IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. VI (Sep – Oct. 2014), PP 43-46
www.iosrjournals.org

Risk Management in Agile Model

[1]Monika Singh, [2]Ruhi Saxena
Faculty of Engineering (FET), Mody University of Science and Technology Rajasthan, India

Abstract: *In traditional waterfall-model risks were usually managed by using project risk management frameworks. Nowadays agile methods have started replacing the traditional models. Agile development is based on short iteration cycles, which allow them to respond to changes in business environment. This paper focus on use of risk management in AGILE model.*

Keyword: *Agile Model, Scrum, Risk, Risk Management.*

I. Agile Model

Agile is an iterative, team-based approach to development which emphasizes the rapid delivery of an application in complete functional components. Rather than creating tasks and schedules, all time is "time-boxed" into phases called "sprints." Each sprint has a defined duration (usually in weeks) with a running list of deliverables, planned one sprint in advance. Deliverables are prioritized by business value as determined by the customer. If all planned work for the sprint cannot be completed, work is reprioritized and the information is used for future sprint planning. As work is completed during each sprint, it is continuously reviewed and evaluated by the customer, who may be considered the most critical member of the AGILE team. As a result, Agile relies on a very high level of customer involvement throughout the project [5].

There are various methodologies that are collectively known as agile, as they promote the values of the agile manifesto and they are consistent with the above principles. The most popular ones are:

DSDM (Dynamic System Development Method)-is probably the original agile development method. DSDM was around before the term 'agile' was even invented, but is absolutely based on all the principles we've come to know as agile. DSDM seems to be much less well-known outside of the UK [4] [7].

Scrum- is also an agile development method, which concentrates particularly on how to manage tasks within a team-based development environment. Scrum is the most popular and widely adopted agile method – I think because it is relatively simple to implement and addresses many of the management issues that have plagued IT development teams for decades [3].

XP- (Extreme Programming) is a more radical agile methodology, focusing more on the software engineering process and addressing the analysis, development and test phases with novel approaches that make a substantial difference to the quality of the end product [8].

II. Risk

A risk has three essential components: **uncertainty, loss and finite duration**. There's always uncertainty as to whether a risk will occur. If an item is certain to occur, we instead call it an issue. Issues are just as important as risks, but we manage them differently. Because a risk may not occur, we have a powerful means of managing it; namely, preventing it from occurring. Risk always involves loss: loss of time, loss of money, loss of market share or some other loss that the project will suffer if the risk occurs. The combination of the likelihood that the risk will occur and the subsequent loss involved determines the risk's severity.

III. Risk Management

In modern software projects, security and risk management are not just something one might do if there are time and resources. Security has become an important part of the end product.

This means risk management must be introduced at the beginning of the project, and risks must be evaluated and assessed during the whole development cycle. Agile software development methods are focused around delivering the maximum benefit to a product owner.

Effective risk management involves:

1. Identifying the risk.
2. Analyzing each risk to determine its severity.
3. Prioritizing the identified risks based on their severity.
4. Creating action plans (responses) to deal with the high-priority risks.
5. Continuous monitoring and follow-up to ensure that your action plans are mitigating the risks [2].

IV. Risk Management In Agile Model

All major risk management phases described in PMBOK (project Management Body of Knowledge), from planning risk management to monitoring and re-evaluating risks, are included in the model. The model revolves around a risk board, which is used and updated constantly throughout the software development cycle.

A. Risk Board

Traditional project management techniques would recommend a risk register to monitor for managing and controlling risks. I prefer to keep a simple risk register with concise information. Too many fields in a risk register complicate the process and its administration. A simple risk register should consist of the following attributes:

- Description of risk: A one- or two-line overview of the risk. It should be simple and easy to comprehend.
- Date identified: Date when the risk was identified.
- Likelihood: Estimated probability of occurrence of the risk.
- Severity: The severity of the risk is assessed based on impact of the undesired outcome.
- Priority (optional): This could be either given an independent value or set as a product of likelihood and severity (above). A high-severity risk with a high likelihood should receive more importance than a high-severity risk with a low likelihood.
- Owner: The person who manages, controls, and takes action in response to the risk.
- Action: The response defined to manage/control the risk.
- Status: Indicates whether the risk is open or closed or being monitored.

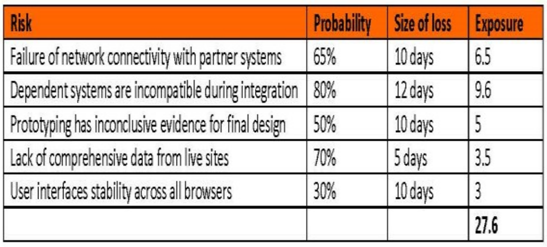
It is imperative that the risk register be made available for the team so that it can be managed and monitored collaboratively. At every sprint meeting, the risk register must be reviewed and updated with any new information obtained over the sprint. This way risk management becomes an integral part of Agile [4].

Another interesting technique, introduced by John Brothers (Agile Times, 2004), relies on using a Risk Burn-down Chart. As elaborated by Mike Cohn in his article, the risks are collated into a table similar to a risk register. It consists of the following elements:

- Risk: Description of the risk in a few lines.
- Probability: Likelihood of the risk.
- Size of loss: Amount of time lost should the risk occur. This could be represented in days or story points.
- Exposure: This is computed as a product of the probability and size of loss (above).

An example is shown below. The consolidated risk exposure is shown in the final row of the risk register.

Table 1: Risk Exposure



Risk	Probability	Size of loss	Exposure
Failure of network connectivity with partner systems	65%	10 days	6.5
Dependent systems are incompatible during integration	80%	12 days	9.6
Prototyping has inconclusive evidence for final design	50%	10 days	5
Lack of comprehensive data from live sites	70%	5 days	3.5
User interfaces stability across all browsers	30%	10 days	3
			27.6

This risk register is reevaluated at every sprint meeting; its values are adjusted based on the current assessment of the existing and new risks. This would define a new value for the consolidated risk exposure [4]. The Risk Burn-down Chart can be created by plotting the consolidated risk exposure across the number of sprints run by the team. An example is shown below:

Table 2: Risk exposure across the number of sprints

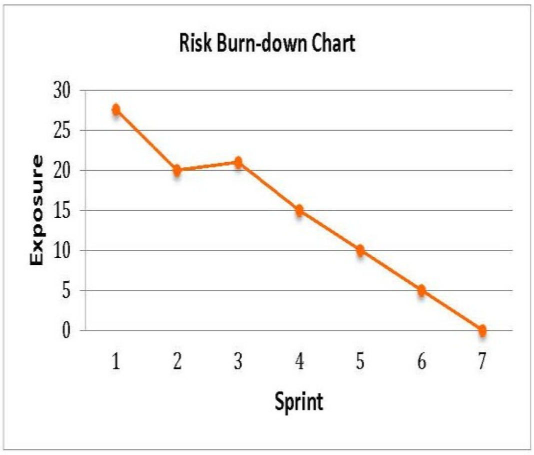


Figure 1: Risk burn-down chart

In essence, the burn-down chart represents the status of the risk across the iterations. From a project management perspective, this is an excellent indicator of how the risks are managed and controlled. Similar to other burn-down charts, the ideal burn-down would be a linear decrease of consolidated risk exposure over the sprints.

These are some of the bare-minimum approaches that are effective in Agile. They're simple to manage and align with the iterative process. Adopting one of these is critical to having a proactive risk management approach.

B. Process

The flowchart presents the idea how risks are identified, assessed and approved in the model.

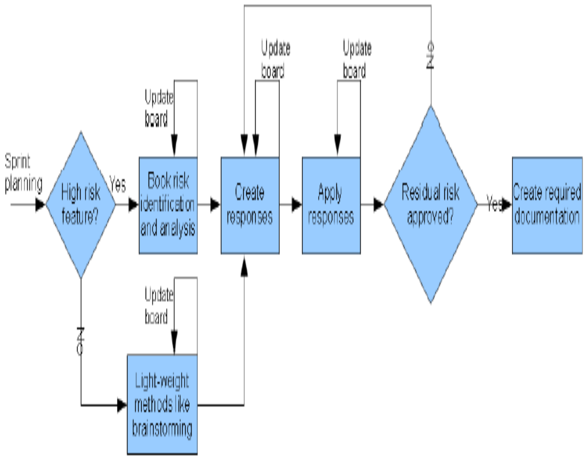


Figure.2: Flowchart Of Process.

The flowchart does not include risk management planning, which is only done once as a part of project visioning. It also doesn't include risk monitoring. The table 1 describes what additional things are done during the Scrum meetings. This does not mean that risk identification, assessment and responses would only be done during these meetings; most of the work will be done between the meetings. X means that it should be done every time; P means it is possible if needed [6].

Table 1: Additional Risk Management Activities in Meetings

Risk Management in meetings	Identification	Assessment	Create Response	Apply Response	Risk Approval
Sprint Planning	X	X	X		
Daily Scrum	P	P	P		

Sprint review	P	P	P		X
Sprint Retrospective					

Risks can be identified during all meetings apart from retrospective. As a risk is identified, one should also suggest the impact and probability it might have. Even if a feature receives quick risk identification during sprint planning, this still means that further identification and assessment should be done by the person responsible for the feature. Identifying and assessing new risks during the sprint review is possible, but applying any responses will be a task for the next sprint.

V. Benefits of choosing Agile

Agile methodology focuses primarily on providing value, while maintaining safety for organizations to implement solutions. Companies choose Agile because:

- In long-term projects, it can quickly respond to changes in business.
- Agile is based on the realities of the requirements of the organization, not imaginative over-analysis.
- By deep collaboration with customers, and readiness to change (within reasonable budget), Agile promotes a win-win philosophy [1].

VI. Conclusion

Using AGILE methodology reduces risk in early phases of software development, which is otherwise considered during project management in traditional model. Moreover Agile is based on the realities of the requirements of the organization, not imaginative over-analysis.

References

[1]. http://resources.infosecinstitute.com/risk-management-agile/
[2]. http://www.polarisft.com/downloads/whitepapers/Risk-Management-in-Agile-Development.pdf
[3]. http://www.allaboutagile.com/what-is-agile-10-key-principles/#sthash.4OEnczZs.dpuf
[4]. http://www.scrumalliance.org/community/articles/2013/2013-may/risk-management-in-agile
[5]. http://www.seguetech.com/blog/2013/07/05/waterfall-vs-agile-right-development-methodology
[6]. http://www.cloudsw.org/under-review/a6f468c9-4857-4206-96ee f67df0583d41/file_initial_version.
[7]. Stapleton J (1995) DSDM −Dynamic system development method. Addison-Wesley, UK.
[8]. Beck, K, Extreme Programming Explained: Embrace Change, 2001, Addison Wesley.