TOWARDS INTEGRATING AGILE DEVELOPMENT AND RISK MANAGEMENT
Jaana Nyfjord
DSV Report Series
No. 08-008
Towards integrating agile development and risk management

Jaana Nyfjord

Stockholm University
Abstract

Risk management has become recognized as a best practice in the software industry. Controlling risks improves essential software development features such as product quality, planning precision and cost-efficiency. For this reason, the inclusion of risk management in software development is an important factor to consider if one wishes to achieve project success.

Agile models claim to be risk-driven. They state that their iterative approach enables continuous attention to risks and that the risks can be reduced by practices such as continuous software integration and early testing. In reality, however, the agile development models implement few risk management practices.

The research problem addressed in this thesis is multi-faceted. The problem concerns the lack of explicit risk management practices in agile development. However, it also concerns the need to address risk management on an organization-wide basis. In addition, it concerns the conflict that emerges from trying to merge the agile process with standard industrial processes, such as risk management, without compromising agility.

The goal of this thesis is to explore integration as a solution for addressing the lack of risk management in the agile model based on empirical research. It involves (a) outlining a model integrating the agile and risk management processes on an organization-wide basis, and (b) providing a foundation for its extension.

The results show that, within the scope of this research, the proposed solution is a valid candidate for improving the agile situation. However, it is still in its infancy. It does not claim to be complete but needs to be further elaborated and complemented with details. Hence, we outline the model and provide a foundation established in empirical investigation for extending it in future research.
Acknowledgements

A former colleague of mine, Fredrik Björck, gave a very striking summary of the nature of the doctoral process. I came in as a young girl and leave as a woman, PhD, wife and mother. Thanks, Fredrik for this brilliant comment.

I would never have been able to realize this PhD endeavor without the help, support and guidance of the many people and organizations that I have met along the path. Thank you all.

Many thanks to Owen Rogers at Thoughtworks, Jas Madhur at Sierra Systems, Shawn McCann, Olivier Radar and John Vanspronsen at MDA, Anders Cöster at Assa Abloy, Mikael Lundgren and Tobias Fors at Citerus AB, Måns Sandgren at Dynabyte, Johan Hernefeldt at Lumi AB, Gustav Boström and Oliver O’Hehir at Omegapoint, Johan Löfstrand and Stefan Kervefors at SVT-I, and all the other 38 anonymous interviewees and their companies for providing the empirical foundation for this entire research. Without your collaboration, this research would never have been possible.

Ass. Prof. Mira Kajko-Mattsson, my supervisor, thank you for your mentorship. Your professionalism, genuine engagement, and excellent supervision stand in a class of its own. You are truly a bottomless source of inspiration.

Prof. Paul Johannesson, thank you for expert guidance and for all the constructive comments and honest feedback that improved this work.

I would also like to thank DSV for enabling my doctoral studies and for supporting me both administratively and financially throughout the process. Thanks to you, I am now a PhD who is privileged with many exciting experiences from both national and international research projects and from eight months of guest research at UBC in Canada in 2007.

Prof. Philippe Kruchten, thank you for inviting me to UBC. This period of guest research was truly a fruitful starting point of this thesis.

Last, but foremost, I would like to thank my family and friends. Thank you all for bearing with me and for understanding my decisions and priorities during these past five years.

Mamma och Pappa, thank you for your love and for never doubting my ability to accomplish things in life, whether small or large in scope. Pappa, tiedän että olet aina vierällämme.

This final acknowledgement is dedicated to the person who did the most outstanding job in supporting me of all. I admire you. I admire your patience. Daniel, my love, how could I ever thank you?
PART I .................................................................................................................................13

1 Introduction ...........................................................................................................................15
  1.1 Research problem ............................................................................................................16
  1.2 Research goal ..................................................................................................................16
  1.3 Research scope and limitations .......................................................................................17
  1.4 Outline of thesis ..............................................................................................................18

2 Methodological foundation and research process .................................................................20
  2.1 Methodological foundation and research process .........................................................20
      2.1.1 Observation phase ..................................................................................................22
      2.1.2 Pattern phase .........................................................................................................24
      2.1.3 Tentative hypothesis phase ....................................................................................27
      2.1.4 Theory phase ..........................................................................................................28
  2.2 Questionnaire design ......................................................................................................29
  2.3 Sampling and profiles of studied organizations ..............................................................30
  2.4 Validity ............................................................................................................................32

3 Research contribution ..........................................................................................................35
  3.1 Main research contribution .............................................................................................35
  3.2 Overview of primary publications ..................................................................................35
  3.3 Other publications ..........................................................................................................42
      3.3.1 Conference and journal papers .............................................................................42
      3.3.2 Technical reports ....................................................................................................44

4 Extended background .........................................................................................................45
  4.1 Methodology, method, process or process model? ..........................................................45
  4.2 Agile software development ............................................................................................48
      4.2.1 eXtreme Programming .........................................................................................49
      4.2.2 Scrum .....................................................................................................................50
  4.3 Software risk management ..............................................................................................51
      4.3.1 AS/NZS 4360 Risk Management Standard ..........................................................52
      4.3.2 Project Management Body of Knowledge ..............................................................53
      4.3.3 IEEE 1540 Risk Management Standard .................................................................54
      4.3.4 Software Risk Evaluation Method ........................................................................55

5 Synthesized models ..............................................................................................................57
  5.1 Creation of synthesized models .......................................................................................57
## PART II

- Appendix A: 111
- Appendix B: 113
- Appendix C: 126
- Appendix D: 129
- Appendix E: 136
- Appendix F: 138
- Appendix G: 150
- Appendix H: 157
- Appendix I: 167
- Appendix J: 179
- Appendix K: 184
- Appendix L: 189

## PART III

- Paper 1: 197
- Paper 2: 207
- Paper 3: 221
- Paper 4: 237
- Paper 5: 245
- Paper 6: 255
- Paper 7: 265
- Paper 8: 273
1 Introduction

Software development, given its diverse and abstract nature, offers unique challenges and risks (Moynihan 1997). Hence, software organizations need a process for managing the risks. Such a process can help them identify, respond to, and mitigate risks effectively.

Risk management has become recognized as a best practice in the software industry (Wiegers, 1998). Controlling risks improves essential software development features such as product quality, planning precision and cost-efficiency (Englund, 1997) (Ropponen and Lyytinen, 2000). For this reason, the inclusion of risk management in software development is an important factor to consider if one wishes to achieve project success (Kontio, 1999).

Much research has been conducted in the software risk management field in the past decades. However, relatively little research has been performed to integrate risk management with development. The spiral model, based on a risk-driven and cyclic approach, is one suggestion for making software development more effective using risk management. However, despite the fact that it was already pioneered in 1988 (Boehm, 1988), it has been only partially realized. Its cyclic character has been adapted by many current development approaches, such as iterative and agile development. Its risk-driven approach, on the other hand, has not been as influential. Still, development and risk management processes live somewhat isolated lives (Bohner and Coram, 2005) (Sliger, 2006).

Agile software development models have emerged as a means to deal with the changing business environment and growing unpredictability of software development. Their adoption in modern software organizations has increased steadily during the last decade.

Agile models claim to be risk-driven (Beck, 2004) (EPF, 2007) (Scrum, 2003). They state that their iterative approach enables continuous attention to risks and that risks can be reduced by practices such as continuous software integration and early testing (Beck, 2004). In reality, however, the agile development models implement few risk management practices (Armenta and Gaono, 2008) (Bohner and Coram, 2005) (Sliger, 2006). Hence, there is clearly a gap well worth investigating bearing in mind the fact that risk management is considered best practice in contemporary software engineering.
1.1 Research problem

Despite the fact that risk management is of crucial importance for software project success, very few models have been found that explicitly relate risk management with agile development processes. One reason is the fact that extending the agile model with additional development practices concerns a highly controversial problem in itself. It is often argued that one of the most difficult impediments to extending the agile model concerns the conflict that emerges from trying to merge the agile process with any existing standards (Boehm and Turner, 2005). The question that arises is: “How can one merge agile, lightweight processes with standard industrial processes without killing agility?”

Although controversial, the demand for such solutions in the industry can no longer be neglected. This is also evident in recent publications addressing enterprise agility (Leffingwell, 2007) (Schwaber, 2007). Many large organizations show a genuine interest in using the agile model, but due to the experienced scalability problems they are hesitant. Essentially, they are challenged by the lack of guidelines for building up the agile process according their needs, where one missing building block is risk management.

Solutions for introducing risk management in agile development have been proposed, for instance by Li et al. (2006) and Sliger (2006). Unfortunately, they are limited. For instance, they implement only a subset of risk management practices or focus only on risk management in selective phases of the overall development model. Risk management, however, is a continuous organization-wide process (SEI, 2008) (Williams et al., 1999). Hence, it needs to be addressed on an organization-wide level.

Considering the state of the art, the research problem addressed in this thesis is multi-faceted. The problem not only concerns the lack of explicit risk management practices in agile development but also the extent to which risk is managed continuously on an organization-wide basis. It also concerns the conflict that emerges from trying to merge the agile process with existing standard industrial processes without compromising agility.

1.2 Research goal

Risk management is a key project success factor. Agile development models do not encompass risk management as required by many development organizations. Hence, the main goal of this thesis is to explore integration as a solution to address the lack of risk management in the agile model based on empirical investigation. More specifically, we intend to (a) outline a model integrating the risk management and agile models on an organization-wide
basis and, (b) provide a foundation for its extension. The model is expected to:

- provide software organizations practical guidance on how to integrate the risk management and agile development models
- provide a reference model for software organizations to examine their risk management practice and see how they compare to the reference model
- provide guidance for reasoning about agility with respect to risk management

1.3 Research scope and limitations

This thesis outlines a foundation for integrating the risk management and agile processes based on empirical investigation. It consists of two parts: (a) an integration model and (b) an integrated model.

The integration model is intended to provide basic guidelines for integrating the risk management and agile processes by mapping out when, where, and by whom risk management is conducted in the agile process. Hence, it is primarily targeted to process engineers, business developers, project managers or others in need of guidelines for introducing risk management in agile development.

The integrated model is intended to be a reference model against which software organizations can compare their risk management process. It is targeted to anyone in the organization interested in comparing their risk management practice in agile development.

Overall, the proposed solution covers risk management on an organization-wide basis by describing risk management and agile development on two organizational levels, the business and engineering levels.

Software development and risk management are both large domains. Our effort is by no means comprehensive, nor exhaustive. What we present here concerns some of the most immediate problems identified in the crossing of the agile and risk management domains. Each domain specifies many more requirements on the management of risk and software development, than possibly could have been elucidated within the scope of a doctoral thesis. For this reason, this thesis is limited in several ways.

Most importantly, this thesis bases its results on only a subset of existing agile models and risk management models and standards. Consequently, other models and standards are out of the scope of our analysis. The agile models included in our research are Scrum (Schwaber and Beedle, 2001) and eXtreme Programming (Beck, 2004), whereas the risk management models and standards are the IEEE 1540 Standard for Software Lifecycle Processes - Risk Management (IEEE1540, 2001), the Software Risk Evaluation Method
(SRE) (Williams et al., 1999), the Project Management Body of Knowledge (PMBoK) (PMI, 2004), and the AS/NZS 4360 Risk Management (Standards Australia, 2004).

The proposed model addresses project-oriented risk management and deals with risks typical for software engineering, such as budget, schedule, and technical risks. At this stage, it excludes more complex risks such as security and safety risk. As a result, it is not concerned with risks and risk management practices as described in other disciplines.

In addition, although the model is intended to cover the entire agile software lifecycle process, we only focus on risk management in the pre-implementation and implementation phases in this thesis. Other lifecycle phases such as corrective maintenance present further challenges. Hence, they are out of the scope in this thesis.

Finally, the proposed model involves integration of processes on the phase level. We do not integrate the processes on an activity level. Hence, we do not describe how to conduct risk management in agile projects. As a result, the model does not provide guidelines for agile risk management. This is a different topic that is subject to future research.

1.4 Outline of thesis

The thesis consists of three major parts.

PART I presents a synthesis of the entire research and its results. It includes a presentation of the research problem, the research goal, the research method, and the research contribution. It ends with a concluding discussion and suggestions for future work. A list of references is provided at the very end. PART I is arranged in seven chapters, as follows:

Chapter 1 introduces the domain of study, research problem, research goal and the research scope and limitations.

Chapter 2 describes the research process and its methodological foundations.

Chapter 3 presents the research contribution.

Chapter 4 provides an extended background. It includes an overview of some of the basic terms and concepts used in this thesis, and of agile and risk management, respectively.

Chapter 5 describes a synthesized agile and risk management model, respectively. They constitute the basic constituents of the integrated model.
Chapter 6 presents the integration model and the integrated model.

Chapter 7 presents conclusions and suggestions for future research.

PART II is a collection of appendices.

PART III presents the eight research papers that have been included in the thesis representing the main research contribution.
2 Methodological foundation and research process

The present chapter discusses the methodological foundation of this research. It presents the research process, its phases and the methods used. The chapter is structured according to the following five parts, Methodological foundation and research process, Questionnaire design, Sampling, and Validity.

2.1 Methodological foundation and research process

To conduct scientific research requires the use of an appropriate research approach. It is ultimately determined by the research question and the research objectives (Robson, 2002). Hence, considering that we aim at addressing industrial needs by providing a basis for extending the risk management capability of the agile process using integration, we can establish that we must (a) understand the fundamentals of risk management and agile software development, and (b) study them and their execution in practical and industrial settings. This implies the use of empirical research as applied within the behavioral sciences.

There are several research approaches that could fall into this category of research, for instance design science (Hevner et al., 2004) or grounded theory (Glasser and Strauss, 1967). However, due to the nature of the research problem and goals, the epistemological status and the context of this research, we chose to proceed with an empirical-inductive approach as suggested by Trochim (2006). We elaborate on this choice below.

Empirical research can be described as a process aiming at creating a theory within a domain of study (Trochim, 2006). A theory is attained by creating and testing hypotheses with empirical data in order to generate general statements. The empirical data is drawn from observation or experience (Carver et al., 2004). The empirical research process uses either the inductive or the deductive method of reasoning depending on the nature and epistemological status of the research.

The inductive reasoning method means that one begins with observations of specific phenomena to detect patterns and regularities used to formulate
preliminary hypotheses that can be further explored in order to develop some general conclusions or theories (Trochim, 2006). On the contrary, the deductive reasoning method proceeds from general theories, principles or premises to derive specific information and conclusions to be generalized instead (Trochim, 2006).

The model proposed in this thesis has been developed according to the empirical research process following the inductive reasoning method. As depicted in Figure 2.1, this process consists of the following four phases, Observation, Pattern, Tentative hypothesis and Theory (Trochim, 2006).

In the Observation phase the emphasis is on the collection of empirical facts. When following the inductive approach this means that one examines specific data, perhaps many pieces of specific data, as a first step towards the creation of a general principle or theory within the domain of study. In the next phase, Pattern, one identifies patterns in the collected data to be further explored. It involves identifying regularities or relationships between the facts observed in the Observation phase. The Tentative Hypothesis phase involves further exploration of the identified patterns by studying and verifying them in new contexts. The aim is to specify a tentative hypothesis that allows one to examine the premises for formulating a theory or drawing general conclusions based on the findings made so far. In the final phase, called Theory, one collects new empirical data and examines whether the conclusions and theories as predicted by the tentative hypothesis can be supported in the new data. One then interprets the results to generate a theory or the statements to be generalized, or identifies ideas for new hypotheses or research if needed. Another round of the empirical process may therefore start again to further develop and improve the results of this round.

In addition to using an empirical research process based on inductive reasoning, this study uses an explorative research approach. When preparing a research project and deciding on the research approach, one must also determine the epistemological status of the area of research (Karlström, 2003). If there is a low level of previous knowledge, an exploratory study is probably the best selection. If a little more is known and a more detailed result is required, a descriptive methodology might be the better selection. If much is known and relationships are to be confirmed, then an explanatory methodology is probably the best selection.
What Karlström (2003) describes here is a research method staircase. As the status of knowledge in a research area increases, one steps up the staircase. Low levels of knowledge imply that the research community must strive to explore and describe the fundamentals of the area of study before examining the exact relationships between elements of knowledge within it. Considering the current status of knowledge about the integration of risk management and agile models, suggests that our research be initiated at the lower levels of the staircase.

From the point of view of the empirical research process, explorative research means that variables and relationships among variables describing the problem cannot be determined beforehand. Hence, the research question put forward in this thesis is not aimed at testing a hypothesis as prescribed by the explanatory research approaches using deductive reasoning (Trochim, 2006). On the other hand, on the basis of the results of exploratory or descriptive research, a hypothesis could be specified afterwards which can then be tested in another round of research. Descriptive research stresses the description of an existing variable whereby the relationships between variables are subordinate, whereas in explorative research it is even more relaxed since the researcher has only suppositions about the nature of the problem and therefore starts with identifying the variables. Both explorative and descriptive research therefore addresses understanding the variables, before they and their relationships can be explained (Karlström, 2003). In other words, exploratory research can be considered a first step in identifying and validating fundamental concepts. Hence, when initiating new rounds of the empirical process, it may use either the inductive or deductive approach depending on the status of the results from the previous round.

In the following, we describe each phase of the empirical research process as applied in this research in more detail. Our research process, its phases and their inherent steps are illustrated in Figure 2.2.

2.1.1 Observation phase

According to our empirical-inductive research approach, the Observation phase involves the collection and examination of specific empirical facts within the domain of study. Hence, we conducted a literature survey of the state of art and practice within risk management and agile development. The goal was two-fold: (a) to understand the fundamentals of each discipline, and (b) to investigate the agile process from a risk management perspective.

As listed in the box of the Observation phase in Figure 2.2, it consisted of three steps: (1) Study literature, (2) Create comparison criteria, and (3) Compare agile and risk management models.

In the first step, we studied both disciplines. To achieve both breadth and depth of the risk management discipline, we chose publications of renowned
Our research process, its phases and their inherent steps

industrial and academic institutions, including: (1) international or organizational standards, such as the AS/NZS 4360 (Standards Australia, 2004), IEEE1540 (IEEE1540, 2001) and the Project Management Body of Knowledge (PMI, 2004), (2) academic and/or industrial models, for instance proposed by Boehm (1991), Carr et al. (1993) and Charette (1989), and (3) various investigations made by individual practitioners, researchers or research groups, for instance by Boehm and Turner (2005), Demarco (2004), Englund (1997), Hulett (2001), Kontio (1999), Ropponen and Lyytinen (2000), Westfall (2001) and Wiegers (1998). We did the same with the agile discipline, where we studied well-known agile models, such as eXtreme Programming (Beck, 2004), Lean Software Development (Poppendieck, 2003), OpenUP (EPF, 2007) and Scrum (Schwaber and Beedle, 2001) and their state of practice in depth.

This literature survey helped us identify and establish fundamental aspects of the two disciplines. We used the risk management aspects as criteria for comparing the agile and risk management models in the second step, Create comparison criteria.

Finally, in the third step, Compare agile and risk management models, we analyzed the risk management and agile process models using the fundamental risk management aspects as comparison criteria. This helped us to identify gaps between the two disciplines studied which recognized future research challenges. The specific observations made regarding these gaps laid the basis not only for further exploration of patterns in the next research phase, but also for the entire thesis work.
The results of the observations made in this first research phase are presented in Paper 1 (Part III).

2.1.2 Pattern phase

The second research phase, Pattern, involves identifying regularities in the data collected and observed in the Observation phase to be further explored. Hence, we moved from theoretical to industrial studies by examining the problems identified in the literature in industrial settings. The goal was to identify patterns describing a typical manner of conducting agile development and risk management in industry.

As depicted in Figure 2.2, the research in the Pattern phase was carried out in two parallel paths, the Agile path and Risk management path. We studied the agile and risk management process models separately to establish their status in industry. It also included a minor study of the information managed in both processes. The two paths were however merged into one single path in the third research phase. Each path and the patterns established for each path are briefly described below.

**Agile path**

The Agile Path consisted of the two sub-steps, (1) Establish state of agile process practice and (2) Study management of software requirement information (see steps listed in the Agile path box in Figure 2.2). The goal was to establish patterns describing the state of agile process practice and the information needed for communicating information about software requirements and their implementation within the development cycle.

In the first step of the agile path, Establish state of agile process practice, we investigated the state of the agile process in three Canadian software organizations. We did this by comparing the industrial practice against a model that we synthesized from a set of current agile process models. Our goal was threefold: (1) to identify the state of industrial agile practice, (2) to compare it to the existing agile process models, and (3) to find out how the industry has approached both agile and heavyweight activities.

In this step, we first studied current agile process models. However, we selected XP and Scrum because they were the most widely accepted models and because they complement each other (Charette, 2001). Together they constitute a comprehensive framework covering both the engineering and management process levels.

We then elicited the development activities and put them into a synthesized agile process model. In doing this, however, we observed that some customary development activities were missing. To ensure the comprehensiveness of our synthesized model and to fulfill our third goal, we complemented it with some heavyweight software development activities taken
from the standard software process model (IEEE12207, 1998). In this way, we could enquire about their applicability in an agile context.

Our synthesized agile model is presented in Paper 2 and Paper 3 in Part III. The reason why it is presented in two papers is that we had to divide the model into a Pre-Implementation Process Model and Iteration Implementation Process Model due to magnitude of the domain studied and the space restrictions. The research was however conducted in the same way for both models.

After having created the synthesized models, we created a questionnaire for each model. Using the questionnaire, which mainly concentrated on finding out whether the companies performed the activities as defined in our synthesized model, we then interviewed the representatives in our sample companies and established their state of agile process practice. The results led to some modifications in the synthesized models, hence representing a pattern of the agile state of practice. The questionnaires can be found in Appendices A-E. The results of the step Establish state of agile process practice are presented in Paper 2 and Paper 3 in Part III.

As part of this study on the state of agile practice, we also investigated what requirements information is communicated both within lightweight and heavyweight software development in the Study management of software requirement information step. We did this by creating a template of information required for describing and managing software requirements within a development cycle and by finding out how it was implemented in one company. We call our template Software Requirements Management Template (SRMT). Our primary goal was to elicit information that is needed for communicating information about requirements and their implementation within the development cycle. However, we did not aim at distinguishing which information is used in different development approaches. Our secondary goal was to find out the state of practice within the organization studied using the SRMT template as a basis.

As a first step, we created the SRMT. Hence, we started our work by studying current literature in search of publications suggesting any templates. As a next step, we created a questionnaire. It focused on finding out the type of requirements information that was managed in industry, represented by one of the Canadian companies studied earlier. As a next step, we interviewed a representative from the participating company. The results from the interview let us establish a preliminary pattern regarding the usefulness of the SRMT template. The questionnaire can be found in Appendix F. The results of this second step are presented in Paper 7 in Part III of this thesis.

Risk management path
The Risk Management Path consisted of two steps similar to those taken in the Agile Path, (1) Establish state of risk management process practice and
(2) Study management of risk information. The goal was to establish the state of risk management process practice and to investigate the coverage and amount of risk information managed and documented in traditional and agile environments.

In the first step, we investigated the status of the risk management in 37 software organizations. We did this by comparing the industrial risk management models against a risk management process model that we synthesized from a set of current risk management process models, including the AS/NZS 4360 (Standards Australia, 2004), IEEE 1540 (IEEE1540, 2001), Project Management Body of Knowledge (PMI, 2004) and the Software Risk Evaluation method (Williams et al, 1999). It was created by studying renowned risk management process models and standards and by synthesizing these models into one common model. Our goal was twofold: (1) to find out the status of risk management process in the industry today, and (2) to evaluate standard process models against the industrial practice.

We then created a questionnaire whose questions were based on the fundamental risk management aspects used as comparison criteria in the first research phase and the synthesized risk management process model. We used students to make the interviews. At this time, the students attended an advanced software engineering course that was part of an international master program. Because we used students in our investigation, we run the risk that some answers might be misunderstood. To avoid misunderstanding, three preventive actions were taken. First, we presented our risk management model in detail to the students. Second, the goal of the interview, the questions and the questionnaire design were described and discussed in class together with the students. Detailed directives regarding the expected answers, and possible follow-up questions were also inserted into the questionnaire. Third, each interviewee was asked to provide their name and contact details to allow us to contact them with follow-up questions.

The questionnaire is presented in Appendix G. The data collected in the interviews helped us establish a pattern regarding the state of risk management practice. The synthesized risk management model and the results of this step are presented in Paper 4.

In the Study management of risk information sub-step, we investigated what information was communicated on risks in traditional and agile environments. Our goal was to find out whether these environments differed with respect to the coverage and amount of information managed and documented. We did this by conducting a study of the management of risk information within industry.

To realize this research, we began with a study of renowned risk management process models and standards, including the AS/NZS 4360, IEEE1540 and Project Management Body of Knowledge. On the basis of the studied literature, we elicited all types of information regarding risk informa-
tion and their management. We then structured it and put it into a template. It is presented in Paper 6.

Using this template, we then created a questionnaire which focused on finding out: (1) the type of risk information that was managed in the industry today, (2) whether the traditional and agile development environments differed with respect to the information managed, and finally, (3) the state of practice within the organizations studied. In this way, we were able to establish patterns regarding the coverage and amount of risk information managed and documented in agile and traditional projects. The questionnaire is presented in Appendix H. The results of this step are presented in Paper 6.

2.1.3 Tentative hypothesis phase

The Tentative hypothesis phase involves further exploration of the identified patterns by studying and verifying them in new contexts. The aim is to specify a tentative hypothesis allowing one to examine the premises for formulating a theory or drawing general conclusions based on the findings made so far. Hence, we explored the patterns established regarding the agile and risk management state of practice in the previous research phase, including the synthesized models and templates, in new industrial contexts. The results supported all the previous findings indicating a lack of risk management in agile models. This allowed us to merge the Agile and Risk management paths into one single research path and to formulate a tentative hypothesis regarding the integration of the risk management agile processes as a possible solution to address the problems identified.

As depicted in Figure 2.2, the Tentative Hypothesis phase consisted of three steps: (1) Establish state of process integration in practice, (2) Elicit process integration criteria, and (3) Outline integrated model.

In the first step, we explored the state of the integration of the risk management process and the software development process in industry. It was studied in the same 37 software organizations as previously. Our goal was threefold: (1) to find out whether and how the industry has integrated risk management with their development processes (2) to identify issues that might aid in improving the integrated process, and (3) to find out the differences of the practice of integration between agile and other development approaches.

The research was conducted in form of interviews with the help of the same students and safeguards as used before. The questionnaire covered questions about the various aspects of integration as stated in the study goals. It is described in Appendix G.

As a next step, we collected data through interviews that provided us with a status regarding how the industry, as represented by the 37 software organizations, has integrated risk management with their development processes.
In the second and third steps, we focused on formulating the tentative hypothesis that would allow us to examine the premises for drawing conclusions regarding integration as a solution to the lack of risk management in the agile model. It was first roughly elicited from the knowledge gained from studying the state of risk management and its integration with the software process in the 37 software organizations. However, its final form was elicited based on the results of a new study of risk management as executed in an agile project at a Swedish software organization where we examined various scenarios of risk management in their agile process. Using these scenarios, we first identified process integration points based on which we created a basic integration model. The integration model was then used to outline an integrated model, that is, a model integrating the risk management and agile processes. These two models comprised our proposal to address the problem of the lack of risk management identified in our research so far. The data collection was conducted in the form of in-depth interviews. The questionnaires are presented in Appendices I-K.

As a result of the work conducted in this phase, we were able to formulate a tentative hypothesis regarding integration as a solution to the problems addressed, which was then evaluated in the fourth and final research phase. The tentative hypothesis was: Our proposal for integrating the risk management and agile models is a valid solution for addressing the lack of risk management in agile development. The results of this research phase are presented in Papers 5 and 8.

2.1.4 Theory phase

The fourth and final research phase, called Theory, aims at examining the premises for formulating a theory or drawing general conclusions based on the findings made so far. This is realized by collecting new empirical data and examining whether the conclusions as predicted by the tentative hypothesis can be supported in the new data.

As depicted in Figure 2.2, this phase consisted of three steps: (1) evaluate the proposed model in industry, (2) identify improvements to the model, and (3) establish a foundation for future work.

The overall goal of this final phase was to evaluate whether our proposal including the integration model and the integrated model provided a valid solution to address the lack of risk management in agile development in industry identified in our research. To accomplish this, we evaluated its validity against three requirements covering the integration model, integrated model and the question of agility (see Section 1.2 Research goals in Chapter 1).

The evaluation was conducted in the form of in-depth interviews with ten agile practitioners representing various software organizations in Sweden. The questionnaire is presented in Appendix L.
As a result of the feedback provided during the interviews, we were able to validate the proposed model within the frame of this research and also establish a foundation for future work based on the suggestions of improvements provided by the interviewees. Hence, as a final result, we were able to generate (a) conclusions about the integration of agile and risk management that were applicable within the scope of this research, and (b) suggestions for extending the model in future research. The results of this research phase are presented in Paper 8.

2.2 Questionnaire design

We conducted twelve rounds of interviews each having its own questionnaire. The questionnaires can be found in Appendices A-L.

All of the questionnaires were semi-structured and open-ended. The purpose was to give freedom to respondents to answer in their own terms (Walker, 1985). Such type of interviewing has a positive effect in a sense that while interviewing, one may elicit more knowledge about the studied domain. Its drawback however is the fact that the interviewer must possess a good understanding of the domain studied, in order to adequately react to irrelevant answers.

It is known that the design of a survey instrument can affect the way the interviewees respond and the way interviewers get an understanding of the interviewees’ domain (Biemer and Lyberg, 2004). To optimize the interview results, we made an effort to ensure good questionnaire design. It included for instance considering the following aspects:

- **Ordered questions**: We made the effort to assure that each question transitioned smoothly from previous questions. Questionnaires that jump from one unrelated topic to another feel disjointed and are not likely to result in effective interviews. We also ordered questions with respect to their ease of understanding. We first asked easy and general questions, to be then followed by more detailed and complex ones. This allowed the interviewers to follow up on interesting leads and understand the domain from the interviewee’s perspective.

- **Used cross-checking control questions**: To establish correctness of the answers, we created several cross-checking questions in the questionnaires.

- **Provided expected answers**: Semi-structured and open-ended interviews can often lead to long discussions where it may difficult to distinguish the actual answer to a question. They may also lead to entirely irrelevant an-
swers if the interviewer does not react adequately. To prevent these situations from occurring, we inserted expected answers in the majority of the questionnaires. This helped the interviewer keep the interview aligned with its objectives and to conduct to more effective and efficient interviews.

2.3 Sampling and profiles of studied organizations

An important aspect of scientific research is sampling. To maintain the validity and the reliability one needs to consider the population and the selection of study objects (Robson, 2002).

In order to find a representative sample in this research, we used convenience sampling. Convenience sampling means that the selection of respondents from a population is based on easy availability and accessibility (Robson, 2002).

The studied population is represented by totally 47 software organizations selected based on criteria that categorized them into certain types and sizes of businesses, certain types of software projects. The interviewees representing the organizations studied, in turn, were selected based on several criteria including minimum experience of the agile and/or risk management processes and role. The minimum requirements were that they were or had been involved in at least one project using the agile and risk management processes in their organizations. They also represented different roles, which resulted in valuable data representing complementary perspectives of the development process (management, product owners, and developers). Note that the criteria were used to target the organizations and to restrict the population to be studied. It is in this population that the sample was found. Eventually, the organizations and individual interviewees were randomly selected based on convenience sampling from this population.

The profiles of the organizations and the roles of the interviewees are presented in Table 2.1. The black separators mark the organizations that participated in the different industrial studies. We conducted seven industrial studies in total. They are indicated in the left column of the table. The first three studies, conducted with three Canadian organizations involve the agile status and management of requirements and risk information. They are reported in Papers 2, 3, 6 and 7. The fourth study involving the status of the risk management process and its integration with the software development processes is reported in Paper 4 and 5. It was conducted in 37 software organizations. The study involving the creation of the integration and the integrated models was conducted in one Swedish software organization. It is reported in Paper 8. The final evaluation of the two models, also reported in Paper 8, involved
Table 2.1 Profiles of 47 organizations and roles interviewed in the industrial studies

<table>
<thead>
<tr>
<th>No of employees</th>
<th>Product/services</th>
<th>Country</th>
<th>Software process model(s)</th>
<th>Roles interviewed</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 2000</td>
<td>Systems engineering, satellite and radar systems, e-commerce applications</td>
<td>Canada</td>
<td>Waterfall, RUP and agile (Scrum, XP)</td>
<td>Project leader, Project manager</td>
</tr>
<tr>
<td>&gt; 800</td>
<td>IT consulting (goverment and public organizations)</td>
<td>Canada</td>
<td>RUP and agile</td>
<td>Process owner</td>
</tr>
<tr>
<td>800</td>
<td>IT consulting (financial industry, agile coaching)</td>
<td>Canada</td>
<td>Agile (Scrum, XP, Test Driven, Lean)</td>
<td>Agile coach</td>
</tr>
<tr>
<td>480</td>
<td>Telecommunications networks</td>
<td>China</td>
<td>-</td>
<td>Project manager</td>
</tr>
<tr>
<td>750</td>
<td>GIS, database, games, business and development applications</td>
<td>Sweden</td>
<td>MSF and iterative agile (internal)</td>
<td>Department manager</td>
</tr>
<tr>
<td>600</td>
<td>Retail, online shopping and delivery services</td>
<td>Sweden</td>
<td>Plan-driven (Integrated Product Development Model)</td>
<td>Vice president</td>
</tr>
<tr>
<td>200</td>
<td>Mobile phones</td>
<td>Finland</td>
<td>Plan-driven (internal)</td>
<td>R&amp;D engineer</td>
</tr>
<tr>
<td>170</td>
<td>IT consulting</td>
<td>Sweden</td>
<td>Plan-driven and agile (internal)</td>
<td>Senior management</td>
</tr>
<tr>
<td>160</td>
<td>IT consulting</td>
<td>Sweden</td>
<td>Agile (internal)</td>
<td>Software engineer</td>
</tr>
<tr>
<td>300</td>
<td>Insurance and banking systems</td>
<td>Germany</td>
<td>Plan-driven and iterative agile (internal)</td>
<td>Executive IT manager</td>
</tr>
<tr>
<td>300</td>
<td>IT consulting</td>
<td>Mexico</td>
<td>-</td>
<td>Project coordinator</td>
</tr>
<tr>
<td>900</td>
<td>Online media, telecommunications services, entertainment services, e-commerce applications</td>
<td>China</td>
<td>Agile (internal)</td>
<td>Project manager</td>
</tr>
<tr>
<td>400</td>
<td>Software and IT service provider</td>
<td>China</td>
<td>Plan-driven and prototypes, MSF (internal)</td>
<td>Team leader</td>
</tr>
<tr>
<td>200</td>
<td>Development software for embedded systems</td>
<td>Germany</td>
<td>Waterfall (internal)</td>
<td>IT consultant</td>
</tr>
<tr>
<td>120</td>
<td>Provider of business services with &quot;on-demand&quot; software (achatCAD plug in)</td>
<td>Austria</td>
<td>Lean and Agile (agile (Scrum))</td>
<td>Senior manager</td>
</tr>
<tr>
<td>110</td>
<td>Business intelligence, data crunching technology, mobile solutions, business app</td>
<td>Finland</td>
<td>Agile (Scrum)</td>
<td>Software engineer</td>
</tr>
<tr>
<td>800</td>
<td>Software outsourcing services, applied on development and maintainance</td>
<td>China</td>
<td>Iterative (XP)</td>
<td>Software engineer</td>
</tr>
<tr>
<td>800</td>
<td>Telecommunications services</td>
<td>Iran</td>
<td>Component-based model (internal)</td>
<td>Project coordinator</td>
</tr>
<tr>
<td>600</td>
<td>Support systems for cell industry</td>
<td>Pakistan</td>
<td>Waterfall, incremental evolution (internal)</td>
<td>Senior software engineer</td>
</tr>
<tr>
<td>500</td>
<td>E-business and system integration solutions provider</td>
<td>Pakistan</td>
<td>Waterfall and evolution (internal)</td>
<td>Technical project manager</td>
</tr>
<tr>
<td>500</td>
<td>IT solution provider</td>
<td>USA</td>
<td>Iterative (XP)</td>
<td>President</td>
</tr>
<tr>
<td>500</td>
<td>Technology and business consultancy, ERP, CRM, outsourcing, c-business</td>
<td>Pakistan</td>
<td>Iterative (CMDD)</td>
<td>Technical project lead</td>
</tr>
<tr>
<td>500</td>
<td>Business technology solutions provider</td>
<td>Thailand</td>
<td>Waterfall, incremental evolution (CMDD)</td>
<td>Project manager</td>
</tr>
<tr>
<td>400</td>
<td>Evolution and maintenance of e-business AIS</td>
<td>China</td>
<td>RUP</td>
<td>IT department manager</td>
</tr>
<tr>
<td>350</td>
<td>Consulting services, specialized products for the aerospace and defense sector</td>
<td>Spain</td>
<td>CMDD, SCRUM, OPEN, QMF, XP, Agile</td>
<td>Project manager</td>
</tr>
<tr>
<td>350</td>
<td>IT consulting, outsourcing, support and system development</td>
<td>Colombia</td>
<td>Iterative (XP)</td>
<td>Director strategic solutions</td>
</tr>
<tr>
<td>300</td>
<td>Data warehouse technology development</td>
<td>Pakistan</td>
<td>Incremental (internal)</td>
<td>Line manager</td>
</tr>
<tr>
<td>300</td>
<td>Business technology solutions provider</td>
<td>Pakistan</td>
<td>Waterfall and iterative (internal)</td>
<td>Senior process engineer</td>
</tr>
<tr>
<td>300</td>
<td>Internal software development tools for testing team</td>
<td>China</td>
<td>Agile (TDD)</td>
<td>Project manager</td>
</tr>
<tr>
<td>150</td>
<td>Business applications provider</td>
<td>China</td>
<td>Agile (XP)</td>
<td>IT department manager</td>
</tr>
<tr>
<td>150</td>
<td>3D multimedia applications</td>
<td>Sweden</td>
<td>Agile (Lan)</td>
<td>System engineer</td>
</tr>
<tr>
<td>150</td>
<td>Travel Management, eProcessment, ad. logistics and service applications</td>
<td>Sweden</td>
<td>Plan-driven and agile (internal)</td>
<td>System architect</td>
</tr>
<tr>
<td>100</td>
<td>Software systems for the healthcare sector</td>
<td>Pakistan</td>
<td>RUP</td>
<td>Senior software engineer</td>
</tr>
<tr>
<td>50</td>
<td>Spatial information processing software</td>
<td>China</td>
<td>Iterative (internal)</td>
<td>Department manager</td>
</tr>
<tr>
<td>50</td>
<td>VoIP and video telephony systems</td>
<td>Finland</td>
<td>Plan-driven (internal)</td>
<td>Senior software engineer</td>
</tr>
<tr>
<td>50</td>
<td>Oracle application solutions provider</td>
<td>Thailand</td>
<td>Waterfall (internal)</td>
<td>Project manager</td>
</tr>
<tr>
<td>50</td>
<td>IT consulting</td>
<td>Sweden</td>
<td>Iterative (XP) and agile (Scrum)</td>
<td>Agile mentor</td>
</tr>
<tr>
<td>50</td>
<td>Mobile content distribution</td>
<td>Pakistan</td>
<td>Plan-driven and prototype (internal)</td>
<td>QA employer</td>
</tr>
<tr>
<td>35</td>
<td>Web applications</td>
<td>Sweden</td>
<td>Agile (Scrum and XP)</td>
<td>Team leader, product owner</td>
</tr>
<tr>
<td>200</td>
<td>Door opening solution for security and safety</td>
<td>Sweden</td>
<td>RUP and agile (Lan, Scrum, XP)</td>
<td>Senior manager</td>
</tr>
<tr>
<td>50</td>
<td>IT consulting</td>
<td>Sweden</td>
<td>RUP and agile (Scrum)</td>
<td>Senior monitor &amp; consultant</td>
</tr>
<tr>
<td>50</td>
<td>IT consulting</td>
<td>Sweden</td>
<td>RUP and agile (Scrum, XP)</td>
<td>Consultant IT manager</td>
</tr>
<tr>
<td>50</td>
<td>Web applications</td>
<td>Sweden</td>
<td>Agile (Scrum and XP)</td>
<td>Senior monitor &amp; product owner</td>
</tr>
<tr>
<td>25</td>
<td>IT consulting, agile evaluation and coaching</td>
<td>Sweden</td>
<td>Iterative and Agile (Lan, Scrum and XP)</td>
<td>2 senior business</td>
</tr>
<tr>
<td>10</td>
<td>Night vision systems for vehicles</td>
<td>Sweden</td>
<td>Agile (Scrum and XP)</td>
<td>CEO &amp; senior engineer</td>
</tr>
</tbody>
</table>
interviews with ten agile practitioners representing six software organizations. Due to the sensitivity of the material presented herein, we do not name any of these organizations. Some of them however are major well-known multinational organizations.

Summing up, in this research we have conducted interviews with representatives of totally 47 software organizations. In some organizations we interviewed several roles, hence the total number of respondents is 55.

2.4 Validity

Validity measures the accuracy of a study and its results (Robson, 2002). However, there exists no universal definition of validity. Its meaning differs depending on the discipline, school of thought, research tradition, values and beliefs (Kuhn, 1970) (Little, 1995) (Longino, 1990).

Software engineering is multidisciplinary and hence influenced by several disciplines including logics, mathematics, engineering, social sciences, psychology and business administration. Consequently, this has some particular consequences on the concept of validity. However, Land (2006) argues that research in software engineering typically focuses on the observation of aspects involving social and psychological aspects in the real world rather than natural aspects. In this respect, the degree of validity should be measured according to the criteria that are found in the tradition of social sciences.

There are four types of validity often referred to be relevant in software engineering research: construct validity, internal validity, external validity (generalizability) and reliability (Wohlin et al., 2003). They are briefly defined below:

• **Construct validity:** ensures objectivity and that the data collected, analyzed and used reflects the phenomenon under study. It can be achieved by triangulating data. Data triangulation involves collecting data from several independent sources, for instance by interviewing more than one individual and by combining different types of data, for instance interviews and statistics (Yin, 2003).

• **Internal validity:** ensures that the conclusions of the study are unbiased and based on accurate descriptions of the data. There are several ways to increase internal validity. For instance, by letting a respondent review the transcript of the interview ensures that the responses have been understood and interpreted correctly. Using a combination of triangulation techniques also reduces the risk of biased results. It could be
achieved by combining data, method, theory and/or observer triangulation (Yin, 2003).

- **External validity**: measures the degree to which results of a study with a sample can be generalized to make statements about a much larger population outside the study. To claim external validity also requires that both construct and internal validity are achieved (Maxwell, 1992).

- **Reliability**: refers to the repeatability of the study (Robson, 2002). Anybody should be able to follow the same procedure and arrive at the same conclusions. This requires that the research method is comprehensive and documented in detail (Yin, 2003).

Below, follows an evaluation of the degree of validity that has been achieved so far in the work of this thesis. Validity is evaluated according to construct validity, internal validity external validity, and reliability.

- **Construct validity**: To ensure construct validity, data triangulation has been applied during data collection. First, we have strived for achieving construct validity by investigating data from several independent sources. In the interviews, the data is represented by the responses from 55 respondents. Second, by triangulating the data resulting from both literature studies and interviews at large contributes to overall construct validity. We have also collected different types of data, including qualitative and quantitative data. For instance, we looked for statistical figures supporting the qualitatively observed data.

- **Internal validity**: Internal validity can be claimed for several reasons. We have applied several types of triangulation, including data, method, observer and theory triangulation (Yin, 2003).

  First, the accuracy of data has been ensured by data triangulation as described under construct validity. The interviewees have also been provided the opportunity to review and comment on the transcripts of the data collected in interviews.

  Second, method triangulation is applied by the use of various data analysis methods. The qualitative approach is dominating but simple forms of measurements have also been carried out, e.g. by analyzing the number of organizations using certain risk management or development activities and by comparing these numbers to the use of these activities as prescribed by existing the risk management standards and agile models. For instance, as is described in Chapter 5, we found that in addition to the planning activities prescribed by the agile models, all the studied organizations had added several planning activities from traditional development models.
Third, observer triangulation has been applied as the research has been carried out in projects where several researchers collaborated and where constant internal peer-reviews took place during the analysis and before conclusions were been drawn to ensure internal validity.

Forth, theory triangulation has also been applied to further ensure internal validity. The analysis has included multiple theories and disciplines to compare and to ensure the internal and external consistency of the results. More specifically, this thesis involved the study of processes on both technical and organizational levels. For this reason, various theories on different organizational levels were used to analyze or explain results relating to the observations made. For instance, various project management standards such as the PMBoK (PMI, 2004) and process standards such as the CMMI (Ahern et al., 2005) were used to discuss the results of the various studies.

- **External validity:** The research is still ongoing. However, the findings found so far should provide some useful guidelines for various organizations aiming at introducing risk management and agile processes. At this stage, however, there is no formal foundation for claiming general application of the integration of the risk management and agile models.

- **Reliability:** The research and the methods used in each step of the research were documented thoroughly in published research papers and technical reports (see Chapter 3 for more details on research publications and technical reports). These include descriptions of the problem(s) addressed, purpose and goals, research method, and results.

Summing up, this thesis has strived for achieving construct validity, internal validity, external validity, and reliability as described above. The state of the validity could also be discussed by evaluating the results according to Kuhn’s (1970) five criteria for distinguishing good scientific theory from bad scientific theory. The thesis at this stage of research could be argued to address three out of five of Kuhn’s criteria including accuracy, consistency and scope. However, it still remains to evaluate the criteria of simplicity and fruitfulness, which only can be answered by further validation.
3 Research contribution

In this chapter, an overview of the research produced during the course of writing this thesis is presented. A summary of the main contribution is presented first. The publications primarily relating to the main contribution are presented next. All other publications are listed at the end of this chapter.

3.1 Main research contribution

The model integrating risk management and agile development presented in this thesis is one candidate for improving the situation concerning the lack of risk management in agile development in certain organizations and development contexts. The major distinguishing features of the model are that it (a) provides basic guidance for integrating the processes, (b) provides a reference model to compare notes, (c) integrates the agile and risk management models on several organizational levels, (d) provides guidance for reasoning about agility with respect to the need of risk management in the specific development context at hand.

The model is however still in its infancy. It does not claim to be complete. For this reason, the contribution of this research is that it provides a foundation established in empirical investigation for extending the model to be used in future efforts towards integrating agile development and risk management.

3.2 Overview of primary publications

The main contribution of this doctoral thesis is based on the results gathered from eight research papers. A brief description of each paper and their specific contributions is provided below.

**Paper 1: Commonalities in risk management and agile process models**  
*Jaana Nyfjord and Mira Kajko-Mattsson*  
In this paper, we investigate and analyze agile processes from a risk management perspective to find their commonalities. Although these processes represent different types of processes (engineering and management), we compare them due to the fact that agile models claim to be risk driven. Our goal is to establish the state of risk management in the agile model. We do it by creating a roadmap covering the fundamental aspects of risk management, which we use as our comparison criteria in this study.

This paper has two main contributions. First, we found that the agile models studied make some assertions about the risk management aspects analyzed. However, they do not provide any specific suggestions for managing risks, thus leaving many fundamental risk management aspects unattended. Second, based on the results from successive studies on the integration of the two processes, we were able to establish that our roadmap captures some of the essential risk management aspects that need be considered when implementing a risk management program.

Based on the results of this research, we also make suggestions on the risk management aspects that need be considered when adopting risk management in agile development. These suggestions are further explored in later stages of this research (see Paper 5, 6, and 8).

Although theoretical, we conclude that this paper shows evidence that there are openings for integrating risk management with the agile process. The risk management gaps found indicates a need to extend the risk management capability of the agile model.

**Paper 2: Degree of agility in the pre-implementation phase**

*Jaana Nyffjord and Mira Kajko-Mattsson*


The results of our first paper motivate the need for more research on risk management in agile development. In this study, we focus on the agile model and its status in industry. We do this by comparing the industrial practice against the Pre-Implementation Process Model, a model that we synthesize from a subset of current agile process models. It describes the phases that take place prior to implementation. Because some customary planning activities are missing in the studied agile process models, we complement the synthesized model with the activities taken from a standard heavyweight software process model. Our goal is three-fold: (1) to identify the state of industrial pre-implementation practice, (2) to compare it to the existing agile process models, and (3) to find out how the industry has approached both agile and heavyweight activities to determine the degree of agility in the phases taking place prior to implementation.
The main contribution of this work consists of an observation of the challenges of using agile processes in certain development organizations, allowing us to establish that the capability of the agile models needs to be extended to address the process needs of these organizations. One of the missing capabilities concerns risk management.

Our results reflect the state of practice within the agile pre-implementation phases of one small and two middle-sized agile software projects. We found that the organizations studied have added traditional activities in the pre-implementation phases, indicating that a substantial amount of planning is conducted upfront which is typical for heavyweight processes. This is particularly evident in the middle-sized projects. It is claimed that this planning is obligatory to carry out subsequent development effectively.

Hence, our findings show evidence that the degree of agility varies in different types of projects. Larger, critical, fixed-budget projects involving for instance the evolution and maintenance of existing business critical systems are claimed to require more traditional planning. On the other hand, the organizations however dare introduce more agility into smaller, more innovative, creative, and risky new projects.

**Paper 3: Agile implementation phase in two Canadian organizations**

*Jaana Nyfjord and Mira Kajko-Mattsson*

*In proceedings of the 19\textsuperscript{th} Australian Software Engineering Conference (ASWEC'08), Australia, March 2008, IEEE Computer Society Press.*

In this paper, we continue our exploration of the status of the agile development phases by investigating the implementation phase within two Canadian organizations. We do this by comparing the industrial agile implementation practice against the Implementation Process Model, an agile process model that is synthesized from a set of current agile process models. It is synthesized in the same way as the Pre-Implementation Process Model. The Implementation Process Model describes the phases where the actual development takes place. Our goal is twofold: (1) to identify the state of industrial implementation practice and (2) to compare it to the existing agile process models.

The main contribution of this paper is that we, once again, observe that larger projects require more guidelines for developing software than currently provided by the agile models. Hence, we establish that the capability of the agile model is an issue that needs to be addressed to meet the needs as identified in the studied organizations.

More specifically, our results show that the studied organizations have adopted most of the activities as suggested in the agile models. However, they had to make some revisions due to the fact that some of the guidelines as provided by the agile models were unclear, conflicting, ambiguous or they
were simply missing. It is claimed that Scrum and XP do not provide full coverage of all software development activities, especially not for larger projects. Hence, they had to fill the gaps with complementary activities from standard development approaches. This primarily concern the activities of integration testing, system testing, release management, reporting, documentation, task planning and guidelines regarding scalability.

**Paper 4: Software risk management: practice contra standard models**

*Jaana Nyfjord and Mira Kajko-Mattsson*


In this paper, we investigate the status of the risk management in 37 software organizations. We do this by comparing the industrial risk management models against a risk management process model that we synthesize from a subset of current risk management process models. Our goal is twofold: (1) to find out the status of risk management process in the industry today, and (2) to evaluate the standard risk management models against the industrial practice.

The main contribution of this study is that we establish the status of the risk management process. Our findings show that there are discrepancies between the industrial practice and the standard risk management models. The industry studied has not implemented all the activities as prescribed by the standard models. The standard models, on the other hand, have failed to identify some of the important risk management practices.

In addition, we explore the theoretical assumptions of the roadmap covering fundamental risk management aspects as created in our initial literature study. The findings allow us to establish that it reflects the fundamentals of risk management also in the industry.

Summing up, this paper identifies a list of issues that need to be addressed in both the industry and within the risk management discipline. For instance, our findings show that the amount of risk management depends on factors such as the project size, criticality, resources, and product status. Hence, guidelines for making appropriate process adaptations are needed. Also, several problems with risk management in the industry were detected including problems with the process and its integration with other organizational processes, support tools, organization, knowledge management, resources, standardization and terminology. The problems identified hence verify that there is a need also from an industrial point of view to address risk management in software development. As for the agile perspective, this paper also shows that the lack of risk management in agile processes as identified in theory is supported by the findings made in the industry.
Paper 5: Integrating risk management with software development: state of practice

Jaana Nyfjord and Mira Kajko-Mattsson


In this second study of the status of risk management, we explore the state of practice of integrating risk management with software development. The goals involve: (1) finding out how the industry has integrated risk management with their development processes (2) identifying issues that might aid in improving the integrated process, and (3) finding out the differences between agile and other development approaches.

The main contribution of this paper is that it merges the results of the Agile path and the Risk management path into one single research path. The results show that although risk management is important, its integration with the software development process is still in its infancy and there is a great need for process integration and process integration models within the software organizations studied.

More specifically, in this paper we study the status of the integration of the risk management and development processes in 37 software organizations. Essentially, our findings show that one needs to carefully consider the organizational levels, and their inherent development phases, activities and roles when considering process integration. We also establish that the integration of these processes is conducted on an ad hoc basis. The organizations studied have not defined any process integration model. That is, they do not have guidelines for how to integrate processes. Hence, there is a need to create a process integration model.

Our study also reveals some problems within the process integration in industry. These problems primarily concern organizational issues, people, skills, processes, tools, resources, and knowledge management. These problems constitute an important platform for analyzing and improving the current process integration practice. Besides identifying a set of integration problems, we also identify a set of critical success factors suggested for achieving maximal results from process integration.

Finally, this research shows that risk management is needed in any development model, whether traditional, agile or other. Despite the claims that risk management is inherent in the agile model and way of working, the results tell us that agile models do not provide explicit guidance on when, where or how to conduct risk management.
Paper 6: Communicating risk in agile and traditional environments
Jaana Nyfjord and Mira Kajko-Mattsson
In proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA’07), Germany, August 2007. IEEE Computer Society Press.

Structured and disciplined communication is a prerequisite for the effective management of risks. In this paper, we investigate what risk management information is communicated in traditional and agile development environments. Our goal is to find out whether there is any difference in the way risks are communicated. We do this by creating a template for managing risk information and studying the management of risk information within three Canadian software organizations.

The main contribution of this research is that we create a template covering information required for managing risk information. Our results show that there are few differences between the information as communicated in the traditional and agile environments studied. We also identify some critical pieces of information that need to be recorded to manage risks effectively.

In addition, we find support that risk management information needs to be connected to the development process. Although a byproduct of this thesis work, this has been of relevance in terms of supporting the successive research towards the proposal of an integrated model.

Paper 7: A Template for communicating information about requirements and their realization
Mira Kajko-Mattsson and Jaana Nyfjord

In this paper, we investigate what requirements information is communicated both within lightweight and heavyweight software development. We do this by creating a template of information required for describing and managing software requirements and by investigating how it is implemented in one Canadian company. We call the template Software Requirements Management Template (SRMT). The primary goal of this research is to elicit information that is needed for communicating information about requirements and their implementation within the development cycle. The secondary goal is to find out the state of practice within the organization studied using the SRMT template as a basis.

Our results show that most of the information as designated in our template is recorded by the organization studied. Based on the results of this study, we identify some main pieces of information considered to be of im-
importance to record in order to manage requirements effectively. We also find support that requirements management information need be connected with the risk management process. Although a byproduct of this thesis work, the results have been of relevance in terms of supporting the successive research towards the proposal of an integrated model

**Paper 8: Outlining a model integrating risk management and agile development**

*Jaana Nyfjord and Mira Kajko-Mattsson*

In proceedings of the 34th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA’08), Italy, September 2008. IEEE Computer Society Press.

In this final paper, we create and evaluate an integration solution addressing the problems identified in previous research. It consists of an integration model and an integrated model. By examining various scenarios of risk management in an agile process within one software organization we identify integration points between the two processes. Using these, we create an integration model consisting of four basic integration points mapping out when, where and by whom risk is managed in the agile process. The integration model is then used as a basis for outlining an integrated model, that is, a model integrating risk management and agile development. It is demonstrated in a practical example represented by our case company.

Essentially, in this paper, we examine the premises for drawing the conclusions as predicted by our tentative hypothesis regarding the integration of the agile and risk management processes. More specifically, we investigate whether integration is a valid solution for addressing the lack of risk management in agile development by creating and evaluating the integration and integrated models.

The validity of the proposed integration solution is evaluated against three requirements covering the integration model, integrated model and the aspect of maintaining agility. The evaluation is based on interviews conducted with ten agile practitioners in the Swedish software industry.

Generally, our evaluation results show that the three requirements are fulfilled. Hence, we conclude that integrating risk management and agile development provides a valid solution to address the lack of risk management in the agile model. However, our solution is considered valid only under circumstances where the inherent risk-driven nature of the agile development model is insufficient.

The need of risk management depends on several factors, but primarily it subsumes to project size, type of development, product complexity and overall project risk profile. More specifically, the results show that the integration of risk management adds most value in development contexts with one or several of the following characteristics:
teams consisting of more than ten people
- distributed teams
- development of safety critical, security critical, embedded systems, other complex software systems, entirely new systems and/or innovative systems
- projects with high risk exposure, e.g. depending on factors such as the external environment, organization, product status, technology, schedule, and so forth.

Summing up, the main conclusions of this paper are (a) integration is a valid solution for addressing the current lack of risk management in agile development, however only in certain projects and organizations, (b) both the integration model and the integrated model need to be further elaborated in terms of the guidance they provide, and (c) the solution needs to be further investigated in terms of its applicability in practice.

3.3 Other publications

Other research that has been published during the course of writing this doctoral thesis is listed below, starting with the most recent publications first. Note that in publications before June 2006, the author’s name was Wäyrynen (maiden name).

3.3.1 Conference and journal papers


Salinesi C. and Wäyrynen J., A Methodological Framework for Understanding IS Adaptation through Enterprise Change. In proceedings of the 8th In-
ternational Conference on Object-Oriented Information Systems (OOIS'02), France, August 2002.

3.3.2 Technical reports


4 Extended background

In this chapter, we provide an extended background of the domain of study. It includes definitions of some basic terms and concepts as used in this research and a brief overview of agile development and software risk management, respectively.

4.1 Methodology, method, process or process model?

The distinction between methodology, method, process, and process model is sometimes fuzzy. They are often used as synonyms despite their large differences in meaning. The distinction is however important and useful. Generally, they are defined as:

- **Methodology**: “a body of methods, rules and postulates employed by a discipline: a particular procedure or set of procedures” or “the analysis of the principles or procedures of inquiry in a particular field” (Merriam-Webster, 2008)
- **Method**: “implies an orderly logical arrangement usually in steps” (Merriam-Webster, 2008)
- **Process**: “a series of actions or operations conducing to an end” (Merriam-Webster, 2008)
- **Process model**: “a simplified description of a software process which is presented from a particular perspective” (Sommerville, 2006)

The term methodology can be used in two ways. It can be used to refer to the study of methods, but it can also be used to describe the rules and principles for an entire family of related methods. In this research, we use it to describe the general principles and inherent elements of agile software development. For this reason, it necessarily covers a larger scope than the concept of method which generally is used when referring to a particular method within the methodology. In this respect, Scrum and eXtreme Programming represent two methods within the agile methodology, which also frees the term method from being referred to as a methodology. However, each agile method, in turn, specifies specific methods, practices or techniques for conducting certain agile software development activities. For instance, Stories
Figure 4.1 Cockburn’s (2002) definition of methodology, its elements and their relations

are used as a method for specifying requirements. For this reason, we do not use the term method to refer to individual agile methods, but we use it to refer to the specific development techniques or practices that each of them prescribe. Consequently, we need to replace the term method with another term when referring to any of the members of the agile methodology, such as Scrum or eXtreme Programming. For this purpose, we use process. A process specifies a particular course of action intended to achieve a result by describing what, when, how and by whom things should be done, which better reflects our intentions in this research. Finally, a process model is an abstraction of the software process being described. In this respect, we use the term process model to refer to a generic description of a software process. Our synthesized agile and risk model management models are examples of process models (see Chapter 4).

In this research, we have used Cockburn’s (2002) definition of methodology as a starting point for understanding the fundamentals of agile software development. Cockburn’s (2002) definition has several advantages. It reflects the particularities of an agile software development environment and the fundamental elements of agile software processes. During the course of our research, it has facilitated the recognition of the elements of agile software processes such as activity, technique and role (see Figure 3). As a result, it has helped us in identifying these elements in the data collection and analysis of research data, and also facilitated the structuring of the research work and presentation of research results.

The elements and their relationships to other elements are depicted in Figure 4.1. Their definitions and relationships are also described below (Cockburn, 2002):
• **Role**: A person who is part of the team, is involved with one or several activities and who beholds the skills and the personal traits needed to carry out the activities.

• **Personality**: The personal traits needed attributed a particular role to carry out the activities.

• **Skill**: The skills required for the role, for instance education or talent in particular techniques or tools.

• **Team**: The group of roles that work together to carry out activities.

• **Technique**: The specific procedure for how to accomplish an activity, that is, a method.

• **Tool**: The means, tools or equipment used to support particular techniques, requiring particular skills and which are described in some tool standard.

• **Activity**: The task or tasks to be carried out by the team or by individual roles to produce the product.

• **Process**: A particular course of action intended to achieve a result or milestone, that is, how activities fit together, often with a start and end condition, implying an orderly logical arrangement.

• **Product**: An output or deliverable, i.e. what is produced with particular techniques in one or a set of activities and described in a particular product standard.

• **Milestone**: A time-related performance indicator marking work progress or completion, that is, marking whether milestones are met or not met at a particular point of time.

• **Standards**: The conventions adopted for tools, products or techniques.

• **Quality**: The criteria used in assessments and evaluations of the products or the activities. Quality can be of both quantitative and qualitative nature.

• **Team value**: The underlying philosophy of the team.

These are our definitions of some of the basic terms and concepts used in this thesis. In the following, we will present a brief overview of the objects of study, agile software development and risk management.
4.2 Agile software development

During the last decades, developers of software intensive systems have primarily applied an engineering approach to development. The traditional, sequential lifecycle model including rigorous planning and design upfront and a constant care to ensure conformity to the plan has served many software development projects well. For instance, a discipline that has taken advantage of the sequential lifecycle model is security engineering (Boström et al., 2006). Early availability of planning and design artifacts for auditors to examine has facilitated the task of certifying system conformity to external standards, and has often also made the life of the software acquirer easier.

On the other hand, the failure rate of software projects is still alarming. The actual success rate of software projects is low (Standish Group, 2001). Much of it is due to management practices (Boehm, 1991). Software design is more akin to research than to construction or manufacturing, and many of the management paradigms adopted from these engineering fields were simply not adapted to the needs of the software domain.

The low success rate in software projects has spawned the emergence of a new breed of approaches to managing projects, known collectively as agile software development. Agile starts with the premise that software development is too complex and unpredictable to be planned exactly upfront and in advance (Schwaber and Beedle, 2001). For this reason, one must adapt the software development process to manage unpredictability.

Agile software development is based on the practices of iterative and incremental development, where simplicity is one of the core values (Beck, 2004). It relies on gradual emergence of the design and the requirements, and emphasizes direct face-to-face communication rather than the written documentation of the traditional plan-driven approach. To manage unpredictability, the agile model exploits quick feedback loops in the process. Rather than using the sequential “plan-design-build” approach, they proceed by multiple short “speculate-collaborate-learn” cycles (Cockburn, 2005).

It can be said that in order for a software development process to be considered agile, it should be (Boehm and Turner, 2005):

- **Iterative**: deliver software in small chunks of working functionality
- **Incremental**: not deliver the entire product at once, but in several increments
- **Simple**: it should be easy to implement and build up according to the project needs
- **Adaptive**: allow requirements changes
Figure 4.2 XP process (Source: XP, 2001)

- **Self-organizing**: enable clients, developers and any other resources needed work effectively together
- **People-oriented**: consider people as the most important driver of project success

There are a number of existing agile software development processes, where the most known ones are eXtreme Programming (XP) (Beck, 2004), Scrum (Schwaber and Beedle, 2001), Lean Software Development (Popependieck, 2003), Feature Driven Development (FDD) (Palmer and Felsing, 2002), Adaptive Software Development (ASD) (Highsmith, 2002), Crystal (Cockburn, 2005) and Dynamic Systems Development Method (DSDM) (Stapleton, 1997). Because XP and Scrum are the most widely used agile processes (Charette, 2001), we have chosen to base our research on them as representatives of the group of agile software development processes. They are presented in more detail below.

### 4.2.1 eXtreme Programming

eXtreme Programming is an agile software development process which primarily describes techniques aimed at producing working software and robust software designs.

The cornerstones of XP are a set of tightly interconnected values, practices and principles. Values are defined as high level criteria that delineate the purpose of any project activities in XP, that is, the practices (Beck, 2004). A practice is a technique for developing software. Because values are universal and practices are situation-dependent, the XP principles are used as guidelines for guiding what practices should be implemented in a particular project situation. The values, principles and practices will not be presented...
here, but we refer to Beck (2000, 2004) when any of them or other specific XP concepts are used in the thesis.

As depicted in Figure 4.2, the XP process is structured into short iterations, preferably two-week iterations. An iteration starts with an iteration planning session to plan the work of the iteration. The main input consists of requirements, called Stories in XP (Beck, 2004). However, the input could also consist of the results from conducting architectural spikes. An architectural spike is a very simple program to explore potential solutions to figure out tough technical or design problems (Beck, 2000). They are thrown away. The goal is to reduce the risk of a technical problem or to increase the reliability of a Story estimate. Once planning of the iteration is completed the actual implementation starts. Here, one uses techniques such as pair programming, test-first programming and refactoring to develop the software. Acceptance tests are specified in parallel with the implementation. They are conducted as soon as a piece of functionality has been implemented to enable quick feedback on the quality of the developed software. The iteration ends with a formal customer approval before the functionality developed is released.

### 4.2.2 Scrum

Scrum, just like other agile processes, starts with the premise that software development is too complex and unpredictable to be planned exactly in advance. Instead, empirical process control must be applied to ensure visibility, inspection, and adaptation. Hence, Scrum applies the ideas of industrial process control theory (Schwaber and Beedle, 2001).
Unpredictability means that some work processes cannot be perfectly controlled or perfectly defined. Processes with these qualities require empirical process control. The basic attribute of empirical process control constitutes a continuous cycle of inspecting the process for correct operation and results and adapting the process as needed. (Berteig, 2005)

In Scrum terms this means that the different environmental and technical variables such as time, quality, requirements, resources, technologies and tools, and the development process must be controlled constantly in order to be able to adapt to changes flexibly. This is achieved through an iterative and incremental development process (Schwaber and Beedle, 2001).

We will not describe Scrum in detail but refer to (Schwaber and Beedle, 2001) when any concepts specific to Scrum are used in this thesis. However, Scrum's process skeleton is shown in Figure 4.3. The lower circle represents an iteration of development activities that occur one after another. The output of each iteration is an increment of the product. The upper circle represents the daily inspection that occurs during the iteration, in which the individual team members meet to inspect each others' activities and make appropriate adaptations. Driving the iteration is a list of requirements, called a backlog. This cycle repeats until the project is completed.

4.3 Software risk management

As business reliance on software grows, so do the business related consequences of software failure (Addison and Vallabh, 2002). As we have already pointed out, many software projects take place in an unpredictable environment in which many pitfalls exist that may affect the successful outcome of a project. One of them is risk.

Risk management is a structured approach to managing uncertainty through a collection of methods, techniques and guidelines aimed at identifying, analyzing and controlling risk (PMI, 2004). Hence, it is the application of appropriate tools and procedures to contain risk within acceptable limits (Wiegers, 1998).

Risk management has for many years been practiced within various traditional domains such as finance, insurance, manufacturing, medicine, warfare, and the like. Risk management in the software engineering domain is, however, a relatively young discipline. Only during this past decade has it become recognized as a best practice in the software industry (Brown, 1996) (Wiegers, 1998).

Today, however, there exist several software risk management approaches, models and standards. For instance, the spiral model (Boehm, 1988), Continuous Risk Management (SEI, 2007), IEEE 1540 Risk Man-
management Standard (IEEE1540, 2001), and as one of the knowledge areas in the Project Management Body of Knowledge (PMBoK) (2004).

Software risk management generally follows the same structure as prescribed in risk management standards and approaches in other disciplines (Padayachee, 2002). The first step in a typical risk management program is the identification of risk, usually involving the use of checklists, questionnaires or brainstorming sessions. The next step is the analysis of the risks identified in such a way that they can be prioritized in a meaningful manner. This is followed by the development of countermeasures to either prevent risks from affecting the project or to reduce their impact. The final step is risk monitoring, which ensures that the risk reducing methods are implemented effectively and determines whether or not the risk-reducing tactics are in fact reducing risks.

Westfall (2001) summarizes that the risk management process is an ongoing part of managing the software development process. It is designed to be a continuous feedback loop where additional information and risk status are utilized to refine the project’s risk list and risk management plans.

Below, we provide a brief overview of a subset of the risk management models studied in this thesis. We choose these models primarily because they are created by international organizations recognized worldwide. In addition, although they present similar structure and characteristics there are some differences. In this way, they complement each other and hence, taken together they provide a more comprehensive overview.

4.3.1 AS/NZS 4360 Risk Management Standard

A risk is defined by the AS/NZS 4360 Risk Management Standard as “the possibility of something happening that impacts on your objectives. It is the chance to either make a gain or a loss. It is measured in terms of likelihood and consequence” (Standards Australia, 2004). According to the AS/NZS 4360, effective management of risk will not only mitigate risk, but also enable maximizing the opportunities and achieve desired outputs.

The AS/NZS 4360 is a generic standard. Hence, it is not specific to software development but applies to any organization, public or private, group or individual. This standard specifies the elements of the risk management process and its inherent phases intended to cover the entire lifecycle of an activity. However, it emphasizes that the eventual design and implementation of the risk management process will be influenced by the varying needs and goals of an organization.

The AS/NZS 4360 risk management process is described as “the systematic application of management policies, procedures and practices to the seven tasks of communicating, establishing the context, identifying, analyzing, evaluating, treating, and monitor and reviewing risks” (Standards
Figure 4.4 Overview of the AS/NZS 4360 risk management process (Source: Standards Australia, 2004)

Figure 4.4 provides an overview of the AS/NZS 4360 risk management process and its seven phases.

4.3.2 Project Management Body of Knowledge

Risk management in the PMBoK is defined as “the systematic process that identifies, analyzes and respond to risk in projects” (PMI, 2004). Its intention is to maximize the probability of positive effects of events and to minimize the probability and consequences of events with negative impact on the project goals.

In the PMBoK, risk management is one of the nine knowledge areas of project management (PMI, 2004). The knowledge area concerns project risk management and describes six processes that are necessary for conducting risk management. They are listed and briefly described below:

- **Risk management planning process** for deciding how to approach, plan and execute the risk management activities
- **Risk identification process** for determining which risks might effect the project and documenting their characteristics
- **Qualitative risk analysis process** for prioritizing risks for subsequent further analysis or action by assessing and combining their probability of occurrence and impact
Figure 4.5 IEEE 1540 risk management process (Source: IEEE1540, 2001)

- **Quantitative risk analysis process** for numerically analyzing the effect on overall project objectives of identified risks
- **Risk response planning process** for developing options and actions to enhance opportunities and to reduce threats to project objectives
- **Risk monitoring and control process** for tracking identified risks, monitoring residual risks, identifying new risks, executing risk response plans, and evaluating their effectiveness throughout the project life cycle

The PMBoK is a general risk management standard that is intended to apply to any type of project.

### 4.3.3 IEEE 1540 Risk Management Standard

The IEEE 1540 Risk Management Standard (IEEE1540, 2001) uses the ISO risk terminology and defines risk as “as the combination of the probability of an event and its consequences” (ISO/IEC 73, 2002). It states that there is the potential for events and consequences that constitute opportunities for benefit or threats to success in all types of software development undertaking.

The IEEE 1540 is a standard particularly aimed at risk management within engineering. It addresses the need for a common terminology and
framework to communicate risk within the engineering field and it groups existing risk management standards in engineering to avoid redundancy (Charette and O’Brien, 1999). It focuses on a set of core risk management practices, thus providing a minimal standard for risk management in engineering.

The IEEE 1540 standard defines the risk management process as a continuous model for systematically identifying, analyzing, treating, and monitoring risks throughout the lifecycle of a product or service. Its purpose is to define and implement risk management strategies and measures, which will lead to actions that help a company to reduce or avoid the impact of risks (IEEE1540, 2001). An overview of the IEEE 1540 risk management process and its phases is depicted in Figure 4.5.

4.3.4 Software Risk Evaluation Method

The Software Risk Evaluation (SRE) is a process for identifying, analyzing, and developing mitigation strategies for risks in a software-intensive system while it is in development (Williams et al., 1999). Hence, it provides the guidelines for the initial phases of risk management. It is developed by the Software Engineering Institute (SEI) (SEI, 2008).

The SRE follows the SEI risk management paradigm called Continuous Risk Management (CRM) (SEI, 2008). CRM defines risk management as a software engineering practice with processes, methods, and tools for managing project risks. It describes three steps for managing risks in a project:

- assess continuously what can go wrong (risks)
- determine what risks are important to deal with
- implement strategies to deal with those risk

The basic principle in CRM model is that the risks have to be taken in account in a continuous way in all phases of the project. This means that the risk management process never stops until the project is finished, and that the process is dynamic and sensitive to other changes in the project. Another important part of the is communication which, as depicted in Figure 4.6, is a core phase in the process and should be present through all phases. The risk management phases identify, analyze, plan, track, and control risk should occur continuously, concurrently and iteratively. This means that while new risks may be discovered, analyzed and planned for, old risks are being tracked and controlled.
The SRE addresses the identification, analysis, planning, and communication elements of the SEI risk paradigm. The SRE is typically the initial method used on a project. The analysis element is also covered fully by SRE activities. Planning elements are partially addressed through the construction of high-level mitigation strategy plans. The SRE also contributes significantly to the communication element, for instance by the templates it describes for communicating risk. The remaining elements of the paradigm, tracking and control, are not addressed during an SRE.

In this chapter an extended background to the domain of study has been presented. In the following, we will elaborate on the research results.
5 Synthesized models

In this chapter, we present a synthesized agile model and a synthesized risk management model. They constitute a basis for outlining the integrated model, that is, the model integrating risk management and agile development.

This chapter is based on the research presented in Papers 2, 3, and 4.

5.1 Creation of synthesized models

In this research, we investigated the status of the agile and risk management models in the industry which we compared to a subset of existing agile and risk management models. This comparison has resulted in two synthesized process models: a synthesized agile process model and a synthesized risk management process model. They are depicted Figures 5.1 and 5.2 and presented in sections 5.2 and 5.3, respectively.

The main reason to creating the synthesized models was the underlying goal of our research. We did not want to bind our integration model to a particular agile model or risk management standard. Hence, we decided to create synthesized models of both the agile development and risk management processes, which we also compared to industrial models. In this way, they would represent general and more comprehensive models of the studied processes and also reflect their industrial state of practice as this research is of empirical nature.

5.2 Synthesized agile model

There is a number of existing agile development processes. However, we decided to base our work on eXtreme Programming and Scrum. We selected these because they are the most widely accepted models but also because they complement each other (Charette, 2001). Whereas XP has a definite focus on development techniques (test-driven development, refactoring, con-
continuous integration), Scrum concentrates on the management of software projects (Schwaber, 2004). Hence, together, they constitute a comprehensive framework covering both the engineering and management process levels. In creating the model, we first elicited the activities belonging to their respective phases and put them into a synthesized model. When doing this, however, we observed that some customary development activities were missing in both Scrum and XP. To ensure its comprehensiveness and that no activity would be missed in our research, we complemented the model with activities from the standard software process model (IEEE12207, 1998). In this way, we could enquire about their applicability in an industrial agile context.

The final result of their synthesis, including the results from the studies of their status in the industry, is presented in Figure 5.3 and Figure 5.4. Each activity is marked with its origin, where XP stands for eXtreme Programming, S stands for Scrum, and HW stands for heavyweight. I stands for the activities elicited from industry.

As depicted in Figure 5.1, the synthesized agile process model covers the entire development process and its inherent phases on two organizational levels: the Business and Engineering levels. The Business Level consists of the Product Vision Planning phase. The Engineering Level consists of the
Figure 5.3 Synthesized Pre-Implementation Process Model. S stands for Scrum. XP stands for eXtreme Programming. HW stands for heavyweight. Bullets (●) and (I) stands for the activities elicited from industry.
Figure 5.4 Synthesized Implementation Process Model. S stands for Scrum. XP stands for eXtreme Programming. HW stands for heavyweight. Bullets (●) and (I) stand for the activities elicited from industry.

Product Roadmap and Release Planning and Implementation phases. They will be briefly described below. Due to space restrictions we cannot present the entire synthesized model in one figure, hence we have divided it into two figures, Figure 5.3 and 5.4. Figure 5.3 covers the following phases: Product Vision Planning, Product Roadmap and Release Planning, and the first part of the Implementation phase, called Iteration Planning. These represent the
phases that take place prior to implementation. We call it the Pre-Implementation Process Model. Figure 5.4 covers the remaining part of the Implementation phase, that is, the Iteration Implementation phase where the actual implementation takes place. Hence, we call it the Implementation Process Model.

5.2.1 Product vision planning
The Product Vision Planning phase belongs to the Business Level. It is the first phase in the synthesized agile development process (see Figure 5.1). It is primarily conducted by business managers together with the product managers. It involves creating a product vision plan describing the product goals, overall business and product structure and return on investment. The resulting product vision plan also guides the work carried out in subsequent planning, decision making, and development (Schwaber and Beedle, 2001).

5.2.2 Product roadmap and release planning
The Product Roadmap and Release Planning phase is the second development phase in the synthesized agile model. It belongs to the Engineering Level and it is supervised by the product manager. It consists of three sub-phases: Release Preparation, Product Roadmap Planning and Release Planning (see Figure 5.1).

Release Preparation involves preparatory activities for the whole phase. Here, one schedules the planning meetings and gathers all the necessary information, including the product vision plan and high-level requirements specification. In the Product Roadmap Planning phase, one first makes a high-level product roadmap outlining future releases. One then plans the actual release in the Release Planning phase. Here, one first creates a high-level plan for the identified releases which one then regularly revisits for more detailed planning before each release starts (Scrum, 2003).

5.2.3 Implementation
The third agile development phase is the Implementation phase. It is performed by a team leader together with team members. The Implementation phase consists of two sub-phases: Iteration Planning and Iteration Implementation (see Figure 5.1).

Iteration Planning is conducted at the start of each iteration. It involves three sub-phases, Iteration Preparation, Iteration Scoping and Task Planning. Here, the team, product management and other relevant stakeholders meet to plan the work to be conducted in the coming iteration (Beck, 2004).

In the Iteration Preparation sub-phase, one conducts preparatory activities for starting the iteration, that is, one schedules the iteration planning
meeting and collects all the necessary input to plan the iteration effectively. In the next two sub-phases (Iteration Scoping and Task Planning), one conducts the actual iteration planning. More specifically, in the Iteration Scoping phase, one studies, analyses and prioritizes the requirements planned for the next iteration in order to determine its scope. In the Task Planning sub-phase, one makes an execution plan to implement the prioritized requirements. The Iteration Planning phase takes place during the first days of the iteration.

The Iteration Implementation phase consists of four sub-phases, Iteration Preparation, Daily Status, Development and Iteration Completion. The Iteration Preparation sub-phase involves making initial preparations for starting the consecutive iterations. The Daily Status is a meeting that takes place on a day-to-day basis throughout the whole iteration. Its goal is to establish project status and to adjust commitments, if necessary (Schwaber and Beedle, 2001).

In the Development phase, the requirements planned for the iteration are implemented and tested. The goal is to deliver an increment of working functionality (Beck, 2004). During the Iteration Completion phase, one delivers the functionality developed during the iteration, evaluates the results, and identifies the improvements to be made in the next iteration (Scrum, 2003).

5.2.4 Agile state of practice

In this research, we have studied the agile state of practice in three Canadian software organizations. They are described in Chapter 2, section 2.4. Our results show that the organizations studied almost follow the synthesized model. There was however some discrepancies observed. Below, we list and describe some examples of the observed differences. The entire study is presented in Paper 2 and Paper 3.

- Organizational Planning Levels: The agile models studied do not describe organizational levels of planning. Our study shows evidence that the organizations studied conduct planning on two levels. The levels recognized in the industry are the Business and Engineering levels. Product Vision Planning, as described in our model, takes place on the Business level, whereas the Product Roadmap and Release Planning and Iteration Planning take place on the Engineering level. We believe that it is a useful division because it clearly communicates the purpose of the different planning phases. Hence, we this led us to modify our original model and designate two levels of planning, the Business and Engineering levels (see Figure 5.1).
• Contract management: The agile models do not suggest any contract writing on the developers’ level. However, this practice is spreading in the organizations studied. It is claimed that it is a necessary quality assurance activity. We have found that the developers sign under their work assignments in the Iteration Planning phase (see Figure 5.3 and 5.4) creating a Team Service Level Agreement for the deliverables. At the end of the iteration, the deliverables provided by the developers and teams are measured against these agreements. In this respect, we have identified two additional activities relevant for the Iteration Planning and Iteration Completion phases. For this reason, we added the two new activities Write/Sign-off personal/team service level agreement in our synthesized model.

• Activity gaps: We have also found that several of the activities that we have added in the synthesized model from the standard software development model are applied in the organizations studied. The organizations claim that Scrum and XP do not provide full coverage of all software development activities, especially not for larger projects. Hence, they had to fill the gaps by finding complementary activities from standard approaches. These concern activities such as requirements elicitation, analysis and specification, architectural analysis and design, documentation practice, guidelines regarding scalability, risk management and thorough task planning.

The synthesized agile process model we have built is based on XP and Scrum. It also reflects the state of practice of three software organizations. Our results show that the organizations studied have adopted an agile approach although they have added some traditional activities. Our interviewees claim that the degree of agility varies among projects. It depends on the phase in the product lifecycle, project type, its size, criticality, innovative character, degree of uncertainty, risk taking, permission from stakeholders’ side to take risk, and budget. For instance, the majority of their fixed budget projects follow a more traditional pre-implementation approach whereas other yet unfunded projects follow a more agile approach. Also, in small, innovative, creative and totally new projects, one has difficulties to conduct detailed planning in advance. Hence, one mainly outlines product vision and high level requirements within the pre-implementation phases and allows the rest to be resolved within the later phases. This observation makes us conclude that two different agile paths are developed: (a) one for projects requiring more upfront planning, such as for instance evolution and maintenance projects, and (b) another one for projects requiring less formal and more agile planning, such as for instance totally new innovative projects.
5.3 Synthesized risk management model

The risk management standards and models studied differ somewhat with respect to various risk management practices. In order not to miss anything, we have therefore synthesized these models into one common model in a similar way as we did with the agile models. Our goal was to create as comprehensive a process model as possible covering all the issues as suggested by the current risk management standards and models. As mentioned in earlier chapters we have focused our studies in risk management on the IEEE 1540, PMBoK and AS/NZS 4360 standards and the SRE method, which are also used in the creation of the synthesized risk management model.

Our synthesized risk management process model consists of six phases: Risk Identification (RI), Risk Analysis (RA), Risk Management Planning (RMP), Risk Monitoring and Control (RMC), Risk Sign-Off (RSO) and Risk Post-Mortem Analysis (RPMA). The phases and their activities are briefly described below. They are also listed in Figure 5.5. Due to space restrictions, instead of referring to the originating standard in the figure as we did with the agile model, we provide it in the textual description of the model.

5.3.1 Risk identification

During the Risk Identification phase, one makes an inventory of potential risks that may have impact on the achievement of the predetermined objectives (Standards Australia, 2004). As listed in Figure 5.5, the phase starts with preparations for the actual risk elicitation. Preparations may include study of relevant historical information and of the domain that is subject to risk exposure. The historical information includes taxonomy of risk types, lessons learned, and any other relevant information (Williams et al., 1999).

One then elicits risks. The choice of the elicitation technique depends on the context. It may involve brainstorming, interviews, scenario analysis, prototyping, or some other knowledge acquisition technique (Carr et al., 1993) (IEEE1540, 2001) (Williams et al., 1999).

When eliciting information about risks, one first identifies them, their consequences, effects, sources, root causes, and categories (IEEE1540, 2001). One then describes and records each of the identified risk (Carr et al., 1993). Finally, one creates a list of risks and circulates it around all the relevant stakeholders for possible complementary additions, improvements, and confirmation.
Figure 5.5 Synthesized risk management model

1 Risk Identification
- RIA1: Study relevant historical risk information
- RIA1.1: Study for organizational taxonomy of risk types
- RIA1.2: Study for organizational taxonomy of risk types
- RIA1.3: Study other relevant information, if needed
- RIA2: Study the domain that is subject to the risk exposure
- RIA3: Define risk
- RIA3.1: Identify potential risks
- RIA3.2: Identify risk consequences and effects
- RIA3.3: Identify risk sources
- RIA3.4: Identify risk response strategies
- RIA3.5: Define risk categories/bases
- RIA4: Determine and record each identified risk
- RIA5: Create risk list
- RIA6: Update risk list accordingly
- RIA7: Confirm risk list

2 Risk Analysis
- RA-A1: Analyze risks
- RA-A1.1: Analyze Step 1: Analyze each risk independently
- RA-A1.2: Assess risk probability
- RA-A1.3: Assess risk impact
- RA-A1.4: Calculate risk exposure
- RA-A1.5: Specify risk severity
- RA-A1.6: Analyze risk information gathered using appropriate technique
- RA-A1.7: Specify probability risk threshold value
- RA-A1.8: Assign priority to the risk
- RA-A1.9: Record any assumptions made
- RA-A2: Analyze Step 2: Analyze risk in groups
- RA-A2.1: Group risks according to the chosen grouping criteria
- RA-A2.2: Analyze risks in a group
- RA-A3: Consolidate risk prioritization
- RA-A4: Summarize risk information
- RA-A5: Create a list of risks requiring further attention
- RA-A6: Suggest a preliminary plan for managing the risks
- RA-A7: Circulate risk list and preliminary plan among stakeholders
- RA-A8: Update the risk list and the preliminary plan, if needed
- RA-A9: Confirm the risk list and the preliminary plan

3 Risk Management Planning
- RMP-A1: Study the risk list, the analysis results, and the preliminary plan
- RMP-A2: Determine strategies for managing risks
- RMP-A3: Create a risk management plan
- RMP-A4: Create a control and monitoring plan
- RMP-A5: Develop a fallback action plan if the primary action plan does not prove adequate
- RMP-A6: Document the action plan
- RMP-A7: Create contingency plan, if needed
- RMP-A8: Define relevant contingency actions
- RMP-A9: Document the contingency plan
- RMP-A10: Make a schedule for implementing the plans
- RMP-A11: Identify constraints
- RMP-A12: Estimate effort
- RMP-A13: Estimate resources
- RMP-A14: Assign budget
- RMP-A15: Assign roles for managing the plan
- RMP-A16: Conclude actions and put them into the risk management plan
- RMP-A17: Circulate the action plan to the stakeholders concerned
- RMP-A18: Update the risk management plan, if needed
- RMP-A19: Conclude risk management plan
- RMP-A20: Update and reconcile the documentation of the risk management plan

4 Risk Monitoring & Control
- RMC-A1: Ensure there are procedures to monitor risk
- RMC-A2: Monitor continuity and risks following the plan
- RMC-A3: Ensure the risk status
- RMC-A4: Control the changes in the risk status
- RMC-A5: Reconcile the risk status
- RMC-A6: Take appropriate measures with respect to the changed status
- RMC-A7: Implement the contingency plan, if needed
- RMC-A8: Circulate the risk management plan to the stakeholders concerned
- RMC-A9: Update the risk management plan, if needed
- RMC-A10: Confirm risk management plan
- RMC-A11: Update and reconcile the documentation of the risk management plan

5 Risk Post-Mortem Analysis
- RPA-A1: Evaluate the risk management process
- RPA-A2: Check other organizational risk taxonomy
- RPA-A3: Identify deficiencies and failures of the process
- RPA-A4: Identify positive effects of the process
- RPA-A5: Identify lessons learned
- RPA-A6: Record results

Feedback to future risk management and process improvement
5.3.2 Risk analysis

During the Risk Analysis phase, one analyzes and prioritizes risks (IEEE1540, 2001). As a first step, one analyzes each risk independently. Here, one studies the identified risk, and assesses its impact, probability, risk exposure and the severity (IEEE1540, 2001). The analysis can be conducted using different techniques, e.g. matrices, decision trees, scenario analysis, game theory, and sensitivity analysis (PMI, 2004).

As a second step, one groups and analyzes the related risks to facilitate their collective mitigation (Carr et al., 1993). One then consolidates the risk prioritization and creates a top-priority risk list (Boehm, 1991). A watch list is also created for all other risks (Williams et al., 1999). Based on the analysis results, one suggests a preliminary plan for managing each risk.

Finally, the prioritized risk list is circulated among the stakeholders for confirmation. It may be updated, if qualified changes are proposed (Williams et al., 1999).

5.3.3 Risk management planning

In the Risk Management Planning phase, one creates concrete plans determining strategies, options, and actions relevant for managing the identified risks (IEEE1540, 2001).

One starts the phase with studying the risk list, the analysis results, and the preliminary plan (Williams et al., 1999). For each risk or risk group, one first determines appropriate strategies (PMI, 2004), and then creates and documents the following three plans:

1. The Control and Monitoring Plan defining relevant measures or metrics for monitoring and controlling the risk (PMI, 2004). It also includes determining the performance indicators for measuring the action effectiveness (PMI, 2004).

2. The Risk Action Plan determining the actions to be used for treating a certain risk or risk group (Standards Australia, 2004). These actions may be complemented with fallback actions in cases when primary actions do not prove effective (Williams et al., 1999).

3. The Contingency Plan specifying the actions to be taken in cases when severe risks materialize and become a serious problem (PMI, 2004). Contingency plans are variants of action plans to be used when attending to very serious risks.

Once the plans are defined, one identifies constraints, estimates the effort, resources, scope, schedules and budgets them, and assigns risk owners to
them (IEEE1540, 2001). One then combines all the three plans into one comprehensive Risk Management Plan (IEEE1540, 2001).

The Risk Management Plan is used for a cross-analysis, during which one makes various optimizations with the purpose of maximizing the impact of the planned measures and minimizing the effects of the possibly conflicting plans (Williams et al., 1999). To ensure that the identified risks get full and appropriate attention, one prepares contractual agreements, where each risk owner’s responsibilities are specified (PMI, 2004). Finally, one circulates updates and confirms the risk management plan and its related documentation.

5.3.4 Risk monitoring and control

In the Monitor and Control phase, one continuously monitors and controls the risks according to the risk management plan. One also identifies new risks and reports them (Standards Australia, 2004).

To make certain that risks are effectively monitored and controlled, one first ensures that there are risk monitoring procedures established (Standards Australia, 2004). For each risk or risk group, one then continuously monitors and records the status (PMI, 2004). In cases when it changes, one takes measures as specified in either the action or contingency plans. Finally, one updates and records the risk status (IEEE1540, 2001).

5.3.5 Risk sign-off

A formal Sign-Off phase is an important part of risk management assurance (Standards Australia, 2004). Here, one first reviews the risks and the way they have been mitigated. For each risk that is judged to be mitigated, one updates the risk management progress status, removes it from the risk list, and ensures that the risk list gets updated (Standards Australia, 2004). Finally, one signs it off.

5.3.6 Risk post-mortem analysis

In the Risk Post-Mortem Analysis phase, one collects and evaluates the risk management process and its results (IEEE1540, 2001). Here, one evaluates the information about the identified risks, their causes, treatment, the process used and the successes or failures of the actions taken (IEEE1540, 2001). Using it, one then creates or updates the organizational risk taxonomy (Carr et al., 1993). Finally, one identifies successes and failures in the process and generates lessons learned (IEEE1540, 2001). This in turn provides an important historical feedback for improving the future risk identification and management process (Standards Australia, 2004).
5.3.7 Risk management state of practice

We have studied the industrial practice of risk management within 37 software companies. Their profiles are presented in Chapter 2, section 2.4. Our results show that all the organizations studied have established their risk management processes. They also claim that our model was exhaustive, and that it does not miss any activities. However, regarding the industrial status of the risk management process that we propose, few organizations have implemented the entire process. The substantial majority of the organizations only apply the initial phases of risk identification and analysis. In addition, more than half of the companies studied have experienced problems with their risk management processes. The problems as experienced by the organizations studied vary. Some examples are:

- **Lack of process sophistication and completeness:** The risk management process is too simple and/or incomplete. It does not cover all aspects of risk management. It may only cover a subset of the phases as identified in our model, e.g., risk analysis or monitoring phases.

- **Lack of process formality/standardisation:** The organizations have not defined a standard risk management process. The organizations do not manage risks in a formal way. The process relies entirely on the experience of the individuals involved in it. This leads to the fact that risks are not formally planned and managed, thus creating problems with analyzing and monitoring risks. Hence, their risk management is very ineffective.

- **Lack of risk management process owner:** Some companies lack a role dedicated to risk management. This results in a fact that most of the responsibilities of the risk management process owner are managed by an already overburdened project manager, who, in turn, may not properly balance his responsibilities and priorities.

- **Lack of formal and standard documents for recording information about risks:** Organizations do not have any formal and standard guidelines for how to document information about risks and information about their activities. This in turn leads to a non-uniform way of documenting risks, thus obstructing various risk management process activities.

- **Integration problems:** Risk management is mainly conducted locally in the context of the software project. Hence, only a subset of risks can be managed. Other risks, such as environmental or organizational ones, cannot be managed by their current risk management process model due to the lack of organization-wide risk management.
This study of the risk management state of practice has provided us with an important feedback for identifying several shortcomings in the existing risk management standards and problems with risk management in software development. The aforementioned problems are just a few examples of the problems identified. The entire study is presented in Paper 4.

Summing up, this chapter has presented a synthesized agile and risk management model. They are a core part of the work of creating the integrated model. They comprise its two basic constituents. The model will be presented in the next chapter.
6 Integrating risk management and agile processes

In this chapter, we outline a model integrating risk management and agile development. To begin with, however, we examine the agile processes from a risk management perspective to identify the risk management aspects that are absent in the agile model. We do this by using a set of fundamental risk management aspects as our comparison criteria. We continue by proposing a model for integrating the risk management and agile process models. It is created based on the identification of integration points between the two processes as elicited from studies of the integration of agile and risk management processes in the industry. Using this integration model, we then outline the integrated model. As a last step, we evaluate the proposed solution. The evaluation is based on expert opinion resulting from interviews that are conducted with agile practitioners in the Swedish software industry. Finally, we close this chapter by summarizing the research contribution.

The work reported on in this chapter is based on Papers 1, 2, 3, 4, 5, and 8.

6.1 Comparison of risk management and agile models

Effective risk management prevents the likelihood that undesirable problems occur or it decreases the severity of their consequences, should they occur (IEEE1540, 2001). By identifying and attending to risks, it aids in making informed decisions and taking appropriate measures before risks become problems. Risk management helps avoid problems, rework, disasters and it stimulates successful project outcomes. For this reason, it should be an inherent component of software development (Boehm, 1988).

Agile software development models claim to be risk-driven (Beck, 2004). They state that their iterative approach enables continuous attention to risks throughout the whole development project. When studying the agile and risk management processes, however, we have found that the agile development models implement few risk management practices.
Figure 6.1 Fundamental risk management aspects

Extreme Programming claims to be a model that addresses risk at all levels of the development process (Beck, 2004). However, risk management is not explicitly described anywhere in the XP model. There are no specific guidelines for managing risks or activities describing how to identify, analyze and control them.

Scrum claims to be risk-driven because it is based on an iterative and incremental approach enabling early and frequent feedback making it possible to identify and reduce risk early (Schwaber and Beedle, 2001). In contrast to XP, however, Scrum makes some explicit statements about risk management during the development process. In Scrum, risk management is described as a part of the planning phase, where it is stated that risk should be identified, assessed and actions for controlling the identified risk defined (Scrum, 2003). The risks should be listed and planned for when defining the project. However, it is not explained how these risk management tasks can be carried out, and there are no precise guidelines for how to identify, classify, assess or manage risks.

These are examples of some general observations of the state of risk management in Scrum and XP. However, in order to more exactly identify the aspects of risk management that are omitted requires a more systematic examination of the agile models from a risk management perspective.

To accomplish this, we conducted an in-depth study of risk management standards and identified some fundamental aspects of risk management that we use as our comparison criteria. They are the following: (1) Risk definition, (2) Risk assessment, (3) Software lifecycle, (4) Templates, (5) Tools/Repositories, (6) Stakeholders, roles and responsibilities (7) Product status, (8) Environment, (9) Organization, and (10) Measures. Each aspect is listed in Figure 6.1 and briefly described below.

- **Risk definition**: Risk is defined as an event or a condition that may affect the outcome of a project (IEEE1540, 2001). It is characterized with two distinctive elements: probability and impact (Boehm, 1991). The probability defines the likelihood that a risk event may occur. The impact defines the outcome of a risk. Risk can be either a loss or a gain (an opportunity) (Standards Australia, 2004). A loss is an unwanted or negative ef-
fect whereas a gain is a positive or progressive effect. In this comparison, we investigate whether and how Scrum and XP define risk. A definition is a prerequisite for defining the risk management process. It helps in understanding the process and facilitates the comparison process.

- **Risk assessment:** To make right and informed decisions, it is of importance to correctly identify and analyze risks. Hence, one needs to classify and assess them (Carr et al., 1993).

  Risk taxonomies (classifications) help identify risks systematically and thereby facilitate the analysis process (Williams et al., 1999). However, they do not suffice for an exhaustive risk analysis. They need be complemented with various assessment properties (attributes) such as **Risk Probability, Risk Impact, Risk Priority, Risk Exposure**, and the like (Boehm, 1991). Values should be assigned to these attributes. Due to the fact that risk assessment is subjective, it may be difficult to assign relevant values (Williams et al., 1999). Hence, guidelines for assessing risks in form of various techniques should be provided.

  Taxonomies, assessment attributes and techniques greatly help organizations in planning various measures such as designation of risk management, estimation of the mitigation effort size, and identification of policies to guide them. We examine if XP and Scrum specify risk taxonomies. We also examine if they suggest attributes and techniques for assessing risks.

- **Software lifecycle:** Effort spent on mitigating risks within development may be wasted if one does not consider risks within the whole lifecycle process (IEEE1540, 2001) (PMI, 2001). Risk management activities may differ in various lifecycle processes. We investigate whether the agile models studied cover risk management within the total lifecycle process, what lifecycle processes they approach and whether they provide guidelines for these processes.

- **Stakeholders, roles and responsibilities:** Stakeholder roles are individual roles or groups of roles who have a stake in or may be impacted by a given activity (IEEE1540, 2001). Stakeholders can either be internal or external (Standards Australia, 2004). Internal stakeholders include any managerial or technical roles participating in a project, including project managers, developers, testers, maintainers, product owners, business analysts and managers, quality managers and support personnel. External stakeholders are other roles, such as customers, contractors, suppliers and sponsors.

  The coverage of stakeholder roles within risk management is very important. It is only then one may be sure that all the risk sources and targets
have been identified and scrutinized from all possible perspectives. Designation of roles is a prerequisite for defining risk management process and responsibilities within the process (IRM, 2002) (IEEE1540, 2001). We investigate what stakeholder roles are covered by Scrum and XP, and which of the roles are provided guidelines for managing risks.

- **Supporting tools/repository:** To enable effective risk information management, analysis and tracking, organizations need repositories (preferably electronic) for documenting risks and the risk management process (IEEE1540, 2001). They should also be able to extract important experience and lessons learned which they may in turn use in various contexts, such as process assessment, improvement, root cause analysis, resource assignment, and the like. For this, they need experience base recording historical risk information. We investigate whether the agile models studied suggest use of tools and repositories for documenting risk management information.

- **Template:** A clear, complete and correct risk description is an important prerequisite for its effective management (Carr et al. 1993) (Hulett, 2001). To aid in maximizing the quality of the risk information, one should provide guidelines for what information should be managed during the risk management process (Williams et al., 1999). We investigate whether the agile models provide suggestions for how to structure risk information. We also investigate the coverage of the information as managed by them.

- **Product status:** Instances of the risk management process vary strongly with respect to the quality of the product, its life expectancy, and life cycle stage (IEEE1540, 2001). For this reason, risk management models should consider the product aspects when managing risks. Risk management may vary greatly in aged systems whose quality is undermined, life expectancy is low, and the system is close to retirement. We investigate if XP and Scrum consider product status related issues such as quality, life expectancy and lifecycle stage.

- **Environment:** To implement risk management effectively, the team should consider the project in its cultural, social, international, political and physical contexts (PMI, 2004). Systems may be developed in distributed and non-distributed environments. In a non-distributed environment, the team(s) is(are) co-located and work(s) together. The environment is 100% non-distributed if the customer is internal.
Figure 6.2 Risk management, its core and interfacing processes

Regarding the distributed environment, it can span from having to deal with only external customers to having to deal with distributed teams in different organizations and different countries, where aspects such as culture can play a crucial role for the effectiveness (Hofstede, 2003). Hence, the degree of distribution and its associated risks should be considered in risk management. We investigate whether the agile models consider a distributed environment and if guidelines are provided for addressing various environmental matters.

- **Organization:** To adopt a risk management program successfully, factors such as attitude towards risks, organizational maturity, competency and training should be considered (PMI, 2004). It is harder to implement risk management effectively in immature and incompetent organizations or with risk averse attitude (Hulett, 2001). Hence, all the stakeholders involved should acquire proper training in the product they produce/acquire and the processes they use. We examine whether the agile models consider attitude, competence and maturity and provide guidelines for dealing with them.

- **Measures:** One of the main purposes of risk management is to identify uncertainties and take actions to either remove or transform them into acceptable risks (Xu et al., 2005). For this, one needs a portfolio of appropriate measures in place. Such a portfolio covers processes suggesting activities for attending to risks, resources required for performing the processes and policies for ensuring that procedures and strategies for conducting risk management are defined and agreed upon (PMI, 2004).
The middle box in Figure 6.2 outlines a risk management process as covered by most of the risk management models and standards today, e.g. by the AS/NZS 4360 (2004), IEEE1540 (2001) and PMBoK (2004). To provide useful feedback to the organization, the process needs be integrated with other organizational processes such as various lifecycle and measurement processes and policy management. These processes should continuously provide feedback on risks, their progress, resources used, effectiveness of the resources and policies chosen, and plans for contingency readiness (Westfall, 2001). We investigate whether the models studied suggest risk management processes and whether they provide guidelines for resource management, process measurement, policy management, and integration with other processes.

The results of the comparison show that the agile models studied make some assertions about risk management. However, they do not provide any detailed suggestions for managing risks, thus leaving many areas unattended. The comparison results for each risk management aspect are listed and briefly discussed below. They are also summarized in Table 6.1.

- **Risk definition:** The definition of risk is the same in risk management and agile models. However, the recognition of risk as an opportunity is not always as explicit in the risk management standards as in the agile models.

- **Risk assessment:** Risk management is aimed at mitigating risks and thus provides well established techniques for identifying and assessing risk. Scrum and XP do not provide any guidance at all. In this respect, the agile models can learn from risk management models to ensure effective risk management.

- **Software lifecycle:** None of the risk management models studied encompasses management of risks within other software lifecycle processes than development. The iterative, incremental and evolutionary approach in agile development means that in each iteration, enhancements, corrections and minor improvements are being made in parallel. Hence, the agile models cover to some extent development, enhancements and problem resolution in one and the same iteration. In this respect, one could argue that they encompass other lifecycle processes than development. However, the management of risk is not made an explicit part of the product lifecycle. We believe that the agile approaches should consider risk over the entire lifecycle.

- **Stakeholders, roles and responsibilities:** Both XP and Scrum suggest that the team shares the responsibility for project success. They do not
however suggest particular guidance for the roles that might be relevant in the risk management process. In our opinion, they should explicitly identify and recognize all the stakeholders, roles and their responsibilities to ensure that all the risk sources and targets have been identified and scrutinized from all possible perspectives.

- **Template:** The agile models do not suggest templates for communicating information about risks and their management, despite the fact that high risk information quality is one of the most important prerequisites for effective risk mitigation (Carr et al. 1993). In our opinion, agile process models should be complemented with guidance for what information to collect and how to structure it.

- **Supporting repository/tool:** Neither Scrum nor XP prescribe any recording of risk information. Generally, risk management models advocate permanent storage whereas agile models advocate temporary one. Risk management models suggest a risk management repository and experience base, supported by electronic tools. The agile models, on the other hand, mainly prescribe the informative workspace. From a risk management perspective, permanent recording of risk information is the only way of assuring that all important information is being remembered, paid heed

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Present</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk definition</td>
<td>Yes</td>
<td>Risk is also recognized as an opportunity</td>
</tr>
<tr>
<td>Risk assessment</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Software lifecycle</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Stakeholders, roles and responsibilities</td>
<td>Partly</td>
<td>The agile team shares the responsibility for the project success, but there is no recognition of its implications for the management of risk.</td>
</tr>
<tr>
<td>Templates</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Repository/tool</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Product status</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Environment</td>
<td>Partly</td>
<td>Although some guidelines are provided for distributed agile projects, risk management aspects are not included.</td>
</tr>
<tr>
<td>Organization</td>
<td>Partly</td>
<td>The XP principle covers organizational aspects on the individual and team level, but not so much on the organizational level (e.g. maturity) and risk management is not included.</td>
</tr>
<tr>
<td>Measures</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
to and that lessons learned can be easily disseminated to enable process improvement (IEEE1540, 2001). In this respect, we suggest that agile methods should consider the idea of permanent storage as a complement.

- **Product status**: None of the agile models studied relates product status to specific risk management measures. This is an important factor to consider when tailoring risk management process to specific product status and allocating resources to it.

- **Environment**: The agile models studied provide some practical guidelines on the environmental aspects and physical context of development that could be used also for risk management, for instance the XP practices of *Sit together and Informative Workspace* (Beck, 2004). However, they do not consider distributed environments. Agile development models have spread to distributed environments, but more evidence is needed from a risk management perspective.

- **Organization**: Regarding guidance for managing organizational issues such as attitudes, training, and maturity, XP provides relatively detailed guidance on how to deal with teambuilding, training and competence development for instance by practices such as *Whole team* and *Pair Programming* and by principles such as *Diversity* and *Failure* (Beck, 2004). We believe these could be useful also for risk management. However, they do not address risk management in particular. Hence, the agile models should be more active in recognizing the importance of these organizational issues also for risk management purposes. It is only in this way, one may make sure that risk management is implemented and run in an effective way.

- **Measures**: The risk management models studied suggest integration of risk management with measurement processes. However, they do not suggest any metrics or measurement models for actually measuring risk management nor do they provide guidance on how to integrate the processes. The agile models integrate measurement processes in their own specific way. Scrum uses empirical process control, implying that one continuously inspects the process. Hence, our conclusion is that agile models seem to integrate measurement models. However, concerning the integration with other organizational processes, such as risk management, there are no guidelines. We suggest that the agile models complement their models with suggestions for how to integrate with other organizational processes to ensure useful and valuable communication and feedback on information needed in the entire organization.
In addition, neither XP nor Scrum explicitly identifies various risk management phases. We believe that this is a serious omission. A phase such as the Risk Sign-Off phase is pivotal for making sure that serious risks have been attended to. Lack of this aspect could also lead to serious legal consequences.

As summarized in Table 6.1, the results of this comparison of the agile and risk management processes clearly indicate that agile development models implement few risk management practices. One can conclude that a lot remains to be done to make the agile models more risk-driven.

In the following, we will use these results and the fundamental risk management aspects listed here as a starting point for reasoning about the integration of the agile and risk management models. We start by studying the possible ways of integration and then continue by proposing a model for integrating them.

6.2 Creation of integration model

By definition, an integrated process is “a series of interrelated activities that share and exchange data to achieve a common purpose” (Byrnes, 2006). Hence, in this context, integration is the activity of connecting the agile and risk management processes and making them communicate with each other to support the goal of organization-wide management of risks in the agile process.

Integrating risk management with agile development requires the integration of two different types of processes, management (risk management) and engineering (agile). The process integration endeavor has been undertaken before, for instance, within the enterprise architecture and business process modeling fields (Zachman, 1987). However, very little has been done regarding the integration of processes from the two disciplines studied here, except for a few attempts (Armenta and Gaono, 2008). More importantly, there is no general integration method or process satisfying our purposes. It has also been stated that the integration of management and engineering processes needs further clarification (Chroust and Hardt, 1996). For these reasons, we need to create our own integration model.

By studying various agile process scenarios in which risk management was executed in industry, we were able to identify several integration points between the two processes. By integration point, we mean anything that explicitly connects the two processes and realizes an exchange between them supporting the risk management goals of the organization.
An integration point may involve (a) means of exchange, such as people, tools and documents, (b) activities, such as communication, and (c) cross-organizational flows, such as process flows, work flows and data flows.

Because there were no general models for integrating the processes of interest here, we started with the most rudimentary set of integration points as identified in our studies of integration in industry. Using them, we created a simple integration model describing four basic integration points (see Figure 6.3). We call them (1) Organizational Levels and Process Phases, (2) Roles and Responsibilities, (3) Communication Channels, and (4) Process Aspects.

### 6.3 Integration model

In this section, we describe the integration model. It consists of four basic integration points. As shown in Figure 6.3, they are (1) Organizational Levels and Process Phases, (2) Roles and Responsibilities, (3) Communication Channels, and (4) Process Aspects. Below, we briefly describe them and explain how we intend them to be used in the integration of the risk management and agile processes.

#### 6.3.1 Process phases and organizational levels

A critical integration point involves the identification of when and where risk management takes place in the development process. The agile development process consists of several phases and it spans over several organizational levels. By finding out where risk management occurs clarifies the various points of integration between the two processes. Hence, one needs to map out where the risk management takes place in the agile process. It is a prerequisite for further integration.
6.3.2 Roles and responsibilities
In the context of process integration, roles comprise a primary crossing point for the exchange of information between the two processes. According to standard risk management practice, risks should also be owned by various roles to ensure that they are effectively managed (PMI, 2004). Hence, to establish productive risk management, one needs to identify appropriate roles and their responsibilities in the integrated process. A role is not always equivalent to one specific person. One person may have several roles and one role may be assigned to several people.

6.3.3 Communication channels
Agile development and risk management are communication intensive processes. The communication gets intensified if risks are managed across the whole organization and several processes and process phases. To achieve effective communication, one needs designate appropriate communication channels. Such channels integrate the processes and their phases by specifying the flow of communication on an organization-wide level.

6.3.4 Process aspects
Processes are variable and dynamic. Their instances vary due to many different factors affecting the process design. Hence, such factors need be considered. In this integration model, they are represented by the fundamental aspects of risk management as discussed in Section 6.1 and listed in Figure 6.1. Each of these determines the magnitude of risk management required within the integrated process, thus aiding in adapting an instance of the integrated model to the specific situation at hand. Hence, they constitute guidelines for making decisions on how to adapt the process according to the needs of risk management. The following aspects should be considered when tailoring instances of the integrated process:

- **Risk definition**: constitutes a control that there exists a comprehensible risk definition to facilitate the communication of risk within the organization.

- **Risk assessment**: defines a control that there are guidelines for assessing risks effectively.

- **Software lifecycle**: defines a control for designating adequate risk management activities according to the varying need of different software lifecycle phases.
• **Stakeholders, roles and responsibilities:** defines a control for streamlining the degree of involvement of various stakeholders, roles and their responsibilities as determined by factors such as project size and project risk profile.

• **Supporting tools/repositories:** constitutes as control for considering the need of tools to support effective communication of risk and risk management within an organization.

• **Template:** constitutes a control for identifying the degree of formality of templates needed for recording risk and its management effectively.

• **Product status:** defines a control for determining the amount of risk management needed with respect to the product’s quality, business value and life expectancy.

• **Environment:** constitutes a control for considering how to adopt risk management with regard to the project’s cultural, social, political and physical context.

• **Organization:** defines a control for considering organizational aspects such as people’s attitudes towards risk, organizational maturity, competency and training and their impact on a risk management program.

• **Measures:** constitutes a control for determining the need for integrating the risk management process with other organizational processes, such as measurement processes, to provide useful information and feedback to the organization.

This summarizes our integration model and its four components. It is used as a basis for outlining an integrated model described in the next section.

### 6.4 Integrated model

In the following, we present the integrated model. It is based on the knowledge gained from studying various agile process scenarios in which risk management was executed at eTV, a Swedish software organization. We start with an overview of the outline of integrated model. We then describe it in practice using eTV as our example.

The integrated model is illustrated in Figure 6.4. Note that we use the synthesized agile and risk management models as the basic constituents of the integrated model (see Chapter 5 and Figure 5.1 and 5.2).
6.4.1 Outline of integrated model

The integrated model manages all risks that are encountered within the agile development process on an organization-wide basis. As depicted in Figure 6.4, the agile process covers three main development phases, Product Vision Planning, Product Roadmap and Release Planning, and Implementation.

Most of the risks undergo a complete risk management process within these development phases as mapped out in Figure 6.4. The risk management process consists of the six phases of Risk Identification, Risk Analysis, Risk Management Planning, Risk Monitoring and Control, Risk Sign-Off and Risk Post-Mortem Analysis. Some of the risks, however, may not be fully managed or mitigated within a single phase. In this case, they may have to be transferred to the next phase, and/or get reported to an organizational function called Risk Management Forum (RMF). This transfer is represented by the double-edged arrows between the three agile development phases and the box representing the RMF on the left-hand side of Figure 6.4.

The Risk Management Forum is a function for coordinating risk management across the organization. It manages any risks that cannot be managed within a certain development phase and/or may have impact on other parts of the organization, and which promptly have to be disseminated to all the concerned parties, for instance other teams or organizational units. It consists of a cross-functional group represented by the roles responsible for or concerned with or otherwise capable of managing organizations-wide risks.
In the following, we provide a brief overview of the outline of the integrated model. It is described according to the integration model and its four integration points: (1) Process Phases and Organizational Levels, (2) Roles and Responsibilities, (3) Communication Channels, and (4) Process Aspects. A more detailed description of them, including the mapping of risk management as prescribed by the Organizational Levels and Process Phases integration point, is provided in Section 6.5 where we present an industrial example of the integrated model.

6.4.2 Process phases and organizational levels

The integrated model covers the entire agile development process. This involves integration of risk management on two organizational levels: the Business and Engineering levels (see synthesized agile model in Figure 5.1 in Chapter 5). The Business Level consists of the Product Vision Planning phase. The Engineering Level consists of the Product Roadmap and Release Planning and Implementation phases. The agile phases and the management of risks in each phase are briefly outlined below.

- **Product Vision Planning**: The Product Vision Planning phase is the first phase in the synthesized agile model (see Figure 5.1). It involves creating a product vision plan describing the product goals, overall business and product structure and return on investment. The resulting product vision plan guides the work carried out in subsequent planning, decision making, and development (Scrum, 2003). Risk management within this phase mainly concerns the identification and analysis of business related risks, such as budget and resource risks.

- **Product Roadmap and Release Planning**: The Product Roadmap and Release Planning phase is the second development phase in the synthesized agile model. Here, one first creates a high-level roadmap plan for the identified product releases which one then regularly revisits for more detailed planning before each release starts (Scrum, 2003). The risk management in the Product Roadmap and Release Planning phase comprises risk identification, risk analysis and risk action planning. It involves both business and technical risks.

- **Implementation**: The third agile development phase is the Implementation phase. In this phase, the team, product management and other relevant stakeholders meet to plan the work to be conducted in the coming iteration. The plan is then executed in the iteration to deliver an increment of working product functionality (Beck, 2004). Risk management in the Implementation phase primarily covers the monitoring and controlling of the risks that has been transferred to this phase from previous development
phases. New risks are also continuously identified, analyzed and planned for during this phase. Risks are mainly of technical character.

6.4.3 Roles and responsibilities

In the integrated model, we have identified several roles having various responsibilities with respect to risk management and its communication within and between the various agile process phases. Generally, however, the risks are owned by the roles in the phase where the risk is originally identified. Below, we describe the roles and their responsibilities.

- **RMF Members** own all the organization-wide risks. Their main task is to supervise and coordinate all the organization-wide risks and make decisions on them. However, they may delegate their management to other roles either within the Business or Engineering levels or both.

- **Business Manager** is responsible for managing risks at the Business Level. This role owns all the risks relevant for this level. However, he may delegate their management to the roles in the Product Roadmap and Release Planning phase or Implementation phase. The choice depends on the character of the risk and where in the organization it is most adequately managed. Still however, he keeps the risk ownership till the delegated risks get mitigated.

- **Product Manager** is responsible for all the risks managed in the Product Roadmap and Release Planning phase. This role owns all the risks relevant for this level. However, similarly to the business manager, he may delegate their management to other roles in the organization, if needed. He still keeps the risk ownership till the delegated risks get mitigated.

- **Team Leader** and **Team Members** are responsible for managing risks within the Implementation phase. The team leader supervises the risk management within this phase. Usually, team members own the risks that concern the development tasks assigned to them. However, same as with the risk owners in the other phases, the team may also delegate risks to others in the organization depending on the risk and risk management needs.

6.4.4 Communication channels

An effective management of organization-wide risks rests on how risks are communicated within an organization. This is because, in the integrated model, risk management takes place throughout the entire development process. Hence, to warrant effective information flow, one needs define
communication channels. As depicted by the double-edged arrows in Figure 6.4, the integrated model identifies six main two-way Communication Channels. The difference between them concerns the different roles in each phase who act as the main senders and receivers of risk information. The channels are:

- **Product Vision Planning - RMF:**
  1. The business manager reports on critical risks to the RMF for organization-wide attention.
  2. The RMF identifies risks and delegates their management to the business manager.

- **Product Vision Planning - Product Roadmap and Release Planning:**
  1. The business manager reports or delegates the management of the risks identified within the Product Vision Planning phase to the product manager, who owns the risks in the Product Roadmap and Release Planning phase.
  2. The product manager within Product Roadmap and Release Planning reports on the management of the risks to the business manager. He reports on new risks that are of relevance to the Product Vision Planning phase.

- **Product Roadmap and Release Planning - RMF:**
  1. The product manager reports on the identified risks in the Product Roadmap and Release Planning phase to the RMF for organization-wide attention.
  2. The RMF identifies risks and delegates their management to the product manager within the Product Roadmap and Release Planning phase.

- **Product Roadmap and Release Planning - Implementation:**
  1. The product manager reports on or delegates the management of risks identified in the Product Roadmap and Release Planning phase to the team leader in the Implementation phase.
  2. The team leader in the Implementation phase continuously reports on the management of the risks delegated to the team by the product manager. The team reports on new risks that are of relevance for the product manager.

- **Product Vision Planning - Implementation:**
  1. The business manager reports or delegates risks identified in the Product Vision Planning phase to the team leader, who manages them in the Implementation phase. These risks of generally of technical character.
2. The team leader in the Implementation phase reports on the management of the risks delegated to the team by the business manager. The team also reports on any new risks of relevance to the business manager.

- Implementation - RMF:
  1. The team leader reports on the risks identified in the Implementation phase to the RMF for organization-wide attention.
  2. The RMF identifies a risk and delegates its management to the team leader in the Implementation phase.

### 6.4.5 Process aspects

Our industrial study has helped us to identify the impact of Process Aspects on the integrated model. This impact is materialized in the following guidance for adapting risk management in the integrated process:

- **Risk definition is a main prerequisite for identifying risks and deciding when to use the risk management process to control risk:** It is of high relevance when making decisions on when to perform risk management and when to incorporate risk management into software development.

- **Risk assessment results identify pertinent actions for performing risk-driven development:** The risk classification and assessment techniques are important for knowing when and how to manage risks effectively in subsequent process phases. Hence, guidelines for assessing risks are needed.

- **The product’s software lifecycle stage aids in designating appropriate risk management actions:** The portfolio of risk types and risk management activities differs considerably depending on whether a project concerns new development or maintenance of a legacy system. In early lifecycle phases, the business risks often get a higher priority value, for instance due to time-to-market concerns, whereas technical risk may be of greater importance in the maintenance phase. This means that the process is adapted with regard to the prevailing risk types and their prioritization.

- **Designation of and coverage of stakeholders, roles and responsibilities are determined by factors such as project risk profile, project type and size:** The coverage of stakeholder roles and the degree of their involvement depends on the type of project, size, the risk complexity, risk scope and the risk criticality. In small agile projects, too many roles managing minor risks can lead to too much coordination and overhead. In more
complex cases, one needs to introduce more formality to make sure that the risks get proper attention

- **The use of supporting tools and repositories are determined by factors such as project risk profile, project type, project size and team distribution:** The relevance of tools and repositories when adopting risk management depends on the size of the organization, the project needs and the project risk profile.

- **The degree of recording formality varies with respect to project risk profile, project type, size, and team distribution:** Templates provide relevant support for describing and communicating risks. However, the level of recording information in them varies with respect to project type, project size, team distribution and risk severity. In agile projects tackling low severity risks, templates should always be kept very simple. The degree of recording formality increases as soon as the risk criticality and/or project or development organization grows.

- **Product status, life expectancy and business value help determining the amount of risk management process needed:** The amount of attention put into the risk management process varies greatly depending on the product status, its life expectancy and business value. For instance, systems close to retirement do not get as much attention from the risk management perspective as systems that are developed from scratch. The reason is that risks and their criticality generally vary with the status and business value of the product.

- **Environment and the project’s physical context determine the formality of the risk management process:** Systems that are developed in a non-distributed environment where the team members and the customer representative (product owner) are co-located and work closely together do not generally need to coordinate risk management in a formal way. However, if the organization is large or on the way to expand, generally require a more formal and conventional risk management process as distributed environments will imply more coordination.

- **Organizational maturity and training aids in adopting a risk management program successfully:** Organizational maturity, such as people’s attitude towards risks, competency, and capability to perform risk management are of relevance for successful adoption of risk management. For instance, training should be part of the risk management program in organizations where the risk management awareness or knowledge is low.
Software development need be integrated with other organizational processes: The development and risk management processes need be integrated to provide useful feedback to the organization, especially in larger organizations. Hence, one needs guidelines for making such integration explicit.

In the following, we describe the integrated model using the example of eTV.

6.5 Integrated model at eTV

In this section, we describe an industrial example of organization-wide risk management in an agile project at eTV, a Swedish software organization.

In brief, the studied organization, here called eTV due to confidentiality reasons, develops a web-based system for broadcasting TV programs on the Internet. It has 35 employees. The development team consists of eight people, including a product owner, a scrum master and six developers. The development process they use is Scrum combined with the practices of Informative Workspace, Stories, Incremental Development, Test-Driven development, Continuous Integration and Pair Programming in eXtreme Programming.

At eTV, all organization-wide risks are managed by the Architectural Forum (AF). The AF corresponds to the RMF in our model. Risk management comprises a critical part of its function. The purpose is to effectively coordinate risks that need attention on an organization-wide basis. For instance, the AF may coordinate the management of a critical technical risk identified in the Implementation phase having impact on the release schedule, and consequently, on the planned system integration and delivery.

The AF consists of a group of roles that are represented by key personnel, such as business and product managers, team leaders, architects, and custom-
ers. They meet regularly on both an incident-driven and periodic basis (every three weeks).

As depicted in Figure 6.5, The AF has its own risk management process. Some risks may be managed in their entirety within the forum. Some other risks may have to be delegated and managed in co-operation with the appropriate role(s) active in the phase relevant for the risk. However, the AF keeps the principal ownership of these risks throughout their whole management. It has the utmost authority to sign them off.

Regarding risk management in the Product Vision Planning phase, it mainly concerns business related risks. It is managed by business management. As depicted by the two arrows in Figure 6.6, the risks that do not get fully mitigated at this level undergo at least the first two risk management phases, Risk Identification and Risk Analysis. These risks will be further analyzed and managed in the subsequent development phases. This means that they will get transferred to the Product Roadmap and Release Planning phase. Some of them, however, may be reported to the Architectural Forum, who then re-analyses them in order to determine further measures.

The risks managed in the Product Roadmap and Release Planning phase initially comprise business risks. All risks in this phase are owned and managed by the product manager. Most of the business risks at this level correspond to the risks that have been identified in and transferred from the initial Product Vision Planning phase.

As illustrated by the different style arrows in Figure 6.7, the mapping of risk management in the agile process shows that the risks that cannot be mitigated in the Product Roadmap Planning or Release Planning sub-phases mainly undergo Risk Identification and Risk Analysis. In addition, the Release Planning phase also involves Risk Management Planning for determining the strategies and actions needed to mitigate the identified risks for that release.

As the high-level product vision plan evolves into the product roadmap plan and finally into more detailed release plans, new risks may be identified.
analyzed and planned for in this phase. The result is materialized in a risk list and a risk management plan, supervised by the product manager. At this stage, risks are still mainly of business character but technical risks may also be identified here.

The risks are delegated to the appropriate candidates within the organization, if needed. However, the product manager still keeps the main ownership of the risks and is the only authority to sign-off the mitigated risks. Hence, the resolution of the delegated risks must be reported to the product owner.

The product manager may also transfer the identified risks to the Architectural Forum for further action. The transferred risks, for instance, may concern schedule risks that should be further coordinated and communicated to others in the organization. The risks of technical nature, such as issues concerning the implementation of certain system functionality, get transferred to the Implementation phase, where they are further analyzed, planned for, monitored and controlled by the team and its members.

The Implementation phase is performed by a team leader together with team members. In this phase, risk management strongly varies. Some risks may be fully mitigated within a daily meeting, some will be mitigated within an iteration, whereas other risks will have to be transferred to the next iteration for continued management. Despite this variety, we have identified pattern of managing risks within the Implementation phase. As illustrated by the different style arrows in Figure 6.8, this pattern is as follows: Iteration Planning mainly involves Risk Identification, Risk Analysis and Risk Management Planning. Risks are monitored and controlled during Daily Status and Development in the Iteration Implementation phase. New risks may also be identified during Daily Status, which are then analyzed and planned for. The
Iteration Completion phase involves Risk Sign-Off and Risk Post-Mortem Analysis. Below, we elaborate on this.

Risk management is continuously performed within an iteration. Risk monitoring, control and treatment are mainly conducted within the development activities. Risk monitoring, however, takes place via the so-called “checkpoints”, where one actively inspects the status of risks. As illustrated in the Implementation phase of Figure 6.8, the main checkpoints are (1) Iteration Scoping, (2) Task Planning, (3) Daily Status, (4) Development and (3) Iteration Completion.

During the Iteration Scoping and Task Planning sub-phases, one studies the known risks and identifies the new ones. After having identified the new risks, one then analyzes both the new and old risks, and makes appropriate decisions. Some of the risks may already have become mitigated during the Iteration Planning. In this case, they are checked off from the risk list and the risk management process is terminated for those risks. For the remaining risks, one creates action plans.

Once the risks have been identified and analyzed, one determines the iteration plan. This plan also includes a list of risks to be monitored and controlled.

Usually, one risk concerns one task or feature to be implemented. Hence, the risk ownership is automatically assigned to each programmer or programmer pair being responsible for that task or feature. In addition, the team leader has a supervising ownership for all the risks identified. The team
leader has also the utmost responsibility for all the remaining risks that cannot be assigned to the team members.

During the iteration, risks are monitored in the *Daily Status* phase. In this phase, one discusses the known risks and identifies the new ones. Regarding the known risks, their status is communicated to the whole team on a daily basis. Regarding the new risks, they are mainly identified during the daily meetings. Some minor risks may even be analyzed and planned for, or even mitigated, during a meeting. For more serious risks, however, one arranges special sessions or new meetings, during which one analyzes risks and determines what needs to be done to mitigate them.

Between the daily meetings, the risks are monitored and controlled by the risk owners in the *Development* phase. In addition, they are continuously supervised by the team leader.

In the final checkpoint, that is, the *Iteration Completion* phase, one evaluates the product, the process and the risks. Regarding the risks, one goes through the list of all the risks and analyses how they have been monitored and controlled, whether they have been mitigated, and one identifies the effects of the risk management process. In this phase, all the mitigated risks get signed off and removed from the risk list. The business risks get signed off by the product owner, whereas all the other risks get signed off by the team. Regarding the unmitigated risks, they are still on the risk list. This list constitutes an input to the next iteration.

As a final remark, eTV is in the process of scaling up the agile project with more teams. This implies greater communication burden among the teams, the various process levels and the *Architectural Forum*. Regarding the communication among the teams, the organization is in the process of creating daily inter-team meetings corresponding to the *Scrum of Scrum* meetings (see Figure 6.9) (Scrum, 2003). During these meetings, the team

---

**Figure 6.9** Scaling up the integrated process with Scrum of Scrum meetings at eTV.
leaders meet to discuss status between the teams on a daily basis. If any major risks are identified, then they will have to be reported to and managed by the Architectural Forum.

6.6 Evaluation of integration solution

In this section, we present the results of an evaluation of the proposed solution to address the lack of risk management in agile development. The evaluation is based on in-depth interviews that were conducted with ten agile practitioners in the Swedish software industry. Their profiles are presented in Chapter 2, section 2.4. The questionnaire is presented in Appendix L. The entire evaluation is also presented in Paper 8.

Below, we briefly describe the evaluation criteria. This is followed by a summary and discussion of the evaluation results.

6.6.1 Evaluation criteria

The overall purpose was to evaluate whether the proposed solution including the integration model and the integrated model provide a valid solution for addressing the lack of risk management in agile development. To accomplish this, we evaluated the proposal against a set of requirements covering the integration model, integrated model and the question of agility. As listed in Chapter 1, the solution is expected to fulfill the following three requirements:

- The integration model should provide software organizations practical guidance on how to integrate their risk management and agile development processes.
- The integrated model should provide a reference model for software organizations to examine their risk management and see how they compare to our model.
- The solution should provide guidance for reasoning about agility with respect to risk management

In the following section we discuss the results of the evaluation of the proposed integration solution.
6.6.2 Evaluation results

Generally, our evaluation results show that all the three requirements are fulfilled. Hence, we conclude that integrating risk management and agile development provides a valid solution to address the lack of risk management in the agile model. However, our solution is considered valid only under circumstances where the inherent risk-driven nature of the agile development model is insufficient.

The need of risk management depends on several factors, but primarily it subsumes to project size, type of development, product complexity and overall project risk profile. More specifically, the results show that the integration of risk management adds most value in development contexts with one or several of the following characteristics:

- teams consisting of more than ten people
- distributed teams
- development of safety critical, security critical, embedded systems, complex systems, entirely new systems and/or innovative systems
- projects with high risk exposure, e.g. depending on factors such as the external environment, organization, product status, technology, schedule, and so forth.

The integration of risk management was not considered necessary in smaller projects developing simple software where the team was co-located. In these cases, it was argued that the agile process itself allowed effective management of risks without adding risk management to it. Below, we discuss the results of each of the three requirements, starting with the integration model.

Integration model

Regarding the proposed integration model, the interviewees agreed that it provided practical guidance for how to conduct integration. The four integration points were also considered relevant and useful. Hence, we do not modify the integration model. However, some concerns were raised regarding the usefulness of Communication Channels. Defining the flow of risk communication between various parts of the organization, was not considered very useful unless guidelines for how to communicate risks in an agile way were provided. We consider this a very important feedback for extending our model.

A similar comment concerns the Process Aspects. Several interviewees requested that our proposal should include practical examples of its impact on the integrated process as well guidance on how to use them in the design
process. We consider this a very important feedback for extending our model.

**Integrated model**

The integrated model was considered to fulfill its purpose as a reference model. It was regarded useful for anybody interested in comparing their risk management practice. However, without providing guidelines on how to actually conduct agile risk management, its utility was considered limited. Hence, it was suggested that the integrated model was extended with guidelines on how to conduct agile risk management. This is also considered a very important advice for improving the model.

Some other concerns raised involved the RMF function and the roles in the integrated model. They were considered to conflict with agile principles. We consider this feedback as highly relevant. However, we do not modify our model due to this observation. We believe that they should be an option and not a must, as far as other forms of achieving good communication, collaboration and shared project responsibility for risk management on a continuous, organization-wide basis are practiced. For instance, the RMF could be part of already existing cross-organizational functions in an organization such as the *Integrating Scrums* as suggested by Schwaber (2007).

**Maintaining agility**

Our model provides a solution for extending the risk management capability of the agile model without changing the agile base. It also provides a simple tool for integrating the risk management process according to the needs at hand and a reference model for comparing notes. For these reasons, the interviewees agreed that agility would be maintained. However, it was also suggested how agility could be further enforced. For instance, it was proposed that any direct conflicts with agile principles, such as collaboration, were removed or further motivated. The reason argued was that adding any activities to the agile process must always be clearly justified.

In response to these comments, we wish to emphasize that none of the current agile models explicitly describe how to extend the agile model with risk management, or other processes. To fill in this gap, we therefore keep our model as originally described until other solutions are provided. We also lay emphasis on the fact that the model is made tailorable to allow the user to maintain any degree of agility. In addition, we clarify that the model is not aimed at all projects. The projects and organizations primarily concerned by our solution are those with the characteristics listed above, such as projects with distributed teams or developing complex software, or which need to extend the risk management capability of their agile models for any other reasons not detected herein.
6.7 Final remarks

In this chapter, we have examined the agile model from a risk management perspective, identified and discussed the gap that exists between them, and proposed and evaluated a solution to reduce it. The solution that we propose subsumes the following contributions:

- **Simple integration model.** Agile and risk management represent different types of processes. Because there are no general models for integrating them, we introduce a simple integration model enabling their integration. It is created based on empirical studies of various agile process scenarios in the industry in which risk management was executed. It identifies four basic points of process integration including a list of aspects that help determining how to adapt an instance of the integrated model according to the risk management needs in every specific situation. Hence, the integration model consists of four components facilitating the design of an integrated model, (1) a component for mapping out the risk management process phases in the agile process, (2) a component for defining the roles and their responsibilities in the integrated model, (3) a component for defining the communication structure of the integrated process via communication channels, and (4) a component listing the factors that have an impact on the design of specific instances of the integrated process also allowing one to reason about the impact of integration on agility.

- **Integrated model.** Despite the fact that risk management is of crucial importance for software project success, very few models have been found that explicitly integrate risk management with agile development processes. We outline an agile model integrating risk management. Its two basic constituents are represented by an agile and risk management model, both of which are synthesized from a subset of existing models in their respective domains. Hence, we do not bind the integrated model to a single standard. The integrated model aims at managing risks encountered within the development process and its inherent phases on two organizational levels, the Business and Engineering Levels. Hence, it recognizes the fact that risk management is a continuous organization-wide process. In addition, the integrated model is created using an inductive approach. It is based on empirical studies of risk management in agile processes in software organizations, thereby reflecting the industrial status of the integration of agile and risk management. Finally, the proposed model is also evaluated in industry. The evaluation resulted in suggestions for improving and extending the model. This feedback provides a basis for future work.
This chapter summarizes the essence of the main contributions of this thesis. Conclusions and suggestions of future work are presented in the next chapter.
7 Epilogue

Controlling risks improves essential software development features such as product quality, planning precision and cost-efficiency (Englund, 1997) (Ropponen and Lyytinen, 2000). For this reason, the inclusion of risk management in software development is an important factor to consider if one wishes to achieve project success (Kontio, 1999). Unfortunately, our research shows that many software development models, both the traditional and agile ones, are not well aligned with the risk management process practices.

In this thesis, we have addressed this by outlining a model specifically aimed at aligning the agile model with risk management. This is accomplished by integrating the agile and risk management processes. The proposed solution consists of an integration model providing guidelines for integrating the two processes and an integrated model, that is, a reference model against which organizations can compare their risk management practice. In its current version, it is primarily targeted towards process engineers and business developers or other roles involved in process engineering and process improvement.

The model has been elicited based on the results of studies of the status of risk management in interviews with representatives of totally 37 software organizations world-wide. The final outline of the solution was however elicited based on the study of risk management as executed in the agile process only within one software organization. The resulting model was also evaluated in interviews with ten agile practitioners. As a result of this evaluation, we conclude that: (a) it is a valid solution for addressing the current lack of risk management in agile development, however only in certain projects and organizations, (b) both the integration model and the integrated model needs to be further elaborated in terms of the guidance they provide, and (c) the model needs to be further investigated in terms of its applicability in practice.

The results show that the model places risk management on specific agile development phases. The integrated model also suggests preliminary patterns for where and when risk management takes place in the agile process. Also, in contrast to other existing models for including risk management in software development, such as the risk-driven spiral model (Boehm, 1988) and the approaches suggested by for instance Sliger (2006) and Li et al. (2006), our model recognizes risk management in development as an organi-
zation-wide activity. It places risk management over the entire software development lifecycle and it involves both the Business and Engineering organizational levels. The RMF and the Communication Channels adds structure to the process and help coordinate and control the organization-wide management of risks. Furthermore, it makes explicit that different roles are responsible for different types of risk as managed on different organizational levels and in different process phases. Finally, the Process Aspects aid in designing the instances of an integrated process with regard to the need of risk management, which is also relevant for reasoning about agility.

The model is however still in its infancy. It needs to be further elaborated and complemented with more details. Hence, the premises presented herein must be further validated. Despite this, we believe that it already provides (1) a platform for communicating about risk management within agile development, (2) a reference model for software organizations to examine their practices and see how they compare to the integrated model, (3) a starting point for researchers and industrial organizations to start defining their own integrated development models, and (4) a foundation for future work.

Finally, regarding the aspect of agility the solution is intended to maintain a level of agility adequate for the context at hand. Essentially, agile development and risk management both aim at the same thing, that is, helping development teams to do the right things in critical development situations. Hence, they should not be in conflict with each other. With our solution, we also argue that agility can be maintained and at the same time support the goals and practices of risk management.
References


Schwaber K. (2004), Agile Project Management with Scrum, Microsoft Press, Redmond, WA.

Schwaber K. (2007), The Enterprise and Scrum. Microsoft Press, Redmond, WA.


Xu R.Z. et al. (2005), Optimizing Software Process Based On Risk Assessment and Control. In proceedings of the 5th International Conference on Computer and Information Technology, China.

