

Agile Risk Mitigation Framework

Anum Naz, Ahmad Salman Khan¹, Muhammad Atif²

¹Department of Software Engineering, The University of Lahore, Pakistan ,

²Department of Computer Science & IT, The University of Lahore, Pakistan

Summary

Software organisations follow different methodologies for the development of software. The software development methodologies are mainly divided into two categories, including plan-driven and agile development. To attain project success, it is very significant to consider risk management during whole project. Agile development is considered risk-driven, but many risks are unreported at the industrial level. The risks can be divided into three categories, including (i) development risks, (ii) organisations risks, and (iii) people-oriented risks.

This paper deals with Development risks specifically. Several risks related to development are faced by people working in the industry while dealing with agile development. Their management among the industry is a big issue, so this paper emphasises ARMF based on development-related risks by following agile development. This research work will help software organisations to prevent different project-related risks during agile development. The risks are elicited at two-level, (i) literature-based and (ii) IT industry based. A systematic literature review was performed for eliciting the agile risks from the literature. Detailed case studies and survey research methods were applied for eliciting risks from IT industry. Finally, we merged the agile development risks from literature with standard industrial risks. Hence, we established an agile risk mitigation framework ARMF based on agile development and present a groundwork established in light of empirical examination for extending it in future research.

Keywords:

Agile, Risk management, Scrum, Kanban

1. Introduction

Software organisations follow development methodologies in order to produce software. For decades, the organisations have followed standardised ways for the development of the software in order to deliver software under budgetary and schedule constraints. Many enhancement solutions have been proposed, from the reliability and measurement of the software process to a large number, procedures and practices. Recently, many of the agile development ideas for improvement have come from skilled practitioners, who have mentioned their techniques as agile software development [25]. The agile methodology has had an enormous impact on how the development of software is performed worldwide. However there are a number of agile methods used, but there is a little set of knowledge present about how these agile methods are

carried out in reality and what their impact on the software industry is.

The methodologies for software development are mainly divided into two categories, including agile development and plan-driven. Agile development is used for developing business software products. While following agile development methodology, software engineers face many risks like development-oriented risks, lack of documentation, criticality, reliability and safety requirements, culture, people, and communication risks. Managing risks in a proper way have traditionally been an essential part of software development. The change from traditional models, such as the waterfall model, to new methodologies, has created new risks in software development. Risks in development will not only cause economic loss but also dissatisfaction of clients. Flexibility and adaptability are also significant issues that are created as a result of the above-mentioned risks. These are the major factors during agile development that will cause an unsuccessful project.

Risk management in agile development is gaining day by day popularity. Nowadays, agile development has revolutionised the software industry by changing people mindset towards software development. A survey study conducted in the USA and Europe showed that agile methods are followed by 14% of software industry, 49% of software companies' shows interest in adopting agile methods [25]. To provide a risk mitigation framework at the industrial level is the need of the current time. As many risks are still unreported at the industrial level.

The aim and objective of this research work is to develop an agile Risk Mitigation Framework (ARMF) for Agile Development. These include (1) the risks reported at the industrial level and (2) the risks that exist in literature. Our work focuses particularly on reporting and providing solution for all risks that are in literature as well as that are existing in industry.

The major concern of this study is to report all risks provided in the literature faced by the agile development team and to develop and validate the agile development risk mitigation framework (ARMF) by obtaining industrial feedback.

Furthermore, the defined ARMF will be validated by comparing it in the industrial setting. The major concerns of this research study are:

- Identify the risks related to agile development
- Identify the existing frameworks for risks mitigation
- Define a new agile development risk mitigation framework (ARMF).
- Investigating the applicability of ARMF among industry.

- RQ1. What are the existing risks in agile development?
- RQ2. What are the existing framework for mitigating risks in agile development?
- RQ3. What are the risks industry is facing in agile development?
- RQ4. Does the published framework mitigating risks faced by the industry?
- RQ5. Is there any best approach to mitigate risks in agile development facing the industry?

2. Research Method

2.1 Research Design

The selected research topic is interconnected to the agile development risk mitigation process and validation of that mitigation process. Both qualitative and quantitative research approaches were used here to effectively obtain study results. Our research method consists of five phases. These are (1) Literature review, (2) Eliciting existing frameworks, (3) Industrial case study, (4) Proposed framework and (5) Framework validation. Figure 1 below shows the stages involved in the study process. Below we explain the research method. Our research method consists of the following phases.



Fig 1: Research Design

2.1.1 Systematic Literature Review

The objective of SLR was to explore existing risks in agile development and mitigation frameworks to find out the answer of our research questions.

- RQ1 what are existing risks in agile development?
- RQ2 what are existing frameworks for mitigating risks in agile development?

The goal of this step was to explore and list down all the existing risks and mitigation frameworks in agile. The first phase, the Systematic Literature review, consisted of three steps. In the first step, we elaborated on the review's objectives. In the second step, we conducted the following activities

- Selection of primary papers discussing agile risks and their mitigation
- Assessment of selected study
- Drawing out and extraction of facts related to agile risk mitigation

In the third step, we extracted the information to be included in the literature review process. The information collected consists of research publications. To gather literature related to agile development, we used different resources and databases. Following databases were used:

- Databases related to electronic media
- Digital library of ACM
- IEEE forum
- Google Scholar (scholar.google.com)

We gathered information mainly on agile development existing risks from literature and related mitigation strategies. Several research papers, journals and proceedings were considered. To find relevant literature of our study area, we used different keywords in above-mentioned search engines and research databases. The following search strings were used to retrieve the required and relevant primary studies.

Table 1 Strings formation

Synonyms	Synonyms	Synonyms
Agile method	Risk management	Framework
Scrum	Threat management	Process
XP	Risk organization	Practices
TDD	Risk mitigation	Actions
	Risk analysis	Activities
	Risk planning	
	Risk monitoring	

2.1.2 Selection Procedure and Criteria:

We examined titles, abstracts and introduction of each article. At the same time, irrelevant papers were excluded. Following is the inclusion criterion in selecting the primary studies

2.1.3 Selection Procedure and Criteria:

The following is the inclusion criterion in selecting the primary studies

- Access to full text study is available.
- Studies with peer-reviewed
- Research thesis and technical reports
- Published studies as a book section or book.

We considered all types of publications mentioned above. The only prerequisite was that the publication should discuss agile risk management.

i. An exclude criteria in selecting the primary studies is as follows:

Studies other than in the English language. Already included studies.

Private web pages. The studies that discuss agile methods but not risk mitigation were excluded.

ii. Quality assessment checklist

The selected research areas were assessed based on the method used for conducting the research, gathered results, analysis and the conclusion section.

2.1.4 Gathered Results

In this phase, we examined the following questions. Does the research article completely define the results of the study? Are these results appropriate from the perspective of our research topic? Are there any validity threats linked with the research article?

2.1.5 Analysis

How was the data assessed and investigated in the research article?

2.1.6 Conclusion section

In this phase, we checked that how appropriate is the conclusion given in the research article? And to what level the conclusion was applicable to our research study? Whether the conclusion discussed about the limitations and restrictions of the research study?

2.1.7 Strategy used for data extraction

Data from the selected primary research studies was extracted by using forms. Data extracted was primarily based on general and explicit information. The search is performed on selected search keywords and data sources. Addition and elimination criteria were applied on search outcomes manually by reading the titles and abstracts, which left with the 38 studies. After reading full articles, 18 studies were extracted out after reading full texted papers and only 20 studies are selected.

2.2 Framework exploration via Industrial Case study

The goal of this phase was to explore our framework in industry and evolve the framework based on the feedback provided by the industry. The word case study signifies importance of observation and field based studies, both paying intension towards different perspectives of research method. Raw data was collected as a result of a case study . We will identify the key challenges by using case study . Then by using analysis of risks mitigation framework case study will be performed to check whether these are the same risks that industry was facing. Case study was conducted by using questioner's technique. The reporting checklist items provided guideline while we constructed questioner for case study.

2.3 Evaluation of purposed frameworks by using survey

In this phase, we evaluated the proposed framework in industry. This was the final phase of our research. In this phase, we were examined the applicability of our framework by applying it in industrial setting. A survey can be expressed as a data-collecting and analyzing whereby using question answer approach results were gathered.

3. Literature Review

In this section, we provide a comprehensive background of agile development risks and risk mitigation frameworks provided in literature. Literature Review includes basic terminologies and concepts regarding agile development and a brief overview of how to manage risks during agile development respectively.

3.1 Risks in Agile Development

In this section, we described the risks that exist in agile development. Table 2 summarizes the risks found in literature.

3.2 Description of categories of risks

Based on systematic literature review, we found agile risks can be divided into three categories. These are (1) Development risks, (2) people oriented risks, (3) Organizational and Technical risks. Below we describe the risk categories.

3.2.1 Development risks [2][4]:

Development risks are challenges that Developers face during agile development.

3.2.2 People oriented risks

People oriented risk in agile method are those risks that are associated with persons that are performing set of activities in agile development.

Table 2: Risk categories

Risks	Ref
Development risks	
Development time	[2]
Code size	[2]
Code quality	[2]
Code coverage	[2]
Legacy code	[2]
Defect reproduction	[2]
Development oriented risks	[13]
insufficient testing skills	[2]
Development process conflicts	[2][4]
Lack of documentation	[2][6]
Design problems	[2]
Risks associated with neglecting non-functional requirements while developing design.	[11]
Architecture oriented risks.	[13]
Increased number of sites.	[15]
People oriented risks	
Perceptions	[2]
Experience and Knowledge	[2]
Communication & collaboration (Customer)	[6]
Culture & Communication	[7]
Communication and interaction patterns risks	[7]
Agile development adoption risks	[8]
Problems with customer incapacity to understand things, lack of harmony between clients	[11]
Customer oriented risk	[13]
People oriented risks	[14][15][16][17][18]
Organisational and technical risks	
Domain and tool specific issues	[2]
Lack of tool support	[2]
Resources risks	[2]
Budget constraints, scope issues estimation of time to market a product.	[5][14]
Criticality, reliability and safety requirements.	[6][14]
Process life-cycle challenges.	[7]
Degree of agility.	[9]
Business process conflicts.	[14]
Inappropriateness of technology and tools.	[17][18]

3.3 Mitigation Framework /Process for Risk Management In Agile Development

In this section, we discuss the existing framework for Agile risk mitigation.

Table 3 Existing Frameworks for Agile risk Mitigation

Framework name	References
Release planning related Risk-driven process.	[1]
Conceptual Framework	[15]
Ville Ylimannela suggested model	[4]
Agile Commitments	[5]
Agile adoption based on four-stage process.	[8]
The framework based on Agile Software Solution.	[9]

3.3.1 Framework 1: Risk-driven method for release planning

Ming Shu Li et al proposed a method for risk-driven XP release planning. He pointed out three risks that may occur during XP release planning. These risks are (1) lack of technique for XP release planning, (2) lack of technique of risk analysis and (3) ambiguous release planning agreement between stakeholders.

Firstly, the ambiguous technique of extreme programming release planning mainly provides wrong observation to development team as due their wrong perception they do not consider different chunks of release plans and move to next iteration. Three major goals were described here. First goal showed the unclear method of XP release planning often provides wrong perception to all stakeholders. Secondly, stakeholders were much confused so might not properly deal with risk management. Lastly, there was no agreement so the stakeholders were unable to keep balance risks management and efficiency in method [1].

Solution presented in Framework1:

Ming Shu Li et al suggested three practices for risk mitigation for XP release planning. These are as follows [1].

- Construction of feasible release plans

System's extent, price tag, timetable and system quality were construed by developers in the form of possible release plans from original project ideas. This step deals XP's stories creation and one more thing that is to combine stories by considering values, dependencies and price tags in the form of different types of possible release plans.

- Analysis of feasible release plans

Analysis of risks is performed for a feasible release plan. Clients and the development team used risk analysis as the best tool when dealing with plan releases. Risks are losses caused by uncertain things; they maybe come from requirements, estimation or technologies and affect system scope, schedule or products quality.

- The decision of release plan

After getting results from risk analysis release plans are decided by stakeholders that are important for next iteration. Project profiles are settled by using information gathered

after the completion of an iteration. Objective plans and even software life cycle use this method as it is used for each extreme programming iteration. For example, this method is used by developers to produce release plan foremost used for extensive-time objectives. And when the results are compiled after the completion of first iteration then results of the iteration can be adjusted based on first iteration. This adjustments guides in form of suitable plan for upcoming iterations and further goals. This method helps for both scheduling for next iteration and for long-time targets. It also builds link between the stakeholders and overall details of project activities. All this helps stakeholders to create a perfect product by considering all condition's and controls [1].

3.3.2 Framework 2: Conceptual Framework

Research risks like temporal, geographical and socio-cultural distance were addressed by Eman Hossain et al in form of a framework named as conceptual framework. Different set of methods were discussed to fix number of risks. They discussed in detail about impact of these risks on project and activities related to collaboration, coordination and teamwork practices. Global software development (GSD) was major focus discussed in conceptual framework, and broad evaluation was performed. Not only evaluation was performed but an analysis of the GSD was presented in form of systematic research literature. The conceptual framework graphically illustrates key challenges that occurred during GSD project, and also scrum practices are reported as existing approaches to cope with the risks [15].

Solution purposed in Framework2:

Eman Hossain et al purposed conceptual framework is based on two major elements A.) Framework Development Process B.) Framework Components [15].

A.) Framework Development Process

While in Framework development several steps are planned to develop framework. That are as follows. A detailed survey was conducted by them related to GSD literature where agile strategies were followed. Then key components were analysed and heuristically data about GSD researcher's practitioners was collected in this step. Then categorisation of that Key risks reported as a result of this step was done by them. And finally this framework was invented to cope with all challenges identified [15].

B.) Framework Components

This framework consisted of components that were widely expressed as 1) Major risks 2) Current strategies to mitigate these challenges (as provided in diagram given in paper). Risks are categorised and against every category current strategies and practices are mentioned [15].

Framework 2 Limitations

The major limitations of this framework was that this framework given strategy has a small and slight focus on project concerns. However, on the other hand it is realized that for GSD projects that are done for real life some major significant problems existed. In mostly cases GSD consisted of a number of projects (sometimes ranges from 10 to 20) and all are combined in form of product assimilation effort. Hence, as a result all projects were never considered properly by using this framework so this is the limitation for this framework [15].

3.3.3 Framework 3: Ville Ylimannela Suggested Model

Ville Ylimannela purposed a model by considering the knowledge from already used models and then extending the idea by adding his opinion, the idea behind his effort was to provide a proper strategy to deal with problems that software developers face because of risks in agile development. The major purpose of Ville Ylimannela contribution was to produce solution for mitigating risks faced by agile development team. For this purpose interviewing approach was deployed in two companies. Both companies were asked by interviewing team that what risk management strategy they used and how they were dealing with risks in agile development.

Solution purposed in Framework3:

This model consisted of all challenges that were defined in PMBOK (project Management Body of Knowledge), and all about performance of main risk administration stages discussed in PMBOK starting from planning phase of risk administration till monitoring. The model rotated around a risk management body, which was fully informed of all the activities and about all updates that the software development cycle faced. This model is till now never used for any real time problem. Following are set of activities that are performed under the suggested model to meet risks [4].

- Risk board and risk notes
- Checklists
- Processes
- Planning
- Risk identification, assessment and response
- Risk approvals and monitoring

Framework 3 Limitations

This model have not used for any real time development scenario so not verified till now [15].

3.3.4 Agile Commitments

Mauricio Concha et al proposed a framework with name agile commitments. This framework is based on the

basic concept of relationship between customers and developers and dedications between customer and developer was discussed in this framework in detail. The major objective of this framework was to improve risk mitigation by changing business hopes regarding risk management, and also shown in form of a cooperation baseline between clients and developers. This framework was purposed to offer risk management. The coordination between the customers and developers during the whole project was the key point discussed through this framework [5].

Solution purposed in Framework4:

The definite ideas for this framework were to:

- Classify and identify the dedication among customers and developers.
- Decision of consent for core inspirations.
- Managing overall control among the agreed dedications throughout the whole project.
- Enhance risk administration by using risk visibility approach on the business objects that are as follows, scope, excellence, financial plan, and time-to-market.
- By offering a cooperation baseline for both clients and development team.

There were two important components of agile commitment framework. One was stated as conceptual schema framework, and second component was an instantiation guideline for project level. First component which was the theoretical classification of the framework described how the framework was organized; and second component as mentioned above described complete guidelines followed by managers for implementation of the agile commitments for some specific projects[8]. This framework was divided into 4 methods, and each one was divided into set of explicit goals.

- Business inspiration
- Project objectives
- Agile procedure measurement
- Risk organisation of project.

3.3.5 The agile adoption four-stage process

Ahmed Sidky et al purposed the agile adoption framework. They introduced a strategy in the form of this framework that was based on an effort to deal with the problems mentioned by declaring a prepared and repeatable method prepared to guide during agile adoption activities. This framework was produced to deal with problems of those people who want to adopt agile practices. The purposed skeleton was based on two components, first component was an agile measurement index, and second component was reported as a four-stage process, both these components provide basics for guidelines for those who want to adopt agile development. Additionally the Sidky

Agile Measurement Index (SAMI) included five agile steps that were used to classify the agile prospective of developments. The four-stage process described in this framework explored some steps that were as follows: whether the organisations were prepared for agile adoption or there was any guideline present related to their potential. And finally set of activities were evaluated for agile development [8].

Solution purposed Framework 5:

The purposed framework consist of two major components. The first purposed component was stated as the Sidky agile measurement index (SAMI). And this was a scale that was used by many people to recognise the potential of some product or association. This agile measurement index mainly consisted of four stages. These four stages worked together to provide guideline for organisations.

These four stages were:

Stage 1:

Recognition of discontinuing parts. Identification of show stoppers was done in this step that cause problem for adoption of agile development.

Stage 2:

In second step assessment for project was achieved. This step was performed by utilising the SAMI that determined the target level for agility of specific product.

Stage 3:

Managerial dedication was performed during this step in form of assessment form. Here SAMI was used to find out the amount of agility level for each project for different organisations.

Stage 4:

Understanding of target agile level was achieved during this step. Identification of final set of agile development to be adopted by development team was specified for projects.

Second component of purposed framework was four-stage assessment process. The purposed process was considered the backbone of the agile adoption framework. Data given in paper [8] provided an assessment component that was used to check whether an organisation ready to move towards agility. Secondly, the purposed process also provide guideline to an organisation in order to judge whether these practices were easily adopted by an organisation. The four steps that were given in detail were grouped together to ensure improvement where it was needed [8].

Table 4 Existing Agile risk Mitigation Framework comparison for risk categories

Risk mitigation Framework name	Development risks	People oriented risks	Technical risks	Organisational risks
Risk-driven method for release planning	✓	✓	✓	x
Conceptual Framework	x	✓	x	x
Ville Ylimannela suggested model	✓	x	x	x
Agile commitments	x	✓	x	✓
The four-stage process for agile adoption	✓	x	x	✓

4. Industrial Case Studies

In this section, we present the results of case studies conducted in IT industry. We conducted three case studies in three different IT-based companies. We gave fictive names Company A, Company B and Company C to companies due to data sensitivity. However, the company's introductory information has been discussed. We explore agile risk management in these companies by following IEEE Risk Management standard (IEEE STD 1540-2001). So that we may evaluate agile development-related risks categories that IT industry is facing. This also helps us answer the question about mitigation methods followed by companies to deal with these risks.

IEEE Standard for Risk management is the standard that describes a process for risk management of software. IEEE risk management standard explains software phases like the acquisition of software, how to deliver software, development of software, operations, and maintenance.

This standard is entirely focused on risk management techniques. There are six major phases that are involved in IEEE standard that are as follows:

- a. Plan and implement risk management
- b. Manage the project risk profile
- c. Perform risk analysis
- d. Perform risk monitoring
- e. Perform risk treatment
- f. Evaluate the risk management process

All these phases are explained in detail in IEEE standard paper [21]. We followed these phases and formulated an expressive questionnaire. By using case study research, we gathered information about three companies whose detail is given in further paragraphs.

Questionnaire:

We designed a questionnaire in order to collect data from given below three companies A, B&C. The questionnaire had two section. Section 1 consists of "General questions" related to the interviewee personal information and company information. The aim of Section 2 is to focus on "Specific Questions". The specific questions are related to each risk presented in IEEE risk management framework.

4.1 Case Study 1

Company A

The company is based in Lahore, and they develop Enterprise Resource Planning systems. The role we interviewed was IT manager. There are 1500 employees in the company. However, only fifteen people are working in IT department. We took ERP system development as a case for discussion. The IT department has been working on the project for last two years. The company follow agile development and they planned for risk management.

Company A follows agile development with a focus on Test Driven Development. Their Test Driven Development process consists of four phases. These are as follows:

- *Red Block*
- *Green Block*
- *Refactoring*.

In the first phase, *Red Block*, the developers write the test cases (testing code) against no code. It basically helps us to test the single module automatically. In short, we write that code for automatic unit testing. The second phase, *Green Block*, is to write the code for that module so that the module can pass the test cases of first phase. Finally, in third phase, *Refactoring*, change in code is adopted and the test cases are re-executed.

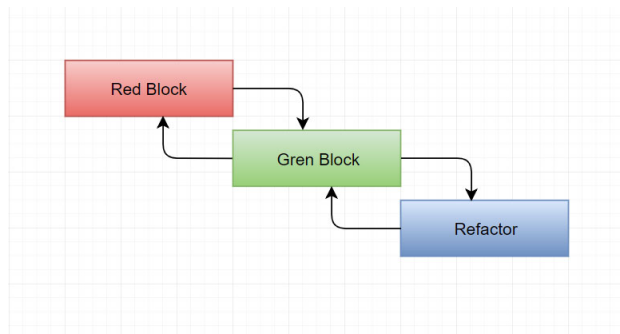


Fig 2 TDD phases of Company A

Design & Architecture	Identified during Req. Analysis	Evaluate d during Designing phase	Monitored during Green Block (TDD)	Treated during planning and Req. analysis
Non-functional Req.	Identified during Implementation	Evaluate d during testing	Tracked during Green Block (TDD)	Treatment of Build

Table 5 Mapping of risk management activities on Agile Process phases in Company A

Development time	Risk Identification	Risk Evaluation	Risk Monitoring	Treatment
Code size	Planning	Req. Analysis	Track & Monitor	Treatment of risk
Code quality	Red Block (TDD)	Risk evaluation of Build	Track and monitor	Treatment in Refactor form
Legacy Code	Identification during development	Evaluation during Code review meeting	Track monitor	Development of reports
Code Coverage	Identification during Refactor (TDD)	Evaluating Build	Monitoring Build	Treatment of that Build
Defect Reproduction	Identified during Red Block	Evaluate d during Testing	Tracking during testing	Treated during Refactor
Insufficient Testing	Identified during Testing	Evaluate d during build	Tracked during build	Treated during build
Development Conflicts	Identified during Green Block	Evaluate d in green block	Monitored during green block	Treated in form of Refactor (TDD)
Lack of Documentation	Identified during Development	Not evaluate d	Development / Building / Release without documentation	No treatment

4.2 Case Study 2: Company B

This company is based in Lahore and they develop Enterprise Web Application for transport. This company deals with different nature of products but mainly they are dealing with the EWA (Enterprise Web Application) from one year approx. They have 25 staff members who are working in this company. It is basically a SME company but is dealing with international clients. We have conducted the interview from one of their senior resource who is currently working on the project from the last year. They follow agile methodology and somehow they do plan for risk management.

Company B follows agile development with a focus on Kanban methodology. Kanban mainly uses the agile phases some of its phases are mentioned below:

- Workflow
- Work in Progress
- Enhance Flow

In work flow phase, they usually list down all the modules to be delivered in a queue and then it is picked by its priority. The next phase balances the flow of development so that the team don't ended up with too much workload. They pick up a task from the queue and start developing it. The last phase started when they finished a module and then they pull a task from the task's queue.

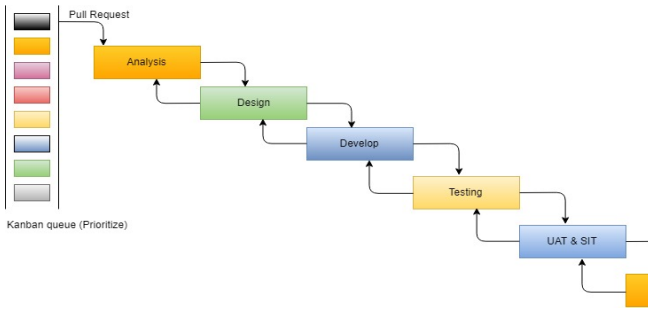


Fig 3 Process of Kanban in Company B

Table 6: Mapping of risk management activities on Agile Process phases in Company B

	Risk Identification	Risk Evaluation	Risk Monitoring	Treatment
Development Time	Analysis	Analysis	Req. Analysis	Planning and Changes
Code Size	Development	Development	Development	Development
Code Quality	Development	Development	Development	Development
Legacy Code	Development	Development	Development	Development
Code Coverage	Testing	Unit Testing	Unit Testing	Development
Defect Reproduction	Testing	Testing	Testing	Development
Insufficient Testing	Testing	Testing	Testing	Testing
Development Conflicts	Development	Testing	Unit testing	Req. Analysis
Lack of Documentation	X	X	X	X
Design & Architecture	Req. Analysis	Design	Design	Design (Architectural)
Non-Functional Req.	UAT	Testing	Testing	Development

4.3 Case Study 3: Company C

This company is based in Lahore and they develop Enterprise Web Application for transport. This company deals with different nature of products but mainly they are dealing with the ERP, Websites, Mobile Apps, Online Management Systems, and Oracle ERPs. They have 120 staff members who are working in this company. We have conducted the interview from one of their senior resource who is currently working on the project from the last year. They follow agile methodology and somehow they do plan for risk management.

Weekly Sprint

Product Backlog

In this organisation, they are following the weekly sprint method for their work meeting so the seniors will be able to know what the team members have done in a week after they have assigned tasks to their related modules. This makes the release of the application quick as they deploy the application in phases. Product owner guides the sprint team related to the modules they are working. They then define the priorities of the tasks they are going to work on and scrum master logs that in their priorities in Product Log.

Table 7: Mapping of risk management activities on Agile Process phases in Company C

	Risk Identification	Risk Evaluation	Risk Monitoring	Treatment
Development Time	Backlog	Backlog	Update Product Backlog	Sprint Planning Meeting
Code Size	Development	Development	Development	Refactoring
Code Quality	Development	Development	Daily Sprint	Daily Sprint
Legacy Code	Development	Development	Daily Sprint	Refactoring
Code Coverage	X	X	X	X
Defect Reproduction	Testing	Testing	Sprint Review	Development
Insufficient Testing	Testing	Update Product Backlog	Daily sprint	Testing
Development Conflicts	Daily Cycle	Daily Cycle	Daily Cycle	Daily Cycle
Lack of Documentation	X	X	X	X
Design & Architecture	Design	Sprint Planning	Daily Cycle	Sprint Planning
Non-Functional Req.	UAT	Peer Reviews	UAT	Development

4.4 Final assessment

By concluding the final results, we assessed that development time risk and insufficient skills are highly critical for industry. These risks cause projects failure. On the other hand, defect reproduction has different reviews

from the industry, thus having different impacts presented in Table 8 below. Similarly, Code Quality, Code coverage and Legacy code are less critical. Finally, there are some risks faced by some organisations which are not much important as a result their impact is minimal on industry. These risks may restrict the use of agile development for industrial projects.

Table 8: Final Assessment

#	Risk Title	A	B	C
1	Insufficient testing skills	High	High	High
2	Development time risk	High	High	Medium
3	Defect Reproduction	Low	High	High
4	Development Process Conflicts	Medium	High	High
5	Lack Of Documentation	High	Low	High
6	Architecture Oriented Risk	High	Low	High
7	Code Quality	Medium	Medium	Medium
8	Code Size	Medium	Medium	Low
9	Risk associated with non-functional requirements while developing design	Medium	Low	Medium
10	Legacy Code	Low	Low	Medium
11	Code Coverage	High	Low	Medium

5. Agile Risk Management Framework

5.1 ARMF based on Test Driven Development

Test Driven Development is a type of agile development. Figure 4 represents TDD working. The TDD comprise five phases that are given below.

- Requirement analysis and planning
- Designing
- Development
- Testing and Red block
- Green block

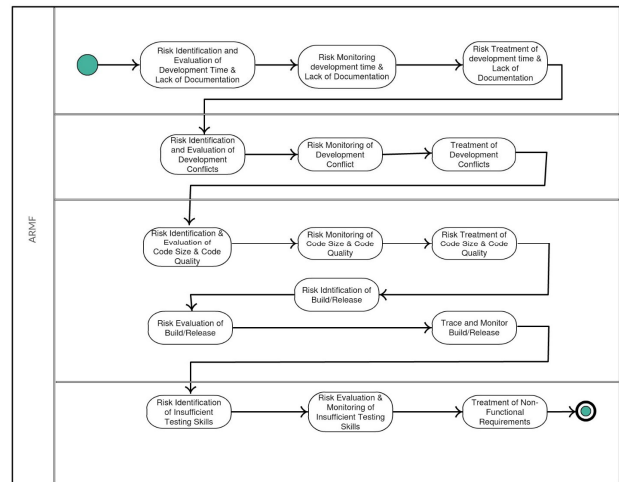


Fig 4 ARMF based on TDD

5.2 ARMF based on KANBAN

This Framework follows basic steps of agile software development that comes in form of Kanban. Every step here act as a phase where companies performs risk mitigation of different risks faced during the software development.

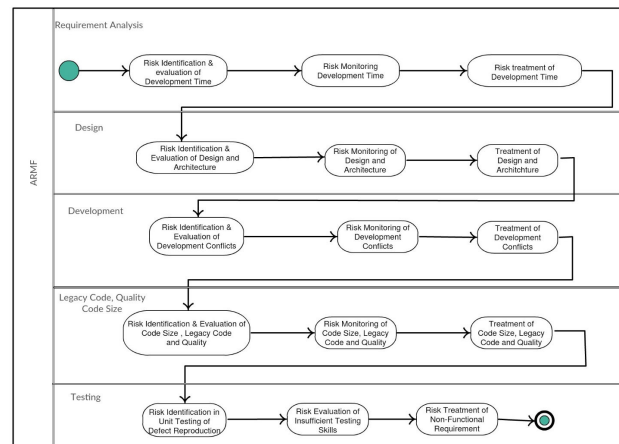


Figure 5 ARMF based on KANBAN

5.3 ARMF based on Scrum

Some companies follow Scrum based on agile methodology. Following are the steps

- Sprint backlog
- Sprint planning
- Daily sprint
- Testing
- Update backlog

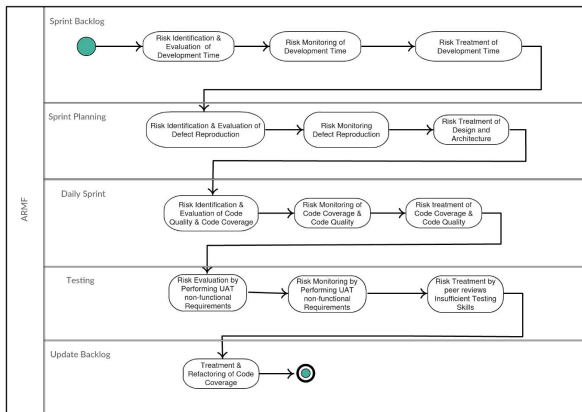


Fig 6 ARMF based on Scrum

6 Conclusion

In this study, we developed and validated a process called Agile Risk Mitigation Framework (ARMF). In the first step to develop the (ARMF) process, we did a systematic literature review. We explored state of the art in agile risk mitigation by performing SLR. After conducting case studies, we developed practices to mitigate risks given in the literature and faced by the industry. Finally, we developed ARMF. Our industrial survey showed that the ARMF process is applicable in the industrial setting. Many large scale organisations are not adopting agile development because they prefer the quality of the process, but in agile development, more focus is given on product quality. We believe that by ensuring risk mitigation, higher quality of product and process will be achieved.

References

- [1] Causevic, A., Sundmark, D., & Punnekkat, S. (2011, March). Factors limiting industrial adoption of test driven development: A systematic review. In *Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth International Conference on* (pp. 337-346). IEEE.
- [2] Mingshu Li, Meng Huang, Fengdi Shu, and Juan Li. A Risk-Driven Method for eXtreme Programming Release Planning. ICSE28th [International Conference on Software Engineering](#) Shanghai, China. 2006. ACM.
- [3] Coyle Sharon, Conboy Kieran. Centre for Innovation and Structural Change, J.E. Cairnes. A CASE STUDY OF RISK MANAGEMENT IN AGILE SYSTEMS DEVELOPMENT. 17th European Conference on Information Systems (Newell S, Whitley EA, Pouloudi N, Wareham J, Mathiassen L eds.), 2567-2578, Verona, Italy.
- [4] Ville Ylimannela Tampere. A MODEL FOR RISK MANAGEMENT IN AGILE SOFTWARE DEVELOPMENT
- [5] Mauricio Concha, Marcello Visconti, and HernánAstudillo. Agile Commitments: Enhancing Business Risk Management in Agile Development Projects. 8th International Conference, XP 2007.
- [6] Mikael Lindvall, Vic Basili, Barry Boehm, Patricia Costa , Kathleen Dangle, Forrest Shull, Roseanne Tesoriero, Laurie Williams, and Marvin Zelkowitz. Empirical Findings in Agile Methods. Second XP Universe and first Agile Universe Conference Chicago, 2002.
- [7] Steffen Bartsch TZI. Practitioners' Perspectives on Security in Agile Development.
- [8] Ahmed Sidky · James Arthur · Shawn Bohner (May 2007). A disciplined approach to adopting agile practices: the agile adoption framework. [Innovations in Systems and Software Engineering](#) 2007.
- [9] A. Qumer, B. Henderson-Sellers. A framework to support the evaluation, adoption and improvement of agile methods in practice. (Nov 2008). *Journal of Systems and Software* Volume 81, Issu 11. Page 1899-1919.
- [10] Frank K.Y. Chan, James Y.L. Thong (2009) Acceptance of agile methodologies: A critical review and conceptual framework.
- [11] Balasubramaniam Ramesh,* Lan Cao† & Richard Baskerville. Agile requirements engineering practices and challenges: an empirical study.
- [12] JaanaNyfjord and Mira Kajko-Mattsson. Commonalities in Risk Management and Agile Process Models. International Conference of [Software Engineering Advances, 2007. ICSEA 2007](#). IEEE.
- [13] Barry Boehm University of Southern California. Get Ready for Agile Methods, with Care. [Computer](#) (Volume: 35, Issue: 1, Jan 2002) IEEE.
- [14] Barry Boehm, Richard Turner. Management Challenges to Implementing Agile Processes in Traditional Development Organizations. [IEEE Software](#) (Volume: 22, Issue: 5, Sept-Oct. 2005)
- [15] Emam Hossain, Muhammad Ali Babar, Hye-young Paik , June Verner. Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework. 2009 16th Asia-Pacific Software Engineering Conference.
- [16] Tsun Chow, Dac-Buu Cao. A survey study of critical success factors in agile software projects. *The Journal of Systems and Software* 81 (2008) 961–971.
- [17] Christof Ebert, Bvs Krishna Murthy, Namo Narayan Jha. Managing Risks in Global Software Engineering: Principles and Practices. 2008 IEEE International Conference on Global Software Engineering.
- [18] Susan Parente. The Art of Agile Risk Management. *PM World Journal* Vol. III, Issue VII – July 2014
- [19] Tore Dyba* , Torgeir Dingsøy. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50 (2008) 833–859.
- [20] JaanaNyfjord. TOWARDS INTEGRATING AGILE DEVELOPMENT AND RISK MANAGEMENT (2008).
- [21] Software Engineering Standards Committee of the IEEE Computer Society IEEE-SA Standards Board IEEE Standard for Software Life Cycle Processes—Risk Management (Approved 17 March 2001).
- [22] Tore Dyba* , Torgeir Dingsøy. Empirical studies of agile software development: A systematic review (24 January 2008).