A SURVEY ON SLIM PROCESS WITH DEVELOPERS IN AGILE SOFTWARE

Dr. Prabhu J*
Associate Professor, SITE, VIT University, Vellore, TamilNadu, India.
Email: jprabhuit@gmail.com

Received on 22-07-2016
Accepted on 26-08-2016

Abstract

In recent years there has been a noticeable shift in attention from agile software development toward slender software system development, Agile software development changes the nature of, coordination, collaboration and communication between customers in software projects, which is often labelled as a shift “agile challenges to slender”,” agile decision to software system developers”. More approaches published in agile development process have been inspected.

As a result, this paper provides a consistency in software development and team decision is acceptable. Therefore, feedback from customer is needed in order to increase decision making with team members based on agile challenges to slender process.

Keywords: User Requirements, Project Team size, customer Involvement, challenges in agile, agile development process.

1. Introduction

Agile methods have evolved as a powerful software development approaches. Agile methods focus mainly on an incremental and iterative development process in which there is a continuously evolve of requirements, specifications, design, implementation, and testing. The critical query in software development project is how to complete a project at a specific duration, Budget and resources. To ensure and measure these three attributes are achieved, the individuals who involved in estimation process especially project manager need to ensure all requirements are met and well-defined. Project schedule overrun is one of the main contributors to project failure [26,27]. However, software development community believes that agile methodology that can assist them in the cost management process. This is often explained with their potential to overcome the challenges of modern software organizations which are expected to operate in highly dynamic and competitive environments.
In such environments, speed, quality and cost of software development are crucial for organizational survival and agile method seems to be successfully delivering on all three fronts through their customer focus, responsiveness to change, iterative and incremental delivery of working software and emphasis on individuals and their interactions [31]. The Requirements Engineering (RE) practices such as observations, interviews, workshops and strong team collaboration are embedded in iteration-based agile methods [33]. Similarly, Requirement Engineering practices such as customer involvement, requirements prioritisation, requirements modelling, requirements documentation, have also been suggested to be used with agile methods.

2. Literature Survey

The main goal of testing is the detection of defects. Developer testing adds to this the ability to point out where the defect occurs [1] fault-based test adequacy criteria. These techniques can be applied to acquire test effectiveness indicators. Error seeding is the technique of planting artificial errors in a software system and subsequently testing the system against these artificial errors and counting the successfully detected ones [2].

Crowd sourcing can be incorporated into conventional software development processes such as waterfall or agile and can contribute to any software development phase. It can be used with most design techniques, such as object-oriented, service-oriented, and user-centred design (UCD); Software-as-a-service (SaaS); and formal methods [3].

Agile software development (ASD) is a lightweight paradigm for software development that emerged in response to the limitations of the prescriptive methodologies, e.g., the waterfall, spiral approach [4].

Agile methods comprise a set of underlying values and principles that are in line with this category. Highlight that agile teams require courage, respect, communication, simplicity, and feedback [5].

Agile methods put a strong emphasis on constant communication and coordination between team members, particularly face-to-face interaction [6]. Likewise, we have observed communication and coordination across teams occurring similarly through face-to-face conversations, collective meetings, collective informative workspaces, tools or frameworks to support automated tests, integration tests, and continuous deployment [7].

In agile software development, intra-team knowledge sharing is regarded as important for accomplishing specific project tasks and for offering the opportunity to discover creative means to improve the organization’s competitiveness.

The agile culture helps nurturing a natural environment for intra-team knowledge sharing, which is achieved by emphasizing face-to-face conversations through agile practices and by using minimal documentation [8].
Requirements Engineering (RE) practices such as observations, interviews, workshops and strong team collaboration are embedded in iteration-based agile methods [9]. Likewise, RE practices such as customer involvement, requirements prioritisation [10], requirements modelling [11], requirements documentation [12], have also been suggested to be used with agile methods. In general, have the potential to make more effective decisions than individuals because teams can pool knowledge and information, which helps them to make a good decision [13].

Agile software development literature by analysing decisions made during the iteration cycle and identifying six key obstacles indicate the six decision obstacles are unwillingness to commit to decisions; conflicting priorities; unstable resource availability; and lack of: implementation; ownership; empowerment. These six decision obstacles are mapped to descriptive decision making principles to demonstrate where the obstacles affect the decision process. The effects of these obstacles include a lack of longer term, strategic focus for decisions, an ever-growing backlog of delayed work from previous iterations, and a lack of team engagement [14].

The software development agility as the software team’s capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle [15].

The client–developer collaboration is also a well-recognized feature of most agile requirements engineering (RE) processes and specifically – of requirements (re)prioritization. Reprioritizing requirements at inter-iteration time plays a pivotal role in agile projects [16].

The delivery stories play a pivotal role in requirements prioritization and vendor’s domain knowledge is a key asset for setting up successful client-developer collaboration. The use of agile prioritization practices depends on the type of project outsourcing arrangement [17].

Over the years agile methods have proven to overcome many of the problems stated above and have become dominant in the software industry. The agile approach is basically driven by self-organizing teams that have the power to coordinate their work on their own. This increases productivity, enables employees to learn, innovate, and finally makes them happy with what they do [18].

In the traditional plan-driven (waterfall) software development processes, work is coordinated by managers and there is a clear separation of roles [19].

[20] Notifies cluster analysis to analyse quantitative data and triangulated the results with content analysis of the qualitative data. We then proposed a new definition for agile software development maturity. The findings show that practitioners do not see maturity in agile software development as process definition or quantitative management.
capabilities. Rather, agile maturity means fostering more subjective capabilities, such as collaboration, communication, commitment, care, sharing and self-organization. The integration of usability activities into the agile software development process has been highlighted as a means through which these tensions might be resolved. This integration is argued to be possible since both agile methods and usability activities focus on delivering value, are iterative in nature and engage in continuous testing [21,22].

The unit of analysis was the agile software development team. There were no specific choice criteria for teams: they could have any size, apply any method, no matter how long agile had been adopted. We were searching for a pattern of evolution in agile teams and, the most diverse these teams were in context, the better. The lack of preference for a specific agile method, such as Scrum or XP, is due to the fact that agile methods are currently being highly tailored [23] and current and future research in agile methods are expected not to focus on a single method [24]. It claims that experience gathered during large scale implementation of agile concepts in software development projects teaches us that the currently popular agile software development methods (like Scrum) do not scale to programme, product and organisation level without change [25].

Several researchers consider the alignment of managerial decisions in organizations, and stress the importance of ensuring that product and project level decisions are well aligned with the organizational business strategy [28, 29]. Since moving from a plan-driven approach to an agile approach is a part of business strategy, it is important that software practitioners understand their role in this strategy. A failure of such understanding may easily lead to the failure of the agile strategy itself. Hence, this research is based on the assumption that the alignment of product and project decisions with business strategy is important for the successful application of an agile approach in software development.

Team performance models aim to describe causal relationships between variables that result in performance outcomes or at least provide actionable advice for managing performance. [32] Discuss three teamwork models concerned with team effectiveness from an internal perspective.

Agile methods promote in relation to traditional software development also impact requirements engineering activities to a certain extent. Many researchers have been keen to gain a better understanding of such impacts and how agile methods and requirement engineering are now related; the investigation of these researchers has focused on this interconnection. The agility and dynamic nature that allow changes based on constant feedback from stakeholders cause the emergence of requirements throughout the development process [34].
Conducted a systematic literature review on the topic of the integration of agile methods and user centred design approaches. The review focused on usability issues in agile methods with respect to design. The findings show that usability issues in agile methods can be addressed by incorporating a user centred design specialist (UCDS) role in agile teams. The authors also defined practices to resolve usability issues in agile methods such as Little Design Up Front, Big Design Up Front, low fidelity prototypes, user testing, interaction models, and close collaboration.

**Proposed work**

![Figure-1. Architecture for slender process in Agile.](image)

**Agile challenges to slender**

The Five inherently interlinked guiding slender concepts underpin slender process are as follows [30]

- Assessment flows
- Tweak
- Excellence
- Quality
- Customer feedback

**Agile decision to software system developers**

The various components are used in during agile software development are as follows

1. Project team leadership.
2. Performance Assessment.
3. Consistency.
4. Customer feedback
5. Team co-ordination
6. Open loop communication.
7. Self-decision and organisation.
8. Problem identification
9. Solution development
10. Sprint planning and delivery.

Conclusion
The study revealed two major characteristics of an agile development, slender process and agile decision to software developers including value based, assessment flows, tweak, and excellence. The agile development process can illustrate the proposed framework. In this model framework, consistency, team leadership, performance assessment and adjustment should be performed in turn through an agile development cycle. The slender process and decision in software development can be used to implement this model framework in real environments. In this, model framework is flexible for utilisation, easier to use and has less overhead by all organisations when compared to the previous frameworks.

References


Corresponding Author:
Dr.Prabhu.J*
Email: jprabhuit@gmail.com