

Best Practices for Managing Risk in Adaptive Agile Process

Sunil Kumar Khatri¹, Khushboo Bahri², Prashant Johri³

Amity Institute of Information Technology, Amity University, Noida, Uttar Pradesh, India

Galgotias University, Gautam Budh Nagar, Noida, India

¹sunilkkhatri@gmail.com, ²skkhatri@amity.edu

²khushboo.bahri@gmail.com

³johri.prashant@gmail.com

Abstract— Agile methodologies has given a break from traditional software development models and has proved to be powerful in accommodating requirement changes and induce degree of flexibility in the software development life cycle. These characteristics have made agile an obvious choice among all software development models. One of the main objectives of any model is to cater to risk, identify, analyze, assess and plan. On the contrary agile does not clearly states anything in regard to risk management. This paper targets in analyzing agile methodologies and risk management model, amount of risk handling capabilities in agile and requirement of merging risk management in agile.

Keywords— Agile, Risk Management, Scrum, Sprint, Abstraction Levels

I. INTRODUCTION

Agile software development methodologies are set of software development methodologies designed to cater a great flexibility with respect to changes in external and internal environment of the system. All the agile methodologies provide an alternative to heavy weight traditional process. Agile manifesto define the values of the processes as: [1]

- Individual and interaction over process and tool.
- Working software over comprehensive documentation
- Responding to change over the following plan.

Whatever may be the model in agile, they all abide to agile manifesto principles of light weight documentation [1], iterative and exploratory processes. Through small iterative cycles agile processes provides an alternative to exhaustive process and solutions to many hurdles. Scrum being the most common agile process, which define the iterative development of the product and will be discussed in this paper.

Deviating from the traditional software scrum and other agile methodologies does not define any risk management implementation and practices. Scrum does not specify any engineering practices and leaves on the understanding of the professional implementing the process to add what is missing. [2]

The entire product description called epic, documented as product backlog, is the user requirement in a raw for which is synthesized and created user stories which need to be

developed. These user stories are documented as sprint backlog and tasks are created on the basis of effort estimation. Each sprint last for one to four weeks depending on effort required. At the end of each sprint a sprint backlog refinement is created which takes approval of user requirements and testing, and planning of next sprint. Daily scrum meetings help to discuss the daily target and thus producing more effort efficiency. Scrum defines roles as project manager, development team, product owner (user) and scrum master. User involvement helps increasing the efficiency of understanding user requirement and validating them at each step, which in turn reduces the development time. Defining correct team structure and estimating effort are very important tasks in scrum implementation, which provides the criteria of degree of success of scrum implementation.

Risk management is one of the most critical part of any software project management. Any risk which has some amount of impact and probability is threat to software development, thus identifying, assessing and planning for risk is very crucial. Any software planning model includes risk management to ensure the smooth conduct of development and delivery of project.

There is a huge difference between the traditional and agile model and thus in level of inbuilt risk managing capabilities. Thus this paper identifies the possibility rather requirement of risk management in agile methodologies. A detailed study of agile and traditional processes and discussion with professionals working in live environment implementing agile, has enabled to define how short iterative nature of agile has given solution to risk but yet lacked the structured risk analysis in terms of assessment and ownership.

II. LITERATURE REVIEW

A. Methodology

There is an ongoing research that has performed to identify the types of risk which might occur in agile implementation and way to merge risk management principles in agile (Scrum) process. Due to efficiency of agile processes it is very difficult to clearly state that agile requires risk management principles to be induced. Whereas we can identify the kind of risks which can occur in agile model, due to iterative life cycle and try to identify then as early as possible to reduce its impact,

perhaps which needs an experience and strong knowledge of agile principles.

In this paper, we have tried to study all the research paper published in different journals and conference which define agile process, effort estimation and risk management in agile. On the basis of the inferences derived from each publication we have tried to understand the types of risk which can occur in any software development model. In that basis, analyzed, how have different from traditional mode, agile cater risk management more efficiently and what other types of risk, specific to agile, can cause harm to system.

Also, we have tried to analyze and confirm these inferences from professionals in industry who are practically implementing agile in live scenarios.

B. Comparison Between Agile and Risk Management Model

There are risks related to any system and process, and agile being a software development methodologies are prone to risks which needs to be identified, assessed and planned, to avoid any harm to the system. Any occurrence of potential risk possesses capacity to hinder development and harm to business.

Being lightweight, agile methodologies does not support heavy documentation.[1] Thus there is no specific declaration of risk management in agile exists as risk management is a heavy weight extensive process. There are many contrasts and similarities between risk management and agile principles which are discussed in following paragraphs.

Both the models define in similar manner that is risk is an event which if at all occurs may hinder the success of the project. Risk may have positive impact as well, which is essentially considered in XP. Risk management suggests creation of electronic repository of risk as a part of experience base where as in scrum process daily meetings and iterative lifecycles helps to manage the risk.[3]

Risk management incorporate various tools to identify and assess the type's risk which are expected to occur and their probability and impact as a part of structured analysis. Whereas agile methods do not specifically define and assess risk on the attribute values of the impact and probability. This might lead in overlooking few risks which may be potential risk.

Another important difference is in defining stakeholders. In traditional risk management model, stake holders both external and internal are precisely defined where as in agile, the entire development team and owner take collective ownership. At times occurrence of risk, this may lead to delay or cause some harm if no one person takes the responsibility. [4]

As discussed iterative lifecycle of agile methodologies help to carefully design and restructure the process as required, and thus is capable of handling risk in more efficient manner. In

agile the types of risk are different from traditional risk discussed in many models. Effort estimation and user story prioritization are the most critical objects in agile process. Risk regarding inaccurately estimating effort can cause disaster in development. There are many errors in effort estimation methods like planning poker. Also, it broadly depends on the structure of team, thus defining correct team with experienced people is a very critical factor. Defining team structure known as optimization of velocity, which is the progress rate of team. There are different methods used to estimate velocity (1) based on historical data, (2) execute iteration and (3) forecasting.

III. TYPES OF RISK IN AGILE

Agile methodology is different from traditional software methodologies thus the type of risks present in agile are also very different from the risk that are discussed in many traditional development models.

As discussed above short iterative life cycle of agile, increases the scope of versioning and validating business requirement, and deciding future direction of implementation. This help in identifying error and functional changes in early life cycle of the project. This way it helps in better risk management, whereas there are many other risk which can occur in agile process which are discussed below

A. Team structure/ Optimization of Velocity

Effort estimation, identifying and handling risks will depend a lot on the structure of the team. The efficiency of team increases if team members have prior experience on same kind of projects. Thus defining team structure and members is very crucial in development. Agile team consists of professionals from all the specialization fields to understand the requirement and analyze the effort, duration and cost and for other business analysis.

Optimization of velocity is defined as progress rate of the team, depends team structure.[5] Requirement and effort analysis defines the number of iterations required and their duration. Thus previous experience of team members on the nature of the current project, is an essential aspect which will lead to efficient effort estimation and reduced duration of the life cycle.

B. Effort Estimation

Estimating effort is the most crucial and difficult task in agile processes. There are many effort estimation models but they have some amount of error in each. Estimation of effort for any user story may differ in member of team as per their experience and understanding. One user story may be easy for one of the member but might be difficult for others, this will impact the estimation of effort, duration and cost.

C. Defining Ownership

Agile methodologies work on collective ownership, thus they do not specify clearly the ownership of any task. This indeed increases the sense of sincerity in team members and motivates them to put best of efforts. But this may induce rift

in team members, while identifying mistakes of others. Also at the time of any occurrence of risk, absence of ownership may create disaster.

D. Expatriation level: Understanding agile level

Agile scrum being new, lack professional with high experience rather agile principles can be easily misunderstood[2] as it leave many statement upto scholar/professional understanding. Also, there is no system which is either 100 % agile or no agile. Thus misunderstanding agile principle can be a potential risk to any new system.

IV. PROPOSED AGILE RISK DOCUMENTATION PRACTICES

Scrum process defines an iterative cycle for software development. This iterative life cycle helps in identifying requirement changes, cater to risk management, Estimate duration, effort, and cost more efficiently. Also user participation makes the entire system even more efficient by reducing the error. But scrum process, as discussed above lacks structured risk analysis. Thus following is proposed documentation which will enhance the efficiency of scrum process with respect to risk analysis.

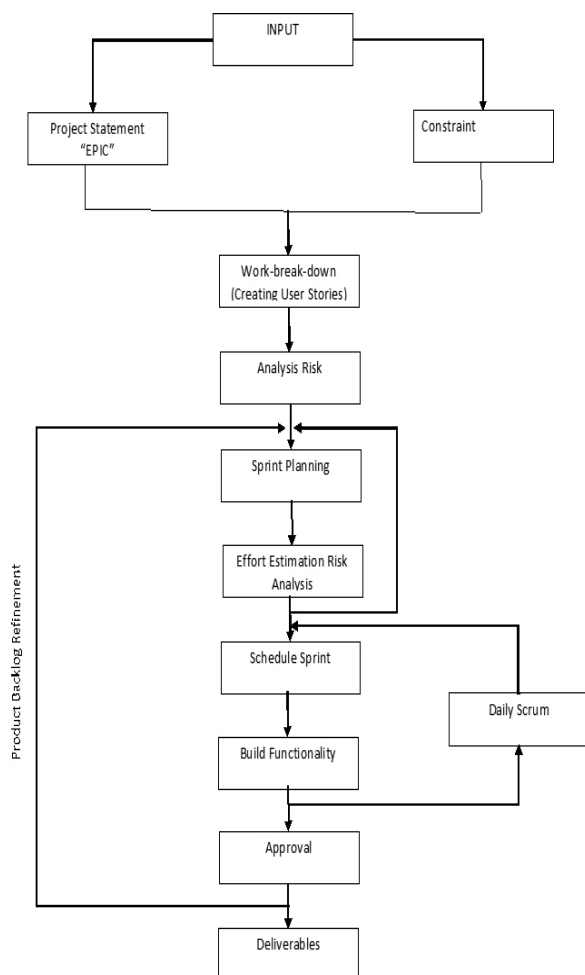


Fig. 1 Flowchart to define risk management in agile process.

Agile scrum development can be divided into majorly four levels of abstractions namely

- Product backlog
- Sprint backlog
- Daily scrum and
- Product backlog refinement

Each level of abstraction targets some valuable output to the system and which proves to be guide in development planning, process and review.

These levels are iterative in nature and thus provide enough amount of opportunities to identify and solve risk, which increases the efficiency of the system to cater risks but fails in structured analysis and assessment of risk.

The entire process is handled in four abstraction level and each should cater risk analysis and documentation. Each level caters to different type of risk handling and stakeholder involved. As per the definition of agile there are only 3 stakeholders of in the scrum process namely the software project scrum master, team and owner himself.

1) *Level-1 Product Backlog Level*: This is the level where all the requirements from owner are detailed. Optimizing velocity and risk to schedule are of major concern which needs to be catered. Also technical risk on what technology is required. Thus using correct techniques to identify requirement and then define team members, will help in correctly estimating schedule and effort.

Stake holders are project management, scrum master and owner.

2) *Level-2 Sprint backlog*: This is the next phase where the iterations are estimated and thus estimating effort and schedule is of major concern. Also technical risks like, introducing new technology by training or new resource hiring should be decided and analysed. Stakeholder is the team entirely which is involved in the planning of the project.

3) *Level-3 Daily Scrum*: Daily meeting are very efficient in finding challenges which may have potential risk,

These risk are regarding effort, human resources, which can hinder schedule. Also inaccuracy in effort estimation can prove to be efficient risk.

Stake holders are the internal team members who are involved in the development and the scrum master.

4) *Level 4: Product Backlog refinement*: Helps in identifying the correlation between user requirement and deliverables designed. This helps in the identifying risk of misunderstanding user requirement critical functional points. Stakeholders include the team and the owner along with scrum master.

The entire risk analysis should be documented by using risk matrix and by different stakeholders at different abstraction levels.

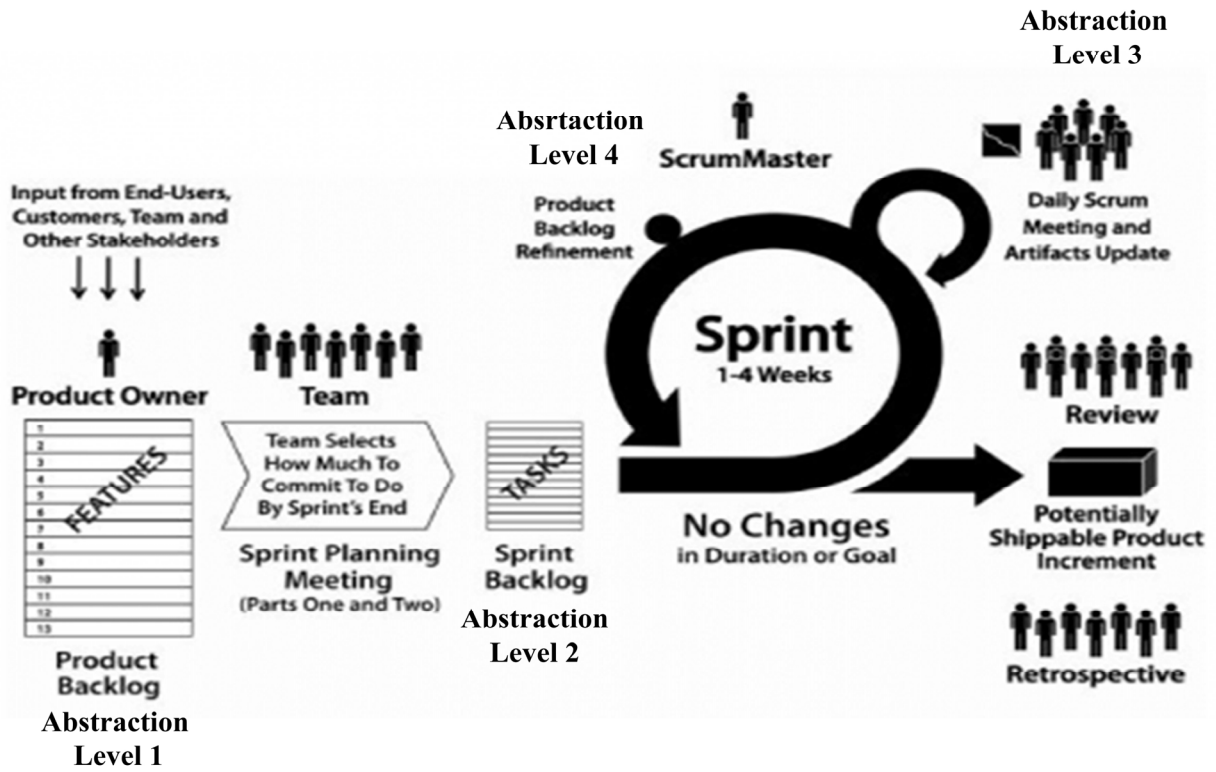


Fig. 2 Abstraction level in scrum process for analysing and documenting risk [10]

These all risks, as per definition of risk are potential to harm system if occur. Each abstraction level includes a sequence of meetings and activities. Thus there is no additional much of effort and time is required, risk activity can be included during all other activities at each abstraction level.

These risks, analyzed in meeting can be documented using a simple format of matrix, which defines the risk definition, size, impact and probability. Each member included in risk management can assess each risk on the scale of one to ten, and a cumulative of all is actual impact of the risk.

$$\text{Impact} = \sum_{i=0}^n \text{Size}_i \times \text{Probability}_i$$

Where i = Number of Stakeholders in the meeting

Whereas this activity can be more efficient if it evolves the concern to solve the risk rather than analyzing the impact. Thus any sort of risk pointed by any member can be documented in the same pattern and also the feasible solution.

This will help in recognizing the possible risks which might occur in future and steps to manage and handle these risks, along with their size, probability and impact.

V. CONCLUSIONS

Agile processes were defined to cater to drawbacks of traditional models. Iterative and incremental life cycle has made the agile processes much more efficient in all respect of software development process i.e. catering to requirement changes, estimating effort and budget more efficiently and reduced risk. To define any software development process 100 % exclusive of any kind of risk is too idealistic statement. Thus even agile processes, scrum are prone to risk which needs structured analysis and efficient resolution.

In this paper we have tried to derive the requirement and possibility of merging risk management in agile model. We have studied the agile and traditional software development and tried to define the type of risks which can exist in agile process. Also defined the level of abstractions in scrum process and risk analysis levels and defined the stakeholders at each level. These define the best practices to document risk in agile process and proposed guidelines will help in managing risk in much more structured manner than earlier.

ACKNOWLEDGMENT

Authors express their deep sense of gratitude to The Founder President of Amity University, Dr. Ashok K. Chauhan for his keen interest in promoting research in the

Amity University and has always been an inspiration for achieving great heights.

Also Khushboo Bahri likes to acknowledge Mr Sukhmeet Sethi for his interest, support and direction to achieve the objective of the paper and helped to attain industry and practical knowledge for the evaluation of the same.

REFERENCES

- [1] Kaushal Pathak, Anju Saha, "Review of Agile Software Development Methodologies", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 2, February 2013, pp. 270-276.
- [2] Radha Shankarmani, Renuka Pawar, S. S. Mantha, Vinaya Babu "Agile Methodology: Benefits and Constraints", International Journal of Computer Application (0975-8887), Volume 58- No.15, November 2012, pp. 31-37.
- [3] Jaana Nyfjord and Mira Kajko-Mattsson, "Commonalities in Risk Management and Agile Process Models", International Conference on Software Engineering Advances (ICSEA 2007), pp. 18.
- [4] Ville Ylimannela, "A MODEL FOR RISK MANAGEMENT IN AGILE SOFTWARE DEVELOPMENT"
- [5] Evita Coelho, Anirban Basu, "Effort Estimation in Agile Software Development using Story Points", International Journal of Applied Information Systems (IJ AIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 3– No.7, August 2012, pp. 7-10.
- [6] Granville G. Miller, "The Characteristics of Agile Software Processes", Computational Intelligence for Modelling Control & Automation, 2008 International Conference, 10-12 Dec. 2008, pp. 49 - 54
- [7] <http://www.ibettertechnologies.com/process.html>
- [8] Danh Nguyen-Cong, De Tran-Cao, "A Review of Effort Estimation Studies in Agile, Iterative and Incremental Software Development", IEEE RIVF International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for the Future (RIVF), 2013, pp. 27-30.
- [9] Pekka Abrahamssona, Juhani Warstab, Mikko T. Siponenb and Jussi Ronkainenena, "New Directions on Agile Methods: A Comparative Analysis", Published in the Proceedings of the International Conference on Software Engineering, May 3-5, 2003, Portland, Oregon, USA, pp. 244-254.
- [10] Ziauddin, Shahid Kamal Tipu, Shahrukh Zia, "An Effort Estimation Model for Agile Software Development", Advances in Computer Science and its Applications (ACSA) 314 Vol. 2, No. 1, 2012, ISSN 2166-2924, pp. 314-324.