

Role of Risk Management in Scrum

Rabeeah Jabeen
Preston University Kohat,
Islamabad Campus, Pakistan

Mohammad Daud Awan, PhD
Dean Faculty of Computer
Science Preston University
Kohat, Islamabad Campus

ABSTRACT

Risk management is the major issue in software development. Managing risk can overcome the software development issues that they are of good quality; delivered in a specified time with-in an estimated budget. Delivering best featured software in expected time and budget is a difficult task. Many traditional software process models are heavily documented and having an inflexible control mechanism which makes them difficult to apply to different software projects. Agile Process models provide a gateway to software industry because of their flexibility and adaptability. They focus on code rather documentation.

General Terms

Software Engineering, Software Project management.

Keywords

Scrum, Scrum Risk Management, Risk Management, Scrum Methodology, Scrum Framework, Scrum Risk Management.

1. INTRODUCTION

Computer software is the product that software engineers design and build [1]. Software development is a development of usable software product. The word software-development can be defined as:

The activity of computer programming, which is the process of writing and maintaining the source code, but in broader sense of the term it includes all that is involved between the creation of desired software through the final demonstration of the software, ideally in a planned and structured process[2].

In software industry the risk management is considered as the best practice. The risk management means the decision making activity that reduces or removes the cost for unwanted measures on a project. Controlling risks improves essential software development features such as product quality, planning precision and cost efficiency [3][4]. For this reason, “the inclusion of risk management in software development is an important factor to consider, if one wishes to achieve project success” [5].

Software development in Agile has emerged as a tool to deal with the varying business environments and unexpectedly growing development of software. Agile adoption in advanced software organization has increased continuously during past couple of years. Agile process models state that they determined risk. They state that their iterative approach enables continuous attention to risks and risks can be reduced by practices such as continuous software integration and early testing [6].

The reality is that Agile development models employ few risk management practices. Hence, there is clearly a gap well worth investigating bearing in mind the fact that risk

management is considered best practice in current software engineering.

This research paper comprises of five sections. Section 2 and 3 is about the Scrum Methodology and Scrum Framework. Section 4 is about how Scrum manages risks, Scrum Risk Artifacts and different Risk Factors that are addressed and not addressed in Scrum. Section 5 contains conclusion and future-work for this paper.

2. SCRUM METHODOLOGY

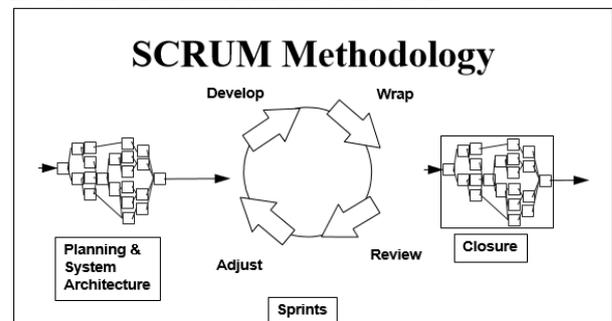


Fig 1: Scrum Methodology

2.1 Planning and System Architecture (Pre Game)

- **Planning:** In planning phase the project team establishes the business case, views for each release, identifies the team members and their roles, necessary tools, risk control, release cost and Review (to determine if enough work is available, and the activities are sequenced to produce the project schedule.) The key deliverable for planning phase is Scrum Release Charter and Release Plan.
- **System Architecture:** After the planning phase the team reviews the backlog, thinks for changes for refinement (old product), learns additional knowledge, analyzes and makes solution. In review meetings the teams exchange information, present progress and highlight problems, reassign changes as required. The key output for system architecture phase is Product backlog.



2.2 Sprints (Game)

A Sprint situates development activities conducted over a predefined period, typically one to four weeks.

- **Planning Meeting:** At the beginning of every sprint there is a meeting for planning in which the product owner creates the sprint backlog, defines sprint goals and divides the sprint backlog into multiple units.
- **Development process:** the Sprint Development consists of analysis (current situation of product and changes if required), designing of process and implementation (creation, testing and document the changes).
- **Wrap** means closing the packets, creating executable version of changes and how they implement backlog requirements.
- **Review** meetings by all teams are held for presenting their work and review (raising and resolving issues and problems, adding new backlog items, review for risk) its progress.
- **Adjust** the information that was made from the review phase is gathered into the affected packets.

The team members at the end of every sprint actualize a burn down chart, sprint review meeting and sprint retrospective.

2.3 Closure(Post Game)

The management team when feels that those variables such as time, competition, cost, requirements and quality are okay, then they enters this phase. During Closure phase Integration, user documentation, training material preparation, system test and marketing material are prepared.

3. SCRUM FRAMEWORKS

The Scrum framework provides support for complex product development. It consists of Scrum Teams, Meetings, artifacts and assumptions.

3.1 Scrum Team

Product owner, the Development Team, and a Scrum Master are the major actors of scrum team.

- The Product Owner is the representative of stakeholders or end users. Scrum teams have one product owner that acts as the member of development team. He or she ensures that the team delivers what is required. The major responsibility of product owner is to write user stories, ranks and prioritize them, and add them to the product backlog.
- Development Team is made up of 3 to 9 individuals. Their cross-functional skills lead to do the actual work which include analysis and design, develop test and documentation and Incremental delivery of product at the end of every sprint.
- The Scrum Master is helpful for the interaction of everyone in the Scrum Team and also suggests which interaction is helpful and which is not.

The Sprint Retrospective is facilitated by Scrum master and usually completed within three to four hours. This meeting answers these questions what went well during the sprint and

what could be improved in the next sprint.

3.2 Meetings

- **Sprint planning meeting:** At the start of every sprint cycle a Sprint planning meeting is held. This meeting is usually eight hours long in which the team selects what work is to be done, prepares the Sprint Backlog , Identifies and communicates how the work that is likely to be done during the current sprint.
- **Daily Scrum:** Daily Scrum meeting is held in computing room at same time. The meeting does not have detail discussions rather answers the questions precisely what work they have done since yesterday, what is their planning for today, are there any impediments/stumbling blocks? The Scrum Master documents the impediments and stumbling block and works towards resolution outside of this meeting.
- **End meetings:** End meetings basically comprise of two meetings the "Sprint Review Meeting" and the "Sprint Retrospective" at the end of every sprint.

The Sprint Review Meeting is of limited time (mostly four hour) in which the review of completed work is done and presents that work in front of customer.

3.3 ARTIFACTS

Artifacts defined by Scrum are specifically designed to maximize transparency of key information needed to ensure [7] Scrum Teams are unbeaten to provide a "Done" Increment.

- Product Backlog is an ordered list of features, functions, requirements, enhancements and fixes to be made in the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.
- Sprint Backlog is the set of Product Backlog items selected for the Sprint plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality.
- Product Increment is the sum of all the Product Backlog items completed during a Sprint and all previous Sprints. At the end of a Sprint, the new Increment must be "Done," which means it must be in useable condition and meet the Scrum Team's Definition of "Done." It must be in useable condition regardless of whether the Product Owner decides to actually release it.
- The sprint burn down chart is a publicly displayed chart showing remaining work in the sprint backlog. Updated every day, it gives a simple view of the sprint progress. It also provides quick visualizations for reference.

4. SCRUM RISK MANAGEMENT

Scrum is an iterative and incremental Agile software development framework for managing software projects and product or application development [8]. Its target is on decisive and complete product development plan where a development group puts their efforts as an entity to reach a common aim. Scrum allows the making of self-organizing groups by promoting co-location of all team members, and oral exchange of ideas among all group members and regulation in the project.

It is a project management framework which emphasis on software development in 30-day sprint cycles in which a particular set of Backlog attributes are carried out. The main idea in application of scrum is its 15-minute meeting on daily basis for regulation and combination. Scrum is being used to deliver several projects successfully.

Scrum is mostly combination of the three aspects (product owner, team and scrum master), credentials (product backlog, sprint backlog and sprint results) and conventions (sprint planning meeting, daily scrum meeting and sprint review).

Software development using Agile well suits for the projects that are scarce in structured planning. Agile development uses automated test cases and assures that it is going to deliver software without any mistakes. Although scrum is test driven development but it is difficult to write test cases during every sprint. Test cases are developed after design phase and testing is made before implementation.

Scrum embraces simple defined processes without help. It cannot be used to well handle difficult and vibrant software projects. Scrum process, project strategy are continuously inspected and adapted based on the empirical reality of the project [9].

In software the major risk factors associated with development project are time constraints, communication, tools and staff. Many projects are unsuccessful because of not providing particular consideration towards the risks and Risk Management.

“Projects usually fail because of management mistakes rather than technical mistakes and it could be argued that managerial issues are more important than technical issues in software engineering projects.” [10]

Table 1: Major Causes of Project Failure [10]

Managerial issues – account for 65 per cent of failure
<ul style="list-style-type: none"> • Inappropriate project structure and channels of communication • Inappropriate resources (poor skill and knowledge mix) • Inappropriate estimates • Inappropriate planning methods (poor scheduling and tracking) • Inappropriate user buy-in • Inappropriate risk management
Technical issues – account for 35 per cent of failure
<ul style="list-style-type: none"> • Inappropriate software requirements • Inappropriate technical design • Inappropriate development and testing tools • Inappropriate technical documentation • Inappropriate technical support • Inappropriate technical reviews (and quality audits)

In Scrum the risks are absolutely handled during daily stand-ups, estimation sessions and sprint planning. Eventually the

completion of sprint indicates that the product is ready to be delivered with the management of all potential risks.

4.1 Scrum Risk Artifacts

Agile risk management employs a variety of artifacts to assist in the visualization of risk tolerance, tailoring of Agile processes, distribution of risk and reward and the tracking of progress towards managing risk [11]. The process is combination risk scoping, risk tailoring and management of risks.

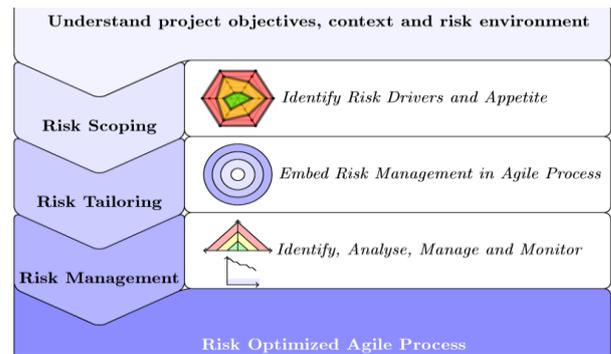


Fig 2: Agile Risk Management Process [11]

- Risk Scoping explains a project risk profile beside the surroundings of the thoughts in the direction of and acceptance of risk inside the organization. It is mostly formed during the preparation of product vision and modified in sprint reviews.
- Agile charts are equally a useful way of corresponding plan, further ask for feedback. Scrum Risk tailoring want alteration in its Agile chart which structure the situation in which risk management actions should take place.
- Subsequently a risk log accounts all known risks, their possibility and collision and suitable risk reaction policy. Risk log is similar to a traditional risk register while there are distinctions in how it is used in Agile risk management.
- Actions to deal with risk can consist of definite tasks for the backlog or risk tagging and may be color veiled by risk response policy.
- At last advancement for taking up risk can be examined by a risk burn down which moreover creates the obvious level of total risk in the Sprint.

4.2 Scrum Role in Risk Management

Scrum teams exercise actual Agile artifacts and meeting to handle risk. They also hang around till the last accountable minute to deal with risk, while they are familiar with the most concerned projects and troubles.

Here are some of scrum artifacts and meetings and their role in risk management:

- Product vision statement facilitates the project team to define the product goals, avoidance of risk and confusion about what the product wants to achieve.
- Product roadmap offers a visual summary of the project requirements and priorities. This summary lets the project team to rapidly recognize gap in



requirements and imperfectly prioritized requirements.

- Product backlog is a helpful means of modifications in the project. It is capable to include changes, to reprioritize requirements and to handle the traditional risk to make better product. This exercise guarantees that the development team works on the good significant number of requirements at the correct time.
- In Release Planning meeting the scrum team talks about which risks to release and how to ease those risks.
- In Sprint Planning meeting the scrum team talks about the risks to the particular requirements and responsibilities in the sprint and how to ease those risks. The discussion regarding risks is in detail but only related to the sprint in progress.
- The sprint backlog presents a rapid vision of the sprint status. This rapid vision facilitates the scrum team to handle risks to the sprint as they happen and reduce its impact by tackling the problems immediately.
- Daily scrum meeting facilitates the development team members to talk about the hurdles that might be or become danger to the project. This discussion lets the development team to handle these risks instantly.
- Task board gives an obvious inspection of the sprint rank, permitting the scrum team to hold risks to the sprint and handle them right away.
- Scrum team guarantees that they develop a product according to the need of its customers. The sprint review also gives a chance to its stakeholders to talk about the amendments and manage change in business needs.
- In Sprint Retrospective the scrum team talks about problems with the precedent sprint and recognize which of those problems may become danger in future sprints. The development team requires finding out conduct to avoid those risks from becoming troubles yet again.

4.3 Scrum Addressed Risk Factors

- **Mitigating schedule flaw:** Schedule flaw is the major risk with respect of planned, opposed to real accomplishment. Estimation is supposition, a prediction. Scrum offers feedback loops to ease unacceptable suppositions. Original Plan is improved and prioritizing is done so the product is delivered with all the fulfillment of requirements. Release plan is updated by the scrum teams at the closing stage of every sprint.
- **Mitigating Specification Breakdown:** Communication and collaboration among the stake holders and the scrum teams clarify and decide what actually they want their product to do. The job of scrum delivery team is to work together with the product owner to guarantee association among what is demanded and how it can be handed-over.
- **Mitigating Scope Creep:** The product owner, and other

shareholders, will determine latest things to incorporate in the product as they observe advancement from the delivery team during the sprint. At the end of every sprint through the demonstration, response from attendees will create new backlog items. The product owner will estimate the latest product backlog items and make a choice from add, delete, and trade-out in priority through further product backlog items.

- **Mitigating Personnel Loss:** Agile development supports self organizing teams with higher degree of self confidence. Working in continuous teams does not indicate that you are confirming an agreement for life time. In Agile project situations there will be return in every team.
- **Mitigating Productivity Variation:** Productivity variation is the differentiation among the unsaid and real accomplishment of the team. In Scrum during the sprint review the delivery teams tackle the performance at the last part of each sprint. Justifying or correcting the difference variation happens in the next sprint. The development team if fails to complete their work in the current sprint then they will do lesser work in the coming iterations.

4.4 Un-Addressed Risk Factors by the Scrum

- **Very large software system:** Since the intrinsic risk management using Agile software development is not enough for large and multifaceted software systems, therefore the resultant increment would be comparatively large. The larger systems can over run its budget and estimated time because the new changes and errors are not easy to accommodate.
- **Large development team:** The major emphasis of Agile development is upon the collaboration among its stakeholders and development team. If the development team is larger than it is difficult to handle communication and collaboration.
- **High reliance on human factor:** Scrum teams must be skilled enough to communicate with its customers for successful development. If the team lacks in expertise then the project fails to meet the demands of its end-users.
- **Inappropriate customer representative:** In Scrum the formal documentation is not created for requirements and specification. The requirements are collected with collaboration of its customer. If the customers do not have the proper knowledge regarding the project or he fails to tell his demands in a proper way the project success must be influenced.
- **Distributed development environment:** Scrum and other Agile practices are best suited for centralized systems not for distributed systems, as these require face to face dealings between the development teams.
- **Scope creep:** Another important risk factor is the scope creep, this usually happens due to the minimal planning conducted in this methodology



which causes developers to become distracted from the project's main objectives [12]. The resultant project will be enlarged and become more difficult to be completed.

5. CONCLUSION & FUTURE WORK

Scrum methodology ensures that the project is well managed by its adoption. Due to shorter iterations the project developed using Scrum meets its timing and cost constraint. Change in requirements is always welcomed in Scrum. Scrum team must need to be skilled for its successful completion. Although the Scrum is having the greater advantages, but as far as the size is concerned Scrum is best suited for small to medium size projects.

The Scrum methodology should be enhanced, so that it can be used for providing the solution for large and complex projects. Although the risk management is built feature in Scrum methodology yet still there is a need of Scrum risk management framework.

4. ACKNOWLEDGMENT

Our thanks to all those who have contributed towards completion of this paper.

5. REFERENCES

- [1] Roger S. Pressman, *Software Engineering a Practitioner's Approach*, (5th edition). McGraw Hill, pp 1-888, 2000.
- [2] June Jamrich Parsons, *New Perspectives on Computer Concepts*, (16th Edition). 2014.
- [3] Ropponen J., and Lyytinen K. , "Components of Software Development Risk: How to Address Them? A Project Manager Survey," *IEEE Transactions on*

Software Engineering, Feb. , vol. 26, no. 2, pp. 98-112, 2000.

- [4] Englund H., "A Case Study to Explore Risk Management Models," M.S. thesis, Royal Institute of Technology, Stockholm, Sweden, 1997.
- [5] Jaana Nyfjord, *Towards Integrating Agile Development and Risk Management*, PhD. Thesis, Stockholm, 2008.
- [6] Beck K., *Extreme Programming Explained: Embrace Change*, 2nd ed., Addison-Wesley, Upper Sadle River, NJ, 2004.
- [7] Ken Schwaber, Jeff Sutherland, *The Scrum Guide, The Definitive Guide to Scrum: The Rules of the Game*, Oct.2011
- [8] Gurdeep Singh, An Informative Research study of Software Development and Testing according to Agile Methodology, *Int'l J. Advances in Computer Science and Communication Engineering (IJACSCE)*, vol. 2, no. I, Mar. 2014
- [9] Enterprise Cloud Development, *An Introduction to Agile Software Development*, White paper, www.collab.net .
- [10] Ana Beatriz ChisteBrandão, *Risk Management in Software projects using Scrum framework*, M.S. Thesis, Stockholm, Sweden 2012
- [11] Dr. Alan Moran, *Embrace Risk! An agile approach to risk management*, Copyright © by the Institute for Agile Risk Management, 2014
- [12] HaneenHijazi, ThairKhdour and Abdulsalam, "A Review of Risk Management in Different Software Development Methodologies," *Int'l. J. of Computer Application (IJCA)*, vol. 45, no.7, May 2012.