop the integration, while work allocation and time is related to the aspect of time and how to divide the work.

6.4 Knowledge – Technical: knowledge is about knowledge about the systems in the integration, while technical is about technical challenges, for instance platforms.

6.5 Personal qualities – Time: personal qualities is a person's ability to understand what the other people want to achieve with the integration. Time is about the aspect of time in the integrations.

6.6 Technical activity – Management activity: technical activity related to technical knowledge about the systems, while management activity is about allocating resources.

7.2.3 Product Managers

7.2.3.1 Product Manager A

7.1 Organization – System: organization is about how the company is organized. System is about challenges that are directly related to the systems that are integrated.

7.2 Development process – Personnel policy: development process is about the way they structure their work in the integration project. The personnel policy is about the individual competence and how to administrate the human resources, for instance keeping the developers updated on new technologies. It is also important to have something to offer to the employees.

7.3 Expectation in the market – Execution: the expectation in the market is about what the customers expect and their requests. This is external. The execution is about how the company executes their projects, i.e. internal challenges.

7.4 Expectation in the market – Personnel administration: expectation in the market is about knowing what the customers want from a product and what the product should contain. The personnel administration is about attracting the right competence and to administer the personnel in the best way.

7.5 Project related – Project independent: some of the challenges are related to a specific project, while others are not related to a project.

7.2.3.2 Product Manager B

8.1 Technical – Organizational: this constructs is about whether the challenges are directly related to the systems (technical) or the organization.

8.2 We can affect – We cannot affect: the construct is about to what extent the employees in Company X have the possibility to influence to change the challenge.

8.3 Technical – Cosmetically: the technical pole is about what is directly related to the systems, while the cosmetically is about how the user experiences the systems and the look of the systems.

8.4 Can be fixed during the project – Should be done right from the beginning: the left pole is about the challenges that could be improved during the project, while the other pole is about challenges that would cause a lot of costs if they had to be improved during the project, i.e. it is important that they are done right from the beginning.

8.5 Technical – Planning: the technical pole is about what is directly related to the systems, while the planning is about the planning of the integration.

7.2.3.3 Product Manager C

9.1 Market – Common platform: market is about meeting the market's needs and is an external challenge. Common platform is not about the technical platform, but the social platform internal in the company.

9.2 Technical – Prioritization: Technical is about the technical solutions chosen for the integration, while prioritization is about prioritizing what to include in the integration and obtain support for what one wants to include in the integration.

9.3 Technical – Human resources: Technical is about the technical solutions chosen for the integration. Human resources are about the resources that should be utilized in the integration project, the cornerstone in delivering what Company X wants to deliver.

9.4 Collaboration – Necessary tools: The left pole is about the collaboration between people in different departments of Company X, while necessary tools is something that has to be there in order to succeed with the integration (could be necessities related to humans or technologies).

9.5 Technological platform – Market communication external/internal: The platform is about what technologies the system is built upon; Company X has to choose a platform for the system. Market communication is about communicating what the market wants.

9.6 Respect for each other – Information external/internal: respect for each other's time and opinion, and not misusing each other's time. The information is about the flow of information internal in Company X and external.

9.7 Active action – Passive action: An active action is where the person chooses to involve or the action demands that the person is active. In a passive action the person does not involve actively.

8 References

1. Baddoo, N. and T. Hall, *Practitioner roles in software process improvement: an analysis using grid technique*. Software Process: Improvement and Practice, 2002. 7(1): p. 17-31.
2. Hurson, T., *An Innovator's Guide to Productive Thinking*. 2007: McGraw-Hill.
3. Shewhart, W.A., *Economic Control of Quality of Manufactured Product. 50th Anniversary Commemorative Issue*. 1980: American Society for Quality.
4. Shewhart, W.A. and W.E. Edwards, *Statistical Method from the Viewpoint of Quality Control*. 1986: Dover.
5. Deming, W.E., *Out of the Crisis*. 2000: MIT Press.
6. Basili, V.R., G. Caldiera, and D. Rombach, *The goal question metric paradigm*, in *Encyclopedia of Software Engineering*, J.J. Marciniak, Editor. 1994, Wiley & Sons, Ltd. p. 528–532.
7. Fransella, F., D. Bannister, and R. Bell, *A manual for repertory grid technique*. 2004, Chichester: John Wiley. XII, 266 s.
8. Jankowicz, D., *The easy guide to repertory grids*. 2004, Chichester, West Sussex: Wiley. XIX; 308 s.
9. Stewart, V., A. Stewart, and N. Fonda, *Business applications of repertory grid*. 1981, London: McGraw-Hill. viii, 211 s.
10. Geraci, A., *IEEE Standard Computing Dictionary—a compilation of IEEE standard computer glossaries*. 1991.
11. Gullidge, T., *What is integration?* Industrial Management & Data Systems, 2006. 106(1): p. 5-20.
12. McGrath, G.M. and E. More. *Data integration along the healthcare supply chain: the pharmaceutical extranet gateway project*. in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*. 2001.
13. McKeen, J. and H. Smith, *New development in practice II: Enterprise application integration*. The Communications of the Association for Information Systems, 2001. 8: p. 451–466.
14. Themistocleous, M., et al. *ERP problems and application integration issues: an empirical survey*. in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*. 2001.
15. Markus, M. and C. Tanis, *The enterprise system experience—from adoption to success*, in *Framing the Domains of IT Management: Projecting the Future through the Past*, R. Zmud, Editor. 2000, Pinnaflex Educational Resources, Inc. p. 173–207.
16. Hohpe, G. and B. Woolf, *Enterprise integration patterns: designing, building, and deploying messaging solutions*. 2003, Boston: Addison-Wesley. LI, 683 s.
17. Dyba, T., B.A. Kitchenham, and M. Jorgensen, *Evidence-based software engineering for practitioners*. Software, IEEE, 2005. 22(1): p. 58-65.

18. Kitchenham, B.A., T. Dyba, and M. Jorgensen, *Evidence-Based Software Engineering*, in *Proceedings of the 26th International Conference on Software Engineering*. 2004, IEEE Computer Society.
19. Houdek, F., *External experiments—a workable paradigm for collaboration between industry and academia*, in *Lecture Notes on Empirical Software Engineering*, N.J.a.A.M. Moreno, Editor. 2003, World Scientific. p. 133-166.
20. Mathiassen, L., *Reflective Systems Development*. Scandinavian Journal of Information Systems, 1998. **10**(1&2): p. 67-118.
21. Argyris, C., *Knowledge for Action*. 1993: Jossey-Bass Publishers.
22. Argyris, C. and D.A. Schön, *Organizational Learning II. Theory, Method, and Practice*. 1996: Addison-Wesley Publishing Company, Inc.
23. Gigerenzer, G., *Gut Feelings. The Intelligence of the Unconscious*. 2007: Viking, Penguin, Ltd.
24. Jarvis, P., *The Practitioner-Researcher*. 1999: Jossey-Bass Publishers.
25. Ford, D.N. and John D. Sterman, *Expert knowledge elicitation to improve formal and mental models*. System Dynamics Review, 1998. **14**(4): p. 309-340.
26. Gentner, D. and A.L. Stevens, *Mental Models*. 1983: Lawrence Erlbaum Associates, Inc.
27. Johnson-Laird, P.N., *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. 1983: Cambridge University Press.
28. Johnson-Laird, P.N., *Mental models and thought*, in *The Cambridge Handbook of Thinking and Reasoning*, K.H.a.R.J. Sternberg, Editor. 2005, Cambridge University Press. p. 185–208.
29. Niu, N. and S. Easterbrook, *So, You Think You Know Others' Goals? A Repertory Grid Study*. Software, IEEE, 2007. **24**(2): p. 53-61.
30. Hornby, A.S., S. Wehmeier, and M. Ashby, *Oxford advanced learner's dictionary of current English*. 2000, Oxford: Oxford University Press. XII, 1539 s.
31. Krippendorff, K., *Content analysis: an introduction to its methodology*. 2004, Thousand Oaks, Calif.: Sage. XXIII, 413 s.
32. Miller, G.A., *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. Psychological Review, 1956. **63**(2): p. 81–97.
33. Kearney, A.R. and S. Kaplan, *Toward a methodology for the measurement of knowledge structures of ordinary people: The Conceptual Content Cognitive Map*. Environment and Behavior, 1997. **29**(5): p. 579–617.
34. Wiegers, K.E., *Software Process Improvement: Ten Traps to Avoid*. Software Development, 1996. **4**: p. 51-58.
35. Ericsson, K.A. and H.A. Simon, *Protocol Analysis*. 1993: The MIT Press.
36. Chi, M.T.H., et al., *Eliciting self-explanations improves understanding*. Cognitive Science: A Multidisciplinary Journal, 1994. **18**(3): p. 439 - 477.
37. Hassenzahl, M. and T. Trautmann, *Analysis of web sites with the repertory grid technique*, in *CHI '01 extended abstracts on Human factors in computing systems*. 2001, ACM: Seattle, Washington.
38. Bhatia, S.K. and Q. Yao, *A new approach to knowledge acquisition by repertory grids*, in *Proceedings of the second international conference on Information and knowledge management*. 1993, ACM: Washington, D.C., United States.
39. Honey, P., *The repertory grid in action*. Industrial and Commercial Training, 1979. **11**: p. 452–459.
40. Hohmann, L., *Journey of the Software Professional: A Sociology of Software Development*. 1997, Upper Saddle River, NJ: Prentice-Hall.

41. Davis, A.M., *201 Principles of Software Development*. 1995: McGraw-Hill.
42. Corcho, O., M. Fernández-López, and A. Gómez-Pérez, *Methodologies, tools and languages for building ontologies. Where is their meeting point?* Data & Knowledge Engineering, 2003. **46**(1): p. 41-64.
43. Gruber, T.R., *A translation approach to portable ontology specifications*. Knowledge Acquisition, 1993. **5**(2): p. 199-220.
44. Gregor, S., *The nature of theory in information systems*. Management Information Systems Quarterly, 2006. **30**: p. 611–642.
45. Glaser, B.G. and A.L. Strauss, *The discovery of grounded theory*. 8 ed. 1967: Aldine Transaction.
46. Pareto, L., M. Staron, and P. Eriksson, *Ontology Guided Evolution of Complex Embedded Systems Projects in the Direction of MDA*, in *Model Driven Engineering Languages and Systems*. 2008. p. 874-888.