Managing risks in Norwegian Agile Software Projects: Project Managers´ perspective

Lubna Siddique[#1], Bassam A. Hussein[2]

[#]*PhD scholar & Department of Informatics & University of Oslo, Norway*

Abstract: *The purpose of this study is to understand the role of project risk management in agile software projects. To achieve the purpose, we conducted a qualitative study. We conducted interviews with agile practitioners working with agile projects in Norway´s software industry. Grounded theory was used to analyse the data. This study aims to study the similarities and dissimilarities between the project risk management process in agile software projects and waterfall software projects, as well as identify the strengths and weaknesses in the current practices being used in agile software projects. Interview results suggested that risk management in agile projects is being done in two ways. One way is adopting implicit risk management strategies, which include communication and collaboration, shorter iterations, frequent delivery, early feedback, and delivering complex parts first. The other way is called explicit risk management strategies, which are relative estimates, burn down charts, SWOT analysis, and risk matrix. Limitations with implicit risk management strategies are also discussed. At the end, guidelines on how to maximize the impact of the risk management process on project outcome are also presented.*

Keywords — *Grounded theory, agile methods, risk management.*

I. INTRODUCTION

According to the 5th Edition of the PMBOK Guide [1], project risk is "an uncertain event or condition that, if it occurs, has a positive or negative effect on one or more project objectives such as scope, schedule, cost, or quality." As stated by Van Scoy [2], "Risk in itself is not bad; risk is essential to progress, and failure is often a key part of learning. But we must learn to balance the possible negative consequences of risk against the potential benefits of its associated opportunity" [2]. Need for risk management in agile projects is shown through the work of various authors [3] [4], [5]. Authors have also shown the importance of risk management for quality software delivery [4], [5]. Lack of risk management is shown to be a source of failure in projects [6].

Risk management is necessary because if it is left unattended it can have a negative impact on project objectives [7]. Risks become even more important to be identified and addressed if they create hurdles in meeting the success criteria (objectives, deliverables deadlines, cost, etc.) or significant resources are required to address them. According to Moran [8], if risks are not adequately identified or addressed, it can have following consequences:

As a result of inadequate information it becomes difficult "to make informed risk and reward decisions."

1. If risks are not identified, it is can result in "failure to identify appropriate risk response strategies based on risk exposure."
2. Consequence to the unavailability of risk related information it will cause "lack of oversight in risk monitoring leading to ineffective or inefficient treatment of risks."
3. Ignoring risks will lead to a "poor understanding of when to engage in risk activities."

The risk management process consists of the identification, analysis, control, and management of possible risks that may arise during the life cycle of a project [9]. In addition, this process also consists of a number of activities to collect information about events that can occur and affect project results [10, 11]. In agile methodologies, risk management is not defined explicitly [8] [38]. For example, scrum is a framework, but it does not describe how project risks should be managed [8]. Therefore, some kind of explicit risk management process needs to be in place in order to define and manage risks according to the needs of the projects.

Despite the fact that the risk management process is not defined explicitly in agile methods, possible risks are discussed at daily sprint planning meetings and retrospectives. Risk management in agile projects is also done by having fixed iterations with fixed deadline, providing support in changing requirements even late in the process, and maintaining close cooperation with the customer. This helps to have quick feedback to fix any deviations and bugs [12,13].

The purpose of the risk management process is to take into account all threats and opportunities that can come up during the project. This can help make decisions accordingly. Methodologies that don't have a risk management process implemented can miss the opportunity of making "informed risk and reward decisions" [8]. Appropriate risk response strategies cannot be planned if risk exposure is not determined, showing subsequent failure in the classification, prioritization, and designing of risk response [8]. Risk management in agile software

projects is shown to be one of the project success factors [13] [28] and "failure to engage in the monitoring of risk results in an inability to judge whether or not risks are being adequately managed" [8].

In this way, agile methodologies provide some kind of implicit risk management [4]. In scrum, at the start of each iteration user stories are prioritized to address the most important features. Before addressing the risks, it is very necessary to identify them first [5,14,15]. As agile processes do not provide a structured risk management process, they lack necessary activities such as risk identification, risk analysis, and mitigation plan [16].

Although significant research has been done in risk management in traditional projects, risk management in agile software projects requires further research [17] [18]. According to Odzaly and Des Greer [17], "Little work has been done to date on the role of risk management in agile methods." Albadarneh et al. also describe the need for research about risk management in agile software projects [18]. According to them, "While there is an extensive body of academic literature on risk management, few little researches have attempted to study risk management in agile development projects." This study is an attempt to address this research gap.

For the purpose of this study, we conducted 21 interviews. The goal behind this study is to:

1. Understand the role of project risk management in agile projects.
2. Gain an overview over the similarities and dissimilarities between project the risk management process in agile projects and non-agile projects.
3. Identify the strengths and weaknesses in the current practices.
4. Identify guidelines on how to maximize the impact of the process on project outcome.

Following are interview questions, we asked our practitioners:

1. Can you please tell me about your background?
2. How many years of experience do you have working with agile projects?
3. Why risk management is necessary for agile projects?
4. What do you do regarding risk management in your project?
5. Do you think agile methods provide some kind of risk management in itself?'

Depending on the answer to question 4, follow-up questions were also asked. If they answered yes we asked them,

Can you explain how?

If they answered no to question 4, we asked them how do they do risk management in agile projects.

6. How risk management in agile projects is different in agile projects than in waterfall projects?
7. In your opinion, how risk management process in agile projects can be improved? Do you have any recommendations?

II. LITERATURE REVIEW

Tomanek and Juricek [38] describe that risks in Scrum and eXtreme Programming (XP) are implicitly managed by keeping smaller iterations, communication with the customer, customer involvement, continuous feedback, and prioritization of user stories based on business value, which also help to identify any kind of deviations and ultimately risks. Therefore, certain risks (related to cost, schedule, and quality) are automatically addressed in agile projects.

Another important point is that agile approaches use the term "impediments" instead of risks, which are discussed in sprint review meetings and sprint retrospectives [39].

In agile projects, instead of focusing on planning, risk focuses more on delivering real working software in small working parts after each iteration. Agile teams are self-directed, therefore, in agile methods, it is not only the project manager who is responsible for the risk related activities, but the team also has a responsibility.

Thomas [40] explained the ways in which risks are managed in agile projects implicitly [40]. According to him risk management is done through the following techniques [40]:

1. Agile Roles and Responsibilities: It means that the project manager and product owner must work in coordination and according to their responsibilities [40]. Defining clear roles and responsibilities is also a suggested strategy by Siddique and Hussein [27] in their study for handling conflicts related to roles and responsibilities in agile software projects.
2. Agile Change Management: "When the product owner asks the project manager to add a feature to the scope, they use requirements trade off to ensure the overall scope, and hence the total effort is unchanged. The scope changes but the overall scope is stable in terms of effort" [40].
3. Agile Project Planning: It "ensures that the high priority requirements are delivered first. The product owner is continuously expected to make priority calls and move the important items to the top of the list. Low priority items are put at the end of the plan and if the schedule doesn't have space these items are put out of the scope." Our findings related to developing and delivering the functionality of highest

business value first are in accordance with this.

According to Thomas [40], following risks are implicitly addressed in agile projects:

1. Risks related to time to market:

It takes months to get a working product in waterfall approaches while agile approaches to deliver a product more often; therefore, time to market is reduced. Another point is that in waterfall project, projects are delivered after a long period; therefore, all potential risks are required to be listed for the entire duration. He argues that agile doesn't require that type of risk management because of the short duration of sprints. The best way to discuss risks daily is by using daily stand-up meetings.

2. Risks related to budget:

In waterfall projects, budget risks are significantly high because delivery time is so far from the time of initial estimates; therefore, there are more budget risks in waterfall projects than in agile methods.

3. Risks related to cancellation cost:

If a waterfall project is cancelled because of some reason, there will not be any product delivered to the customer. On the contrary, if an agile project is cancelled, there are still some parts of the product that are delivered to the customer. Customers can still have some parts of the project or functionality delivered.

4. Risk related to scope creep:

Feature or scope creep means that the scope of the project increases. In other words, scope creep means an uncontrolled increase in requirements (not changing requirements). Missed requirements are added to the project without removing other requirements; this may result in scope crept. While in agile projects, a balance is kept in adding requirements by removing certain other requirements.

5. Risks related to requirements:

In waterfall, requirements are specified long ago, while in agile, requirements are specified for each sprint. There are fewer chances of errors related to requirements. Also, if some requirements are missed because of some reason, these can be added in the upcoming sprints.

It is easy to terminate a project when the customer feels that certain functionality is delivered and they want no more functionality to be developed or there are other constraints related to budget, time, etc.

6. Risks related to technology:

As a result of a large time frame between project specifications to delivery, there are greater risks regarding technology, because it can take several months to uncover the problems related to technology. In agile projects, frequent delivery and testing of functionality ensures that risks related to technology become evident early in the project.

According to Thomas [40], the following can be risks when using agile methods:

1. Agile can lead to inadequate design.
2. Agile can lead to silver-bullet syndrome.
3. Agile doesn´t address weak personnel.
4. Agile doesn´t address contractor failure.

Ylimannela [46] conducted empirical study and found different challenges while performing risk management in agile software projects. According to him, agile methods are meant to deliver working increments, therefore, finding a person who will be responsible for risk management for each sprint was a challenge.

Moran [8] argues that agile methods got attention in balancing the risk and delivery value through prioritization of tasks is "too limiting" and according to him, "Some of the risks, which a project must contend with are not inherent in the execution of specific tasks, but rather in the circumstances surrounding that execution and might otherwise be considered part of a project governance profile (i.e., the effective and efficient deployment of resources towards the achievement of the goals of the enterprise)." According to Odzaly and Des Greer [17], "Little work has been done to date on the role of risk management in agile methods."

Derfer [45] argues for the need of this explicit risk management and according to him, "Sprint retrospectives are the mechanism for identifying and mitigating risks or challenges to the team," but this is not enough for doing risk management. The reason for this, according to him, is that retrospectives are most focused on what happened in the previous sprint and don't discuss risks on the project level [45]. The author argues that during retrospective meeting not all the right people are present and team members are more focused on finding and discussing improvements in previous sprints [45]. Different models for integrating risk management in the agile process can be found in the studies of [12] [8], but the question is: how effective are these models when used in agile software projects and are these models of any use for agile practitioners. Although new data collection was made for the purpose of this study, findings of this study are aligned with our previous research findings that were conducted about risk management [19].

III. METHODOLOGY

We conducted 21 practitioners' interviews from 18 different software development organizations in Norway. These organizations vary from consulting organizations to in-house development organizations. The practitioners had many years of experience with the software industry ranging from 3 to 40 years. The interviewed practitioners were project managers. Most of the practitioners were using agile methodologies since its inception or before it got the name agile. The products and services offered by the practitioners' organizations include web-based applications, front and back-office applications, and software development services. We conducted semi-structured interviews through various mediums,

which include face to face, email, and Skype meetings. To take multiple issues into consideration, the interview questionnaire was designed to incorporate different issues related to the risk management process in agile software projects. We asked practitioners open-ended questions. The sampling technique we used for our study is called non-probability sampling [20]. Keeping suitability in mind for the research, we used purposive sampling. Deliberate contact was made with the participants who had relevant experience with the agile projects. We performed internet searches for the practitioners and after finding out their suitability with our research questions we requested them to participate in the study. We assured participants of their anonymity. We will refer interviewed practitioners with AP1-AP21. The interview duration was 30 to 60 minutes.

A. Grounded theory

The research method we choose for our research is grounded theory because:

1. Grounded theory is useful to understand the phenomenon undergoing in the current scenario [21]. Grounded theory tries to explain what's going on, "what is the main problem of the participants, and how are they trying to solve it" [22]. As the purpose of this study was to know how the risk management process is being done in agile software projects, Grounded theory was a suitable choice for this research.
2. Grounded theory is the most suitable research method for underexplored areas [23]. Risk management in agile software projects is an underexplored area [17] [18], therefore, grounded theory was the most suitable approach for our study.
3. Another reason for using Grounded theory study for this research was that we wanted our findings to be grounded in the data. Our data is based on project manager´s several years of experiences with software projects and agile software projects in particular. Grounded theory is a suitable approach for such research involving participants' opinions [24].

This study is a part of larger study [25]-[27] that implied Grounded theory study as a research method. This was another reason for implying Grounded theory in this study.

B. Data analysis

In grounded theory, data analysis is called coding. Coding by using systematic approach of data analysis helps in understanding the data [29]. Data analysis in grounded theory is a continuous process. Therefore, we started coding very early after conducting the first interview.

1) ***Open coding:*** The first step of data analysis in Grounded theory is called open coding [30], which the data was analysed in [30] [31]. Interview transcripts were analysed to find key points and a suitable code was assigned to these key points [32].

2) ***Constant comparison:*** Codes that emerged after open coding are compared with other codes emerged from the same interview transcript and with codes emerged from other transcripts to produce a higher level of abstraction called concepts. This is called the comparison method [21] [22]. This method was also repeated at the concepts level to produce a higher level of abstraction called categories.

3) ***Core category:*** Open coding ends with the emergence of core category [30]. One of the categories that emerged after open coding is selected as core category. Core category "accounts for a large portion of the variation in a pattern of behavior." Therefore, it shows the "main concern or problem" for the participants [30]. It is central to all categories and all categories can be linked to this category. Core category in this study is "risk management."

4) ***Selective coding:*** Selective coding is done for "only those variables [concepts or categories] that relate to the core variable [category] in sufficiently significant ways as to produce a parsimonious theory" [30], [33]. After the emergence of core category, selective coding was done, i.e. risk management.

5) ***Theoretical saturation:*** Coding stops when theoretical saturation is reached. It means that newly collected data didn't give any new codes and categories [21]. We stopped data collection when we felt that there were no new codes emerging from the collected data.

6) ***Memos:*** Memos are the researcher´s ideas, which are written down regarding categories. These are the flow of ideas, which later can be used to write research results. Memos play a very important role in Grounded theory research [33].

IV. RESULTS

Interview data was analyzed using Grounded theory. Grounded theory analysis suggested that risk management in agile software projects is being done in the following ways. To make this clear we have divided it into two categories. Table 1 presents the interview findings.

A. Implicit risk management strategies

Some risks are handled implicitly in agile projects. By implicit, we mean that the strategies of managing risks are inherently provided in agile methods. When we asked practitioners whether agile provides risk management or not, they asserted that agile methods provide risk management in certain ways. Practitioners believe that with agile methods

the need for rework is reduced. They believe that standup meetings in the mornings are a good way to discuss risks (called impediments in agile).

"Agile helps to decrease need of rework."__AP19

"Not in traditional way. [When teams] are communicating more it's easier to see risks and if you do standup in the morning … because one of the question is do you see anything that can prevent us from reaching goal if you do that and they see we see something; I think that's part of risk management, so if you follow up …, you should do I think that could prevent risk."__"__AP6

"In agile there is no definition of risk. The only thing agile does is you have standup each morning and there are questions to point out anything that might endanger the project. That's an informal and affective way to risk for the people of the team."__AP8

"In scrum, we don't exactly have risk management."__AP9

"I guess in daily scrum you are reporting impediments."__AP14

According to practitioners, another advantage of using agile methodology is that it is easy to make changes anytime during the project.

"You can work with risk all the time; therefore, it is easy with agile. In agile because you can make changes all the time. If something unexpected comes you can make changes and you don't need to have complicated change procedures. I think that's the big point."__AP15

"The only way risk management in agile projects is done is that you have standup each morning and there is question to point out anything that might endanger the project that's informal and affective way to risk for the people of the team."__AP8

"After every standup each morning. I will go through notes and see if there are some areas that really are risks".__AP11

Practitioners are using the following risk management strategies for managing risks implicitly in agile software projects.

1) ***Communication and close customer collaboration:***

Practitioners told us that they try to collaborate closely with customers and discuss all kinds of risks with them. They also ask customers for any kind of risks they feel can impose threats on the project. Practitioners believe that the agile philosophy that customer and supplier should work in collaboration helps to discuss any kind of risks and uncertainties in agile projects. If it is required, an appropriate action plan can be designed to address these risks. "I think communication with customers is a key feature. If we communicate on a daily basis, it's a lot easier to know risks and discover them as well. That may be the number one thing."__AP7

"We discuss that in every sprint planning, then we discuss what are known risks and what are potential risks in upcoming sprints. That's something we discuss on every sprint planning."__AP7

2) Early feedback:

Practitioners asserted that early feedback helps to manage risk related to deliverable quality and functionality. Practitioners believe that delivering frequent can help to get customer feedback along with building trust between supplier and customer. Customers can give feedback about deliverables quality, functionality, cost, etc. If a project becomes very expensive, customers have the option to shut down the project if it is stated in the contract. Even if the project ends in such an abrupt way, customers will still have some working parts of the project or part of functionality delivered.

3) Short iterations:

Another way practitioners are handling risks in agile projects is by keeping the iterations shorter. They believe that instead of delivering functionality in larger parts, delivering it in smaller parts can help to handle risks.

By keeping iterations shorter, the project manager and team can make sure that they work and deliver functionality to customers, because this smaller part can be delivered to customers to get feedback.

4) Prioritization:

Practitioners asserted that one way they use to handle risk is through prioritization. User stories with the highest business value are prioritized first. Prioritization in such a manner is very helpful to develop and deliver important functionality first. This helps to ensure that customers get the most important part of the software delivered first.

"In scrum, you have to decide which stories have the highest business value so you have to estimate those [risks].

From the start of the project in scrum, you have to do risk estimation and flag the stories with the highest risk. The product owner's job is to prioritize between business value and risk; they are given guidelines in scrum and Kanban."__AP4

"Also, by implementing the most difficult or high risk things early."__AP17

5) Frequent Delivery:

Practitioners believe that delivering frequent can help to manage risks to a certain extent. By delivering in iterations, customers can get the working software part and can give feedback to improve or change the software if needed.

6) Dealing with the complex part first:

Practitioners told us that they also prioritize tasks on the basis of complexity. Dividing the project into smaller parts and working with the complex parts of the project first helps to control risks related to complexity. It is easier to make functionality for smaller parts and get it tested. Therefore, practitioners try to work and complete the most complex task first followed by the next relatively less complex part.

"In this scrum team, the risk was part of the task; you estimate complexity and discuss risk. Most of the time when you get the project you do the risk assessment."__AP4

"Instead of delivering all the simple parts first, we can show off to the client that we can handle this. We do a lot of talking with the client and solving the complex issue first."__AP10

B. Explicit risk management strategies

Besides above mentioned implicit risk management strategies, practitioners are also using explicit risk management techniques for managing risks in agile software projects.

"The risk management process is done in a traditional way. Agile doesn't have project management techniques, but they need them."__AP12

These are relative estimates, SWOT analysis, burn down charts, and risk matrix.

1) Relative Estimates:

According to practitioners, risks related to estimates are handled by giving relative estimates. Relative estimates help to control risks that results because of inaccurate estimation.

2) SWOT Analysis:

SWOT analysis is another way for practitioners to find out the strengths, weaknesses, and threats in the project. According to practitioners, SWOT analysis can help to figure out threats in the projects and appropriate strategies can be designed in order to deal with threats pointed out through SWOT analysis.

3) Burn down chart:

One way that practitioners are using to manage risks are burn down charts. Burn down charts provide an effective way to track effort and schedule on a daily basis; therefore, according to respondents' risks related to effort and schedule can be mitigated with the help of the burn down chart.

4) Risk matrix:

According to practitioners, for smaller projects, an implicit way of risk management that agile methods are providing can be beneficial, but if a project is large and complex then it is necessary that traditional approaches of risk matrix should be used as defined in PMBOK [1].

Practitioners told us that they sit with the customer and brainstorm to find and list all the risks associated with the project. Brainstorming is used as a part of the risk identification step. These identified risks are listed and appropriate strategies are designed after finding the risks associated with the project.

"We do brainstorming and list risks in the form of matrix, then assess probability and impact."__AP4

"I have weekly reports about risks."

"You have to do some risk analysis anyway."

"We have been using risk matrix."__AP5

"I use risk management from traditional risk management."__AP8

"I normally do it using matrix."__AP15

"I always have some kind of matrix but on various levels. We make a risk matrix from risks that came through brainstorming with customers and the team. These risks are handled in the same way as in waterfall projects."__AP19

"We have it usually in the form of matrix."__AP20

Table 1: Summary of results (strategies of handling Risk management in agile software projects)

Risk management in agile software projects	Implicit risk management	Communication and close customer collaboration Early feedback Short iteration Prioritization Frequent Delivery Delivering complex parts first
	Explicit risk management	Relative estimates SWOT analysis Burndown chart Risk matrix

V. Discussion

Interview data suggested that risks in agile software projects are handled in the following ways: communication and collaboration, prioritization of user stories, short iterations, frequent delivery, early feedback and relative estimates, burn down charts, SWOT analysis, and risk matrix. We have further divided these risk management strategies into two categories. Table 1 presents a summary of these results. These are described below:

A. ***Implicit risk management strategies***

Some risks are handled implicitly in agile projects. By implicit we mean that these strategies of managing risks are inherently provided in agile methods. Some of these implicit risk management strategies practitioners are using managing risks that include: communication and collaboration, prioritization of user stories, short iterations, frequent delivery, and early feedback. This is in accordance with Cohn [36] who states that "the short iterations, single-minded focus on working software, heavy emphasis on automated tests, and frequent customer deliveries help teams avoid the biggest risk most projects face—that of eventually delivering nothing." Keeping the iterations shorter is also in accordance with Williams et al. They suggest keeping iterations shorter [37] because smaller iterations and frequent delivery can help to make a project successful.

Practitioners believe that frequent delivery helps to reduce certain risk. This is in accordance with Siddique and Hussein [25] who suggested frequent delivery options in agile projects are useful in order to build trust with the customer. Trust comes when customer can see that they are getting value for their

money, which in turn is helpful for making the customer and supplier relationship work. Frequent delivery also helps to assess success for each delivery, point out any deviations, and address them accordingly [26].

B. **Need for explicit risk management**

SWOT analysis, risk matrix and burn down charts are explicit risk management techniques that practitioners are using for managing risks in agile software projects. Practitioners asserted that risk management in agile software projects is not provided explicitly. According to Moran [8], "Agile methodologies don't provide explicit risk management related to identification, recording, or management of risks" [8]. Tomanek and Juricek [38] also state that scrum doesn't provide explicit risk management process.

Practitioners told us that brainstorming is done with teams and customers to come up with all possible risk in the project. After that, risk matrix is made and each of the risk is listed in it. An evaluation of each risk is done, which is a necessary step to assess its impact and probability of occurrence. At last, mitigation action is planned for highly probability risks. This process is the same as it is done in waterfall projects. This is in accordance with Smith, who states that risk management techniques are similar in waterfall and agile approaches. He suggests using agile methods to reduce any kind of risks. For this purpose, he suggests using dedicated and collocated teams. According to him, communication is the key to avoid any kind of conflicts. This is in accordance with Siddique and Hussein [27], who suggested using communication effectively to resolve all kinds of issues in agile software projects. To make the project successful, Smith suggests using the risk management process in the same way as it is done in waterfall projects [41].

Smith suggests the risk management process in agile projects should be done in such a way that it should implement risk management activities into iteration planning activities. Risk management practices should be made very simple so that it is easier and quicker to follow by all team members [41].

According to Nelson [42], although agile methodologies address risk management implicitly, important risk management activities are ignored.

According to Hubbard [43], risk management consists of a set of risk identification, risk assessment, and risk prioritization, assessing impact of risks and planning mitigation strategies. Without having explicit risk management, it is very likely to ignore important risks that can prove very dangerous for projects.

Lu and Ma [44] studied the need for risk management and they found that risk identification is a very important step and it is required to be performed effectively for effective risk management. Nyfjord [50] also suggested the necessity of obtaining information about risk explicitly by performing risk identification activity. They think that if this activity is not performed correctly, then effective risk mitigation plans cannot be performed effectively, which further leads to ineffective risk tracking and control. Nelson et al. [42] states:

"Managing risks explicitly, but with techniques that stay true to the spirit of agility, is a necessary next step to improve risk management in agile processes and increase the probability of successful projects."

As a result of the importance of risk management in agile software projects, risk management is described as one of the knowledge areas in Project Management Institute's (PMI) [1]. The need of explicit risk management in agile projects is emphasized by many authors. According to Derfer [45], agile methods XP, Scrum, and Kanban are helpful "at delivering a software in a more collaborative, transparent, and predictable fashion than traditional waterfall processes. However, agile practices, by themselves, are not sufficient to address the risks that impact most medium-to-large software projects." [45]

VI. RECOMMENDATIONS

Practitioners are using two types of strategies to handle risks in agile software projects. One we called implicit risk management strategies (communication and collaboration, shorter iterations, frequent delivery, early feedback, and delivering complex parts first) and explicit risk management strategies (relative estimates, burn down chart, SWOT analysis, and risk matrix).

It is clear from discussion that agile methods of delivering short iterations, close customer collaboration, and frequent deliveries can help to minimize risk to a certain extent. All of these activities can help to reduce risks that are associated with technology and the scope of projects. This is in accordance with Horvath [47], who thinks that agile methods have short iteration and risks are discussed in every sprint planning meeting and risk register is revaluated quite often usually at sprint planning meeting and retrospective meeting after each iteration. Using agile methods "can reduce a variety of risks related to budget, time to market, scope creep, requirements, and security. However, you'll still need to define a process to manage risks" [48].

Although some risks are handled by implicit risk management strategies that agile methods offer, there are a number of other risks that a project comes across during its life cycle. Therefore, explicit risk management is required for such projects.

Small and low risk projects can rely completely on agile methodologies for implicit risk management, but for large and complex projects explicit risk management has a vital role to play [19]. Although agile methods claim that they provide risk

management in a form of delivering multiple iterations or sprints, this doesn't mean that for a given project, these are the only risks, which are required to be managed at the iteration level. There are other kinds of risks that need to be managed at the project level.

As agile methods lack explicit risk management, writers suggest using risk management from traditional approaches [9] [38]. Risk management in the waterfall project is a comprehensive way to address all types of risks, so we suggest continuing its use in agile projects as well. Interviewed practitioners were also satisfied about using risk management activities the same way it is used in waterfall projects. It is evident from interview data and discussion that explicit risk management in agile projects is necessary and practitioners are doing it in the same way. Based on interview data and discussion we came up with following suggestions:

1. One suggestion is making risk matrix for all the identified risks. Risk matrix can help to manage risks. Risk matrix should be made and risk response strategies must be chosen for "avoiding, transferring, or mitigating risks" [47]. Positive risk strategies are to "exploit, share, or enhance" [47]. Our suggestion of explicit risk management is also in accordance with Lant [48], who thinks that implicit risk management will be effective if "only things that affected the outcome of the project were the decisions that the developers made to implement the solution," but because of the existence of the number of factors that can affect the project success, "explicit risk identification and management can further improve on the success rate of agile projects" [48].
2. Teams can use wall charts and post-it notes to make risks more visible to all stakeholders and is not only held by project manager [49], but team members must also contribute to make this process effective. The team´s experience with previous sprints and iterations can help them provide better estimates for upcoming sprints [49].
3. In traditional approaches, the project manager is the one responsible for all the risk management activities held in a project and creating a strategy for addressing the risks. But in the agile approach, the whole team can be made responsible for managing risk. Nelson et al. [42] suggested explicit ways of risk management in agile projects. According to him, as agile teams are self-organizing, every member of the team must be responsible for identifying and prioritization of risks. However, for facilitation of this work, they suggested to make a person responsible for managing risk called the "risk manager." They suggested to make a risk register to write risks and use small workshops for risk evaluation [53]. We also suggest making a risk register as an effective way of managing risks.

VII. EVALUATING A GROUNDED THEORY RESEARCH AND LIMITATIONS

A. Evaluating a grounded theory

The product of a Grounded theory is called grounded theory. According to Breckenridge [34], "The emergent grounded theory offers an integrated probability statement that is not intended to be verified as right or wrong, but instead has relevant applicability and modifiability within the substantive area." Suggested criteria for evaluating a grounded theory is the fit, work, relevance, and modifiability [30].

1) ***Fit:*** Codes, concepts, and categories must be validated in terms of their fit in the data. Glaser [30] suggests "the analyst's goal is to ground the fit of categories as close as he can" (p.4). One way for ensuring fit is avoiding literature review before the emergence of all concepts and categories. This can help the researcher to avoid any pre-conceptualization and pre-assumptions about the data and the research topic. Keeping these guidelines in mind writers didn't performed literature review before all categories were emerged.

2) ***Work:*** Work describes the ability of the theory to "explain what happened, predict what might happen, and interpret what is happening in an area of substantive or formal inquiry" [30] (p.4). This must be assured by presented participants´ main concerns. For this study, the practitioner's main concern of managing risk was taken into account and presented. Following theory data analysis guidelines, all codes, concepts, and categories are grounded in the data.

3) ***Relevance:*** Relevance shows if the developed theory is grounded well in the data and is developed systematically through careful analysis of the data [30]. This is ensured through Grounded theory analysis of the data; this analysis gave rise to codes and categories related to "risk management" that are presented in this study.

4) ***Modifiability:*** Modifiability refers if a grounded theory is modifiable or can be altered by further data collection and analysis [35]. The grounded theory is an "ever developing entity, not as a perfected product" [22] (p.43) and the theory of "risk management" presented in this study can be modified with further data collection and analysis.

B. ***Limitations***

This study has following limitations:

1. Grounded theory research is said to be strongly context specific [28]. Therefore, it cannot be generalized to a large population.
2. Data collection was performed without keeping any specific project cases in mind. Therefore, data for this study is the collective experiences of participants working with agile projects.
3. Interviewed participants were project managers. Therefore, this study might present project managers' perspectives, excluding all other stakeholders involved in a project.

VIII. CONTRIBUTION AND FUTURE WORK

This study has contributed to an existing body of knowledge by studying risk management process empirically in agile software projects. The need of more empirical studies in agile software projects are identified by Odzaly and Des Greer [17] and Albadarneh et al. [18]. This study will contribute to understand the role of risk management in agile software projects. This study will help to understand the similarities and dissimilarities between project risk management process in agile projects and non-agile projects. This study has made contribution in terms of identifying the strengths and weaknesses in the current risk management practices in agile software projects, and presented guidelines on how to maximize the impact of the risk management process on project outcome by implementing risk management practices appropriately. Practitioners can use these presented risk management strategies (presented in the recommendation section) to manage risk effectively in agile software projects. We intend to find further empirical evidence from agile projects to make this study more generalizable to the wider population and make the risk management process more effective.

IX. CONCLUSION

This study made an attempt to focus on the agile risk management process in agile software projects. This study presented the ways through which agile risk management is being done in agile projects. Interview data suggested that practitioners are handling risks in agile software projects mainly by two strategies: implicit risk management strategies (communication and collaboration, shorter iterations, frequent delivery, early feedback, and delivering complex parts first) and explicit risk management strategies (relative estimates, burn down chart, SWOT analysis, and risk matrix). In this study, the role of project risk management in agile projects is discussed along with providing an overview over the similarities and dissimilarities between the project risk management process in agile projects and non-agile projects. Strength and weaknesses in current practices are also presented. Based on interview findings, risk management recommendations to make the process more effective are also presented.

REFERENCES

[1] Project Management Institute, inc. PMBOK, *A Guide to the Project Management Body of Knowledge*, 5th Ed., 2012.

[2] R. L. Van Scoy, *Software Development Risk: Opportunity, Not Problem*, Software Engineering Institute, Pittsburgh, PA CMU/SEI-92-TR-030, 1992.

[3] M. Paulk, "Agile Methodologies and Process Discipline". Crosstalk (October 2002).

[4] M. Concha, M. Visconti and H. Astudillo, "*Agile Commitments: Enhancing Business Risk Management in Agile Development Projects*," In: Concas, G., et al. (eds.) XP 2007, LNCS, 2007.

[5] M. J. Carr, S.L. Konda, I. Monarch, F.C. Ulrich and C. F. Walker, *Taxonomy-based risk identification* (No. CMU/SEI-93-TR-06). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[6] N. Cerpa and J. M. Verner, "Why did your project fail?" *Commun. ACM*, vol. 52, no. 12, p. 130, Dec. 2009.

[7] D. Hillson, *Managing risk in projects*, Gower Publishing, Ltd., 2009.

[8] A. Moran, *Agile Risk management*, Springer Briefs in Computer Science. http://www.springer.com/series/10028, 2012.

[9] B.W. Boehm, "Software risk management: principles and practices," *Software*, IEEE 8.1, pp. 32-41, 1991.

[10] C. Chapman and S. Ward, Project risk management: processes, techniques and insights, John Wiley & Sons, Inc., 2003.

[11] M. T. Pich, C. H. Loch and A. D. Meyer, On uncertainty, ambiguity, and complexity in project management. Management science, vol. 48, no. 8, 1008-1023, 2002.

[12] J. Nyfjord and M. Kajko-Mattsson, "Commonalities in Risk Management and Agile Process Models". In: ICSEA 2007, Cap Esterel France (August 2007).

[13] P. L. Bannerman, "Risk and risk management in software projects: A reassessment," J. Syst. Softw., vol. 81, no. 12, pp. 2118–2133, Dec. 2008.

[14] A. J. Dorofee , J. A. Walker, C. J. Alberts, R. P. Higuera, and R. L. Murphy, Continuous Risk Management Guidebook. CARNEGIE-MELLON UNIV PITTSBURGH PA, 1996.

[15] K. Schwaber, Agile Project Management with Scrum. Microsoft Press, 2004.

[16] O.K. D. Lee and D. V. Baby, "Managing Dynamic Risks in Global It Projects: Agile Risk Management Using the Principles of Service-Oriented Architecture," *Int. J. Inf. Technol. Decis. Mak.* vol. 12, no. 6, pp. 1121–1150, Nov. 2013.

[17] E. E. Odzaly, D. Greer and D. Stewart, *Lightweight Risk Management in Agile Projects.* In SEKE pp. 576-581, 2014.

[18] A. Albadarneh, I. Albadarneh and A. Qusef, "Risk management in Agile software development: A comparative study," In *Applied Electrical Engineering and Computing Technologies (AEECT)*, IEEE Jordan Conference, pp. 1-6. IEEE, 2015.

[19] L. Siddique and B. A. Hussein, "Practical insight about risk management process in agile software projects in Norway," In *Technology Management Conference (ITMC)*, 2014 IEEE International, pp. 1-4, IEEE, 2014.

[20] P. Advice, "Study design in qualitative research—2: Sampling and data collection strategies," *Education for Health*, vol. 13, no. 2, pp. 263-271, 2000.

[21] B.G. Glaser, *Emergence vs Forcing: Basics of Grounded Theory Analysis*, Sociology Press, pp.16, 1992.

[22] B.G .Glaser and A. L. Strauss, "The discovery of grounded theory: Strategies for qualitative research." Aldine, pp. 105-115, 1967.
[23] M. Birks and J. Mills, *Grounded Theory: a Practical Guide*, Sage Publications Limited, 2011.
[24] C. Marshall and G.B. Rossman, *Designing qualitative research*, Sage publications, 2014.
[25] L. Siddique and B. A. Hussein, "Grounded Theory Study of the Contracting Process in Agile Projects in Norway's Software Industry," *The Journal of Modern Project Management*, vol. 4, no. 1, 2016.
[26] L. Siddique and B.A. Hussein, "A qualitative study of success criteria in Norwegian agile software projects from suppliers' perspective," *International Journal of Information Systems and Project Management*, vol. 4, No. 2, pp. 63-79, 2016.
[27] L. Siddique and B. A. Hussein, "Grounded Theory Study of Conflicts in Norwegian Agile Software Projects: The Project Managers' Perspective," *Journal of Engineering, Project, and Production Management*, vol. 6, no. 2, pp. 120-135, 2016.
[28] M. E Hussein, S. Hirst, V. Salyers and J. Osuji, "Using grounded theory as a method of inquiry: Advantages and disadvantages," *The Qualitative Report*, vol. 19, no. 27 , pp. 1-15, 2014.
[29] J.M. Corbin and A. Strauss, "Grounded theory research: Procedures, canons, and evaluative criteria," *Qualitative Sociology*, vol. 13, no. 1, 1990.
[30] B. Glaser, *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*, Sociology Press, Mill Valley, CA, 1978.
[31] B. Glaser, *Doing Grounded Theory: Issues and Discussions*, Sociology Press, 1998.
[32] S. Georgieva and G. Allan, "Best Practices in Project Management through a Grounded Theory Lens," *Electronic J. Business Research Methods*, vol. 1, pp. 43-52, 2008.
[33] B. Glaser, "Remodeling Grounded Theory," *Forum: Qualitative Social Research*, vol. 5, no. 2, article 4, 2004.
[34] J. Breckenridge, *Being person driven in a service driven organisation: a grounded theory of revisioning service ideals and client realities* (Doctoral dissertation, Queen Margaret University), 2010.
[35] H. Thulesius, A. Hakansson, and K. Petersson, "Balancing: A Basic Process in End-of-Life Cancer Care," *Qualitative Health Research*, vol. 13, no. 10, pp. 1353-1377 Tilley, N. 2000, Realistic, 2003.
[36] M. Cohn, *Succeeding with agile: software development using Scrum*, Pearson Education, 2010.
[37] M. Williams, J. Packlick, R. Bellubbi and S. Coburn (2007, August). "How We Made Onsite Customer Work-An Extreme Success Story," *In Agile Conference (AGILE)*, 2007, pp. 334-338, IEEE.
[38] M. Tomanek and J. Juricek, "Project risk management model based on PRINCE2 and SCRUM frameworks," arXiv preprint arXiv:1502.03595, 2015.
[39] K. Schwaber and J. Sutherland, "The Scrum Guide: The definitive guide to Scrum: The rules of the game." *SCRUM.org*, Jul-2013.
[40] S. Thomas, Agile Risk Management, Available: http://itsadeliverything.com/agile-risk-management, 2008.
[41] P. G. Smith, and R. Pichler, "Agile risks/Agile rewards." *Software Development*, vol. 13, no. 4, pp. 50-53, 2005.
[42] C. R. Nelson, G. Taran and L. de Lascurain Hinojosa, (2008, June). "Explicit risk management in agile processes," *In International Conference on Agile Processes and Extreme Programming in Software Engineering*, pp. 190-201, Springer Berlin Heidelberg.
[43] D. W. Hubbard, *The Failure of Risk Management: Why It's Broken and How to Fix It*, Wiley, 2009.
[44] X.N. Lu and Q.G. Ma, "Risk Analysis in Software Development Project with Owners and Contractors", In: *International Engineering Management Conference* (October), 2004.
[45] B. Derfer, *Introducing the Agile Risk Management Framework, Agile Six Applications*, Inc., 2016.
[46] V. Ylimannela, *A Model for Risk Management in Agile Software Development*, Tampere University of Technology, 2011.
[47] K. Horvath. (2014) Risk Management in Agile and Waterfall Environments, available at: http://intland.com/blog/sdlc/risk-management-in-agile-and-waterfall-environments/ (assessed on 31-03-2016).
[48] M. Lant (2010), Five Simple Steps to Agile Risk Management, Available at: https://michaellant.com/2010/06/04/five-simple-steps-to-agile-risk-management/.
[49] B. Livingstone on (09/09/2015), Using Agile practices to manage project risk. Available at http://www.equinox.co.nz/blog/agile-practices-manage-project-risk.
[50] J. Nyfjord, "Towards integrating agile development and risk management," 2008.

Lubna Siddique

Lubna Siddique is studying for a PhD in agile software project management from the University of Oslo, Norway. Her research interests include working with agile methodologies, agile software project management, working within agile software teams and software process improvement.

Bassam A. Hussein

Bassam A. Hussein is an Associate Professor at the Norwegian University of Science and Technology (NTNU) Trondheim, Norway. His interests include application of gaming simulations, e-learning, requirements management, and organizational learning. He teaches project and requirements management and has been involved in the design, development, and implementation of a wide range of customized education programs in project management.