Importance of Teamwork and Design in Agile Software Development

Mayank Rawat  
 Software Engineering  
 Arizona State University   
 Tempe Arizona US  
 msrawat@asu.edu

Vaibhav Bhasin  
 Software Engineering

Arizona State University  
 Tempe Arizona US  
 vbhasin2@asu.edu

ABSTRACT

Agile method has been quite popular in software engineering for over a decade. This article will try to explain the effect of agile teamwork quality on the team performance and how the effects differ from that of a traditional software team.

INTRODUCTION

As the word “Agile” suggests, agile software development means development of software where requirements and functionalities of a software change continuously and regularly. It refers to a group of software development methodologies based on iterative development, where requirements and resolutions evolve through collaboration between cross-functional agile teams[1].

Before the evolution of agile software development, the Waterfall methodology was the standard for software development. This process required documentation up front before any coding was done. Business requirement document was written first that contained all the requirements that the business needed in the application. These documents were long with every aspect of the application explained in detail. This documentation was then followed by some non-functional requirement and user-interfaces and the coding will finally kick off. Coding would be followed by integration and then testing. If in between this process there was a change in the requirement then it would take a long process to review the change, communicate with other teams and fixing the code would be expensive.

However, the agile methodology focused more on documentation, self-organization rather than rigid management practices, and the ability to manage constant changes. This required collaborative work and high customer involvement.

**TEAMWORK IN SOFTWARE DEVELOPMENT** Teamwork in agile software development is very important and plays a key role. The following are the various factors that describe a teamwork [1].

**1.Teamwork Quality**

Teamwork quality is a construct with many sub-constructs [1].

**1.1. Communication**

Communication between the teams must be frequent, open and clear so that the information can be shared without any confusion [1].

**1.2. Coordination**

There should be common understanding among the team members when working on parallel subtasks, and agreement on budgets and deliverables. Many activities and task processes are delegated to different team members and synchronization of these activities is important for a successful project [1].

**1.3. Balance of Member contribution**

The ability to bring out the team members full potential. Contribution should reflect every individual’s specific knowledge and experience [1].

**1.4. Mutual Support**

Team members willingness to support and help other team members in carrying out their tasks. Some agile development methods include collective code ownership, which in turn stimulates mutual support and collaboration [1].

**1.5. Cohesion**

Team members motivation to maintain the team and realize that team goals are more important than individual goals [1].

Team should work in a way that increases the motivation of the team members and their ability to engage in future teamwork. Collaborating with other team members also provides the opportunity for learning their technical and creative skills. Teamwork has a huge contribution in the success of a project and its outcome.

SOFTWARE DESIGN AND ARCHITECTURE IN LARGE SCALE AGILE PROJECTS

Software design has been the focus of software engineering since its beginning. Majority of the software engineering research arguably direct at improving the ability to efficiently come up with software design and to meet the challenges that accompany. Design remains the focus of software engineering [2]. Herb Simon, in his *‘The Sciences of Artificial’*, talks about how software design provides an interface between the inner environment (The tasks, requirements) and the outer environment (the means, software languages).

**Software design in agile**

Agile methods however, in terms of the process of software design can be considered as a step backwards in software design research as the design implements only in the code. Agile emphasizes mainly 3 design practices [2].

1. By involving user continuously into the iterative development process the product is continuously shaped and designed according to user’s needs.
2. Test driven development, which emphasizes functionality design as much as system structure design.
3. Lastly, continuous design through which design is a part of the complete product development lifecycle.

When we talk about the architecture of the software, we mainly focus on how the components of the software are laid out and how these structures shape the software’s design decisions. Many strands of work in software design have blended into the field of software architecture. Therefore, software architecture is a set of principle design decisions governing a system.

**Software architecture approaches for multiple Agile teams**

Agile methodologies are believed to have best suited to small, co-located teams but this has also led to inspiring its use in large scale development [3]. In terms of software architecture, several approaches have been taken up in companies with multiple agile teams working on single software.

The team can start with a large up-front design and follow Agile on later stages.

Some teams have found to focus on deciding on architecture in the first iteration.

Other start development directly and let the architecture emerge itself through iterations.

For large scale projects involving multiple teams, it’s important that a common architecture be agreed upon and communicated through the different teams without overhead.

**Inter-team Coordination**

Inter-team coordination can be defined as managing of dependencies involving tasks, knowledge, decisions among the teams working on the product. For this concept of “*Scrum of Scrums*” can be used for the project, which is a technique to scale scrum to bigger groups and each group are divided into agile teams of 6-8 members [4]. The structure could be represented as in figure 1. [5]

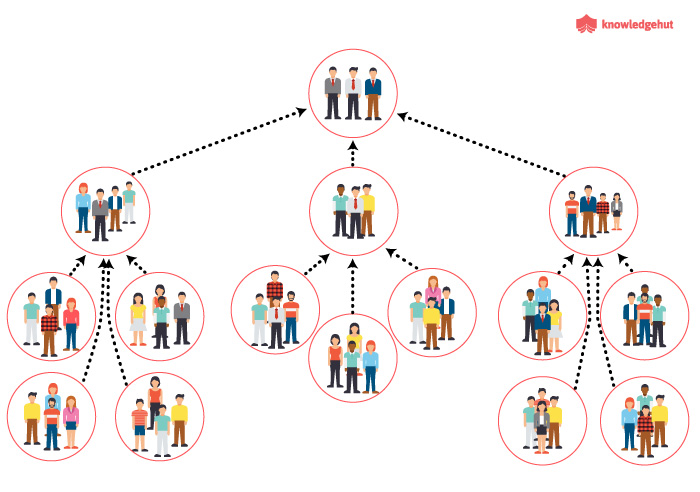


Figure 1: Scrum of Scrums organization

REFERENCES

[1] Teamwork Quality and project success in software development: A Survey of agile development teams.

[2] Software Design and Architecture The once and future focus of software engineering

[3] Software Architecture as a Set of Architectural Design Decisions.

[4] Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation <https://link.springer.com/article/10.1007/s10664-017-9524-2?shared-article-renderer#Sec2>

[5] <https://www.knowledgehut.com/tutorials/scrum-tutorial/scrum-of-scrums>, Scrum of Scrums

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00