Chapter 28
Scrum Framework Based on Agile Methodology in Software Development and Management

Upasana Ameta, Mayank Patel, and Ajay Kumar Sharma

Abstract With the advancement in technology, the demands for software has also increased. Companies are applying different methods to increase performance and business-related profits. The methodology employed in software development is software development life cycle (SDLC), but with the developing technology and growing market, the software-related requirements changes rapidly. So, another methodology called "Agile" came into existence. This is an improved version of SDLC which increases speed, flexibility, and responsiveness in software development. Scrum is a framework, that is based on Agile methodology, that follows a repetitive and incremental procedure. This paper starts with the introduction of Agile methodology and scrum, followed by procedure and description of Scrum values.

Keywords Agile methodology · SDLC · Scrum · Sprint · Product backlog

1 Introduction

To meet the need of growing market demands, companies are required to deliver reliable, economic, and quick-delivering software. Also, companies attempt to limit the development time and maximize their business profit. Software development can be understood as creating a new software to take place of an old one or rectify the existing one.

For developing faster and accurate software, there is a need for some kind of methodology [1]. A software development life cycle (SDLC) is used to create high-quality software in minimum time [3]. SDLC is used for software development, as well as maintenance [1].

Agile method is also one method that can be employed in software development. It is a kind of small-term software development methodology that mainly focuses on how quickly a system can adapt changes. Agile methodology focuses more on communication than the procedure, a basic working software is more important than

U. Ameta (✉) · M. Patel · A. K. Sharma
Geetanjali Institute of Technical Studies, Udaipur, India

R. Mathur et al. (eds.), *Emerging Trends in Data Driven Computing and Communications*,
Studies in Autonomic, Data-driven and Industrial Computing,
https://doi.org/10.1007/978-981-16-3915-9_28

a full-fledged system. The most important aspect is—'being responsive to change [1]. It breaks the project into small modules and these modules are executed iteratively. The main difference between traditional SDLC and Agile methodology is the client involvement and responsiveness [4].

Scrum is a framework under the Agile methodology that works on the concept of repetition and increment [5]. Scrum aims of delivering high-quality software in the shortest time, which are flexible enough to incorporate changes. Scrum breaks the project into small modules that are mini-projects in themselves, thus opposing the traditional sequential approach [4].

2 Literature Review

2.1 Software

A computer program that is responsible for interaction between the user and a variety of hardware is called software. Software translates the commands of the user into an understandable format for the hardware. So, it basically serves as a "translator" between the two [1]. Software is made up of a number of programs, functions. and procedures including the data. The demand for software is increasing and the quality of the software is very important for developers as well as customers [2].

2.2 SDLC (Software Development Life Cycle)

SDLC is a life cycle for developing a software by following several steps [1]. For developing software in a systematic and sequential way, the SDLC is very useful [6]. This life cycle enables us to develop quality software in given timeline according to the requirements and also for maintaining it [1]. Following are several stages of SDLC:

1. Survey and feasibility study
2. Analysis
3. Determine user requirements
4. Finding best possible solutions
5. Requirements gathering (hardware and software)
6. Designing
7. Coding
8. Implementation
9. Maintenance

From the several models of SDLC, the most-used model is the waterfall model. Other than this, fountain, spiral, rapid, incremental, prototyping, etc. are the SDLC models [1].

2.3 Agile

An enhanced version of SDLC called the "Agile methodology" was first introduced in 2001 [2]. The word Agile means quick, active, brisk, and lightweight [1]. Agile methodology is a way to develop software by breaking the project into small, time-boxed iterations. Each iteration undertakes the pieces of project development in a specific duration [4]. This type of development process promises rapid delivery of software, inculcate changing requirements, enhances communication, measures progress from time to time, and ensures client interactions [2]. An important aspect of Agile is that it reduces risks because projects are not carried out in one go, in fact, they are built in pieces, which decreases the chances of entire system failures. Rapid applications Development (RAD), Dynamic system Development method (DSDM), and SCRUM are the frameworks that comes under Agile Software Development [4].

2.4 Traditional Versus Agile

Agile methodology is just a variation in SDLC with the agenda to develop flexible software in a given time. Agile does not follow the hierarchical procedure of conventional SDLC, rather it tends to work in an interactive and iterative format. The most important difference between the two is that agile methodology is more into collaborating with clients and is responsive to change [4].

Agile software development is the latest trend in the industry and 65% of the industry have moved from traditional SDLC to Agile development [4]. However, this development process also has some limitations. Agile methodology works best for small and medium scale projects, but for large scale projects, companies still prefer to use traditional approaches [7].

Both approaches have their merits and demerits which needs to be considered. Refer to Table 1. For a detailed comparison between Traditional Software Development approach and Agile Software Development approach.

2.5 Scrum

Jeff Sutherland developed the concept of Scrum in 1993. Scrum is a framework used in software development and management. It uses values and principles of agile methodology to develop responsive and efficient software [1]. The area of focus for

Table 1 Comparison between traditional SDLC and agile methodology

	Traditional SDLC	Agile
User requirements	All the requirements mentioned by the customer are well-studied before implementation starts	Also welcomes late changing requirements by clients
Responsiveness	Does not incorporate changes	Is flexible and adapt changes easily
Testing	One time testing is done after coding of the entire project is completed	Testing is done iteratively after each module completion
Client involvement	Not much involvement is seen	Clients are invited to see progress from time to time
Focus	Complete documentation by sequentially following the procedure	Developing a working software with maximum interactions

the Scrum framework is to develop flexible, high-quality software, deliver it in the minimum possible time and increase the business value [3]. It does not follow the conventional sequential approach, rather the entire team works together in achieving a common objective [1].

The key principle of Scrum is that it welcomes late requirement changes. This means that if the customer changes his mind and asks for adding some new features during the project development, it is readily acceptable by the developers [2]. Scrum encourages self-organizing teams by maximum casual and verbal interactions between team members and maintaining discipline throughout the project. It follows an empirical approach to meet the responsive requirements [4].

It is a kind of "Inspection and Adaption" framework. This means that the software development process is inspected on a timely basis and if any changes are needed to be incorporated into the system, they are adapted quickly [5]. Scrum framework is a repetitive life cycle for developing high-quality and responsive software. It can be defined as a combination of a set of activities or iterations that are undertaken using a variety of tools and techniques.

3 Methodology

3.1 Scrum Roles

For carrying out the process, some specific roles are required as mentioned below

Product Owner. The role of the product owner is to identify and illustrate the characteristics and specifications of the software that has to be developed [1]. The product owner has the authority to decide the priority of the features that need to be built by the team members [3]. The product owner puts all the basic initial requirements of the project into the product backlog [1]. This person directly communicates with the customer to understand the software requirements and demonstrate these details to the SCRUM team. It can be said that the product owner is the mediator between customer and developer [2].

Scrum Team. The product backlog developed by the Product Owner is executed by the Scrum team. The user stories from the backlog that need to be worked on, in the following, sprint may be allotted to different team members according to their skills, knowledge, and understanding [2]. A Scrum team may consist of 4–9 people, who are responsible for running the project [4].

Usually, a Scrum team comprises of following people:

Project Manager. A person who is responsible for monitoring and controlling the entire project.

Business Analyst. A person who is responsible for analyzing the business.

System Analyst. A person who conducts feasibility studies and analyses for the software.

Programmer. A person who designs the system, writes the code, and undertake internal testing.

Tester. A person who runs and check the program, whether it is appropriate against the specifications or not [1].

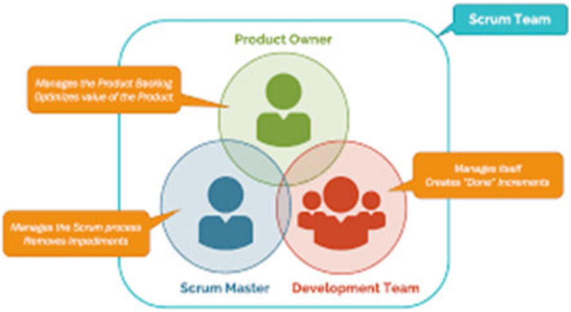
Scrum team members should be cooperative so that there is a state of well-communication among the team and it encourages teamwork [5].

Scrum Master. Scrum Master is responsible for introducing the Scrum process to the team members. He is the one who looks after the implementation of SCRUM process within the team during the entire project development [1]. Scrum Master is a role in the Scrum framework that coaches the team in developing the software using SCRUM values. Scrum Master also solves the issues that are encountered during daily Scrum meetings. This person is more into training than controlling [2]. Scrum Master encourages teamwork and builds healthy relations between team members. A major task of this person is to analyze and monitor the progress of work as well as the team [4].

Refer to Fig. 1 that depicts the scrum roles and their responsibilities.

Every individual role holds its own importance in the Scrum process. If any of them is not present during the Scrum flow, it becomes difficult to ensure a smooth working flow. This will eventually hamper the development process, slow down its progress, and may lead to inappropriate results.

Fig. 1 Scrum roles


Scrum Team
Product Owner
Manages the Product Backlog
Optimizes value of the Product
Managing itself
Creates "Done" Increments
Manages the Scrum process
Removes impediments
Scrum Master
Development Team

3.2 Sprint

A Sprint is an iteration in which a piece of work is undertaken in a time period of about two to four weeks. This time period is fixed for all the iterations that are going to be executed throughout the project [2]. The user stories from the product backlog that needs to be executed, are divided into a number of sets. These sets are implemented in their respective allotted sprints [4]. Before the beginning of every sprint, there is a meeting which covers the identification of work to be done in the following sprint and estimated goal prediction. The sprint ends with a review meeting to inspect sprint work [4]. At the end, the result of sprints is combined to form a project [5]. To cancel or re-schedule the sprint, the product owner needs to be approached, as only he has the authority to do it [2].

3.3 Scrum Process

The scrum process is a combination of several stages performed by different roles.

Determining product backlog. The project developed using the SCRUM process begins with an illustration of the system that needs to be prepared. The product owner illustrates the plan into a product backlog [1].

A product backlog is a list of features called user stories. These features are nothing but the requirements of the system [3]. The product owner lists down all the features that need to be included in the system. Then he prioritizes this list of features, according to his immediate and later requirements. Now, this list of features is used to build a product backlog [1]. All the user stories and functionalities are registered into the product backlog. Which backlog items need to be developed in the followings sprint are filtered by the product owner on the basis of priority and feasibility [2]. Then the product owner informs team members about the features to be included in the system. All the user stories in the product backlog are executed by the team. After sorting the list of features based on end-user's perspective, the development of the system proceeds [4].

There is a term "sprint" used in scrum, which means the goals that we need to achieve in the next 30 days. It is an iteration of about 2–4 weeks. The authority to cancel a sprint is only given to the Product Owner [1].

Purpose. The stories that need to be developed are identified according to priority which makes the depiction easy for the team and leads to efficient development.

Sprint Planning. After determining the product backlog, a meeting is summoned at beginning of each sprint, called "Sprint Planning meeting". Product owner and all the team members attend this meeting [1]. Following are the activities that are undertaken in this meaning:

1. The goals and agenda of the sprint are identified.
2. The importance and purpose of each user story is discussed.
3. An approximate timeline to acquire the goals is made [4].

4. Team asks a number of questions about the user stories and functionalities to the product owner, so that they can make a sprint backlog. A sprint backlog is just a detailed version of a product backlog [2].

Also, each team member determines that how many hours he will spend to execute each user story. After deciding the working hours, the team can start carrying out the first story is the product backlog. Multiple user stories from the product backlog can be worked on together by the respective team members [1]. The first sprint is generally of 2 weeks because the users-stories are still just in a documentation format. Then a consistent time period of sprints set the entire project (say 30-days/sprint).

This meeting can last up to 4 h and it is dedicated to making plans for what will be done in the following sprint [5].

Purpose. This kind of open sprint planning meeting provided a summarized view to the team and clarifies the roles and responsibilities of every individual [4].

Daily Stand-up meeting. To monitor performance, a meeting should be held daily to report what has each team member done, the previous day [1]. A short meeting of about 15 min is daily held, usually at the start of the day [3]. This meeting is called "Daily stand-up meeting".

This meeting is chaired by Scrum Master, who is responsible for monitoring performance and solving the problems that the team members encounter. All the team members need not the present in this meeting. The team member, that are executing the specific features, needs to attend the meeting [2].

All the team members discuss and coordinate their daily work. Each team member discusses the following 3 questions-

- What you did yesterday?
- What will you do today?
- What problem is he facing [5]?

The basic agenda of this meeting is to monitor the progress of the project against the sprint goals, exchange daily progress reports, solve the issues, and optimize time for scrum [3].

At the end of each meeting, the renewed time of completion, i.e., the remaining time to complete the project is calculated [4].

A *burn-down chart* is a graphical representation that shows the collective time remaining for the project. It shows how much more time is required to complete the project. It basically shows the progress of the project, efforts of the team, and the remaining effort. The progress of the project is good if the average pending effort is less than estimated effort [1].

Purpose. Daily meetings lead to transparency, timely monitoring and shared understanding. There is a development of informal communication between team members that makes the work process easy and fast (Fig. 2).

Sprint Review. After completion of every sprint, a 4 h meeting is held, referred to as sprint review [5]. It does not involve any presentation of slides or documentations.

A demo of the running features of the software is presented by the whole team. These are the features that have been developed during the last sprint period. Product

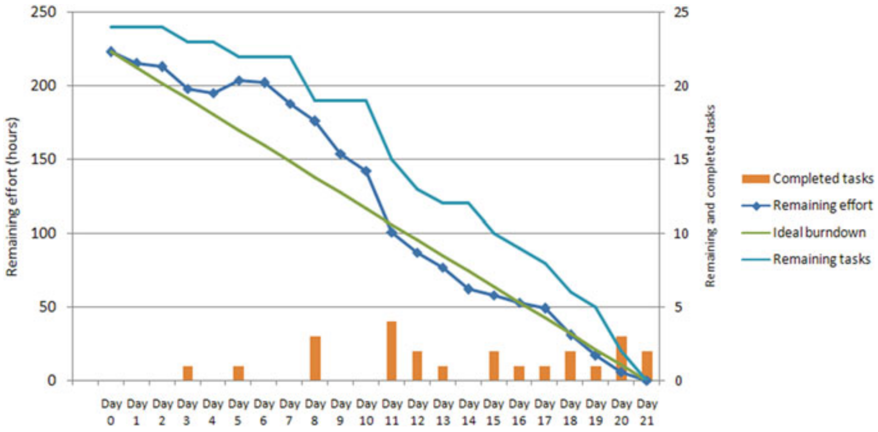


Fig. 2 A burn-down chart for monitoring

Owner attends this review meeting to see the state and progress of the project [4]. Sometimes, the customer wants to see that what the progress of the development of the software is. So, the product owner invites the customer to attend the sprint reviews meeting and present the online demo of features developed so far [5].

Purpose. The purpose of this meeting is to discuss the state of the project at this very point and review all the work done in the sprint. The plan for the next sprint is also discussed.

Sprint Retrospective. Sprint Retrospective is a time-boxed meeting, where following discussions take place-

- The task that is encountering issues [1].
- The tasks that can/cannot be continued for the next upcoming sprints.
- Ways to improve the developments process [3].
- Lesson for upcoming sprints [4].

Purpose. Such kind of meetings also monitors teamwork. What went good in the last sprint and what improvisation can be made in the next sprint is determined [3].

Scrum process is easy to carry if performed properly (Fig. 3).

4 Scrum Values

4.1 Courage

Scrum teams are well-introduced to the complexity of project development. They have the courage to take the best decision in difficult situations. The scrum master

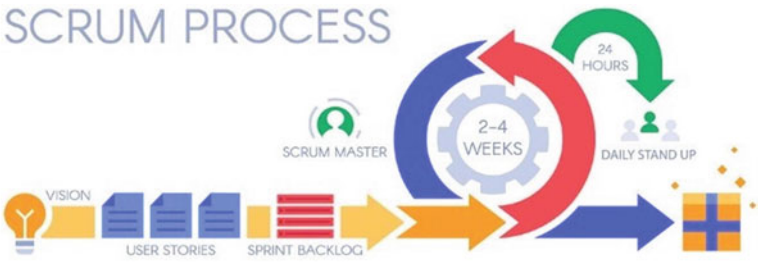


Fig. 3 Scrum flow

has to manage the entire team, encourage communication, and moral values between team members for good teamwork [5].

4.2 Focus

From product owner to all the team members including scrum master, all should be focused on working together and achieving the sprint goals. Scrum masters should focus on smooth scrum flow and solving the encountered issues [3].

4.3 Commitment

The entire scrum process needs commitment for work. Product owner should be committed to choosing the correct priority for backlog. Team should execute that backlog and commit to do whatever it takes to acquire the goal. Scrum master should be committed towards solving problems and assisting the team throughout [1].

4.4 Respect

Since scrum framework encourages teamwork, it is necessary for team-mates to have respect for each other. Product owner is in charge of the entire project, but he is only responsible to check whether work is done or not. He is not questionable to the team, that is how work is being done. Scrum master should not control the team, rather he should facilitate it. Each team member should be treated with equal importance [5].

Table 2 Benefits of scrum framework

Case	Description
Flexibility	Scrum is the most flexible framework as it is responsive to change. Traditional SDLC models like waterfall does not support this
Quality	Scrum is designed in such a way that it develops high-quality and efficient software in a specific time
Client involvement	Constant feedback from end-user is taken and they are also invited to see demos after every sprint
Monitoring	Daily monitoring helps to keep an eye on the progress of the project and also measures individual productivity
Live application view	At the end of each sprint, the client can see a demo of running features that have been developed Since testing is done after each sprint, issues are encountered and solved early
Quick testing	Since testing is done after each sprint, issues are encountered and solved early. It is efficient in comparison with waterfall where testing is done after the whole project is implemented
Estimations	Work and time estimations are done early to avoid the missing deadlines
Quick delivery	Since a working software is delivered initially to the client, rather than a full-documented software, the delivery time is quite short
Utility	Scrum framework makes sure that the usage of time and money in an efficient manner, which is essential for business profit

4.5 Openness

Product owner is the head of the team, but he should be open to welcome ideas from team members as well. Team members should be transparent and open to each other and also to scrum master. They should openly discuss ideas and solutions among themselves [5]. A sense of open-casual interaction should be there between product owner, team and scrum master [2] (Table 2).

5 Limitations

Just the way every coin has two sides, there are some limitations of scrum framework as well. It is no exception.

Lack of commitment. If there is a lack of commitment in the product owner, team or scrum master, then it will hamper the scrum flow [1].

Scalability. It is difficult to apply this methodology to large scale projects; whose development is complex. Traditional SDLC models are preferred in these cases [7].

Team Management. Sometimes, it becomes difficult for the scrum master to manage the team together as there may be a lack of cooperation between them. Teamwork is highly necessary.

Challenging. Daily meeting between large teams is difficult to manage. Also, when a team member leaves a project in the middle, it is difficult to train a new one in his place [5].

Need Skill and Training. Proper knowledge about the values and principles of this methodology is required. Lack of enough knowledge will raise issues while following the Scrum flow between large teams.

6 Conclusion and Future Scope

This review paper concludes that integrating Scrum and Agile is a good approach towards software development. Taking constant feedback from the customers would help, the software development cycle adapts changes, therefore, overcoming the limitation of the traditional SDLC waterfall model. On the basis of the analysis made, the following conclusions can be pinned:

- Scrum offers a high degree of responsiveness
- Open work atmosphere for self-organizing teams is provided
- Encourages sharing of knowledge and informal communication
- The live demo of working software is displayed to the customer after each iteration.
- Real time speed and progress measurement.

In future, enhancement can be done in scrum framework with respect to business change. Also, scrum could be implemented on other domains as well.

References

1. Permana PAG (2015) Scrum method implementation in a software development management. IJACSA
2. Singhal A, Divya G (2014) Scrum: an agile method. IJETMAS
3. Ng GC (2018) A study of an agile methodology with scrum approach to filipino company-sponsored IT capstone program. Int J Comput Sci Res
4. Jeldi NP, Chavali VKM (2013) Software development using agile methodology using scrum framework. Int J Sci Res Publ
5. Hossain EM, Babar MA, Paik HY (2009) Using scrum in global software development: a systematic literature review. In: Fourth IEEE conference on global software engineering
6. Modi HS, Singh NK, Chauhan HP (2017) Comprehensive analysis of software development life cycle models. IRJET 4
7. Leau YB, Loo WK, Tham WY (2012) Software development life cycle agile versus traditional approaches. IPCSIT, vol 37. IACSIT Press, Singapore
8. Dingsoyr T et al (2010) Agile software development: current research and future directions. Springe Berlin
9. Sutherland J (2001) Agile can scale: inventing and reinventing scrum in five companies. Cutter IT J
10. Ambler S (2008) Agile practice and principles survey: Feb 2008. http://ambysoft.com/surveys/agileFeburary2008.html. Accessed 11 Apr 2009

11. Stober T, Hansmann U (2010) Agile software development: best practices for large software development projects. Springer, New York
12. Langer AM (2012) Guide to software development designing and managing the life cycle. Springer, London, New York
13. Elbert C (2005) SpringerLink (Online Service): best practices in software measurement how to use metrics to improve project and process performance. Springer, Berlin
14. Maher P (2009) Weaving agile software development techniques into a traditional computer science curriculum. In: Proceedings of 6th IEEE international conference on information technology: new generation
15. Misra SC, Kumar U, Kumar V, Grant G (2006) The Organizational changes required and the challenges involved in adopting agile methodologies in traditional software development organizations. IEEE, pp 25–28