

## **Agile Methodology (SCRUM) Approach for Web Application Testing Process to Reduce Time, Cost and Improve the Quality**

**V. Vamsi Krishna<sup>a</sup>, G. Gopinath<sup>b</sup>**

<sup>a</sup>Research Scholar, School of Computer Science, Engineering & Applications, Bharathidasan University, Tiruchirappalli, Tamil Nadu, India.

<sup>b</sup>Professor, School of Computer Science, Engineering & Applications, Bharathidasan University, Tiruchirappalli, Tamil Nadu, India.

**Article History:** Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

---

**Abstract:** According to the finding of agile turn of events, Testing is perceived not to be a different stage, however a basic piece of Software Development with Coding. The "Entire group" approach is being utilized by the agile group to "heating quality in" to the Software Product. The agile group containing the Testers, who loan their aptitude in evoking instances of wanted conduct from clients. They work together with the improvement group that guides Coding, to transform those into executable details. Testing and Coding are being performed gradually and iteratively until it offers sufficient benefit to delivery to create each element developing is made. Its quadrants give a supportive scientific classification which helps groups in distinguishing and arranging the testing need.

**Keywords:** Agile Methodology, Scrum, Software Testing, Software Development.

---

### **1. Introduction**

The techniques of programming advancement life cycle follow consecutive models or are iterative actually like cascade model. With basic testing, programming advancement is the most perplexing errand; consecutive models can't effectively adjust the progressions that happen during improvement. This drawback has been taken out with the Agile testing measure which depends on the iterative philosophies and conquers the detriments of successive models [1]. The principle distinction between these two strategies, Agile programming testing and customary testing philosophies are as per the following: In the majority of the Software ventures, the improvement group alludes to steady methodology as opposed to successive one. Gradual and fast cycle techniques are utilized in nimble turn of events. It leads gradually towards the steady technique yet additionally discharge producing on past usefulness. The intensive testing in this cycle guarantees all issues are tended to in the next emphases. More pressure is laid on individuals and connections rather than cycles and instruments. Analyzers, designers, and clients reliably facilitate with one another. This coordination depicts attention to prerequisites to the analyzer for the highlights being created during a particular emphasis and can seclude the inconsistency among framework and its necessities. Live Software is the ideal move off the point-by-point documentations. Agile approach chips away at the reality of highlight points for example vis-à-vis correspondence and coordinated efforts with groups working two by two. Depending on the reality of broad correspondence, with clients alongside colleagues, the task doesn't need a thorough necessity record [2]. Client coordinated effort is in more thought in contrast with contract arrangement. Client contribution is a significant part of agile interaction. The engineers promptly look for the idea on prerequisites from the client for a better explanation. Reacting to change is of more thought in contrast with broad arranging. Extraordinary programming lays accentuation on changing the arrangement to oblige any adjustments in suppositions rather than strongly following the first arrangement. Programming testing structures a significant piece of interaction that helps to improve the nature of the item. Organizations are contributing a lot of their interest in testing the product. All Software organizations are going through a gigantic measure of cash and their experience on testing activities [3]. The agile strategy which is one of the exceptionally rehearsed philosophies considers altogether the enterprises these days. Agile is a way to deal with build up excellent Software and satisfy all the client necessities with a self-coordinated group for conveying practical Software with a period that additionally meets the client's all prerequisites. The product created with agile procedures conveyed rapidly to the client and as it is an iterative interaction checks every one of the mistakes and imperfections with the difficulties in the prerequisites [4]. With the client advancement group rolls out certain improvements and buoy the Software to the client, he proposed the improvement and joined in the Software/program. Various books, studies, and papers relating to this territory have been distributed and we mean to allude to these to control us with the examination work. Agile testing is getting utilized often these days as it includes client coordinated efforts and brief week cycles. It makes the task speed quick because of every one of these highlights. It is the best technique as it eliminates the inconveniences of existing models. It is the best model for the project where prerequisites are changing and project extension isn't clear. The continuous inclusion of the client at each progression likewise builds certainty and fulfillment of the client in the finished result and diminishes odds of imperfection in the

future. As customer connection is associated with every single cycle so final result getting conveyed after each cycle is as indicated by the necessities. Spry testing likewise diminishes the cost of the venture as working item is getting conveyed after each cycle in increases so odds of getting imperfection will turn out to be extremely less in future. Likewise, this system expands QA and Build group certainty and correspondence. Along these lines, Agile at present is new and quite possibly the most becoming acclimated to strategy because of its benefits, less conveyance cost, and different highlights in Industry nowadays [5].

## **2. Existing System**

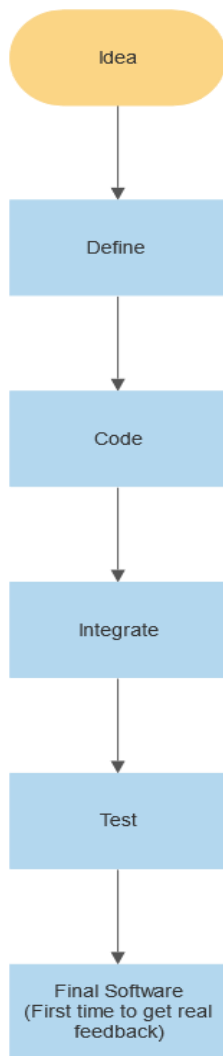
The drawbacks of waterfall development are that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage. No working software is produced until late during the life cycle. High amounts of risk and uncertainty [6].

Not a good model for complex and object-oriented projects. Poor model for long and ongoing projects. Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty are high with this process model. It is difficult to measure progress within stages. Cannot accommodate changing requirements. Adjusting scope during the life cycle can end a project. Integration is done as a "big bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early. The drawback with this SDLC model is that it applies only to large and bulky software development projects [7].

This is because it is hard to break a small software system into further small serviceable increments/modules. The disadvantages of the Iterative and Incremental SDLC Model are as follows—More resources may be required. Although the cost of change is lesser, it is not very suitable for changing requirements. More management attention is required. System architecture or design issues may arise because not all requirements are gathered at the beginning of the entire life cycle. Defining increments may require a definition of the complete system. Not suitable for smaller projects. Management complexity is more. At the end of the project may not be known which risk is. Highly skilled resources are required for risk analysis. Projects' progress is highly dependent upon the risk analysis phase [8].

The drawbacks of the Spiral SDLC Model are as follows –Management is more complex. The end of the project may not be known early. Not suitable for small or low-risk projects and could be expensive for small projects. The process is complex. The Spiral may go on indefinitely. A large number of intermediate stages require excessive documentation. The drawbacks of the V-Model method are as follows –High risk and uncertainty. Not a good model for complex and object-oriented projects. Poor model for long and ongoing projects. Not suitable for the projects where requirements are at a moderate to high risk of changing. Once an application is in the testing stage, it is difficult to go back and change functionality. No working software is produced until late during the life cycle. The Drawbacks of the Prototyping Model are as follows –Risk of insufficient requirement analysis owing to too much dependency on the prototype. Users may get confused about the prototypes and actual systems. Practically, this methodology may increase the complexity of the system as the scope of the system may expand beyond original plans [9].

Developers may try to reuse the existing prototypes to build the actual system, even when it is not technically feasible. The effort invested in building prototypes may be too much if it is not monitored properly. The drawbacks of the Big Bang Model are as follows –Very High risk and uncertainty. Not a good model for complex and object-oriented projects. Poor model for long and ongoing projects. Can turn out to be very expensive if requirements are misunderstood [10].

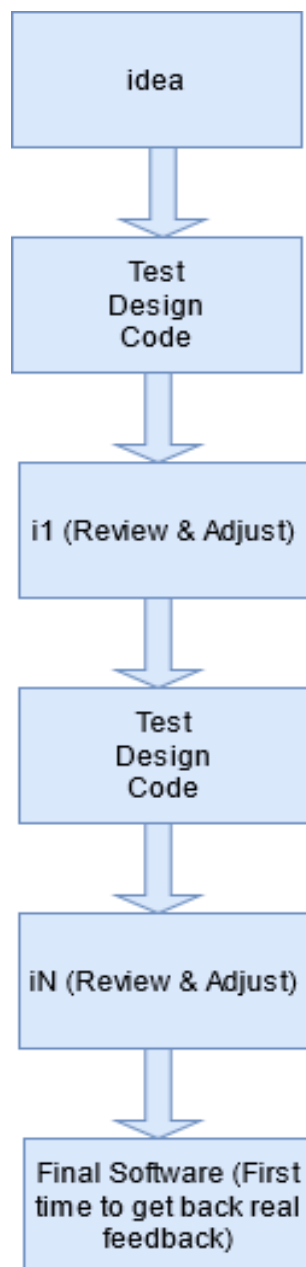


**Fig. 1.0.** Existing System Work Flow (Traditional)

### 3. Proposing System

Agile SDLC model is a combination of iterative and incremental process models with a focus on process adaptability and customer satisfaction by rapid delivery of working software products. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross-functional teams working simultaneously on various areas like –Planning, Requirements Analysis, Design, Coding, Unit Testing, and Acceptance Testing. At the end of the iteration, a working product is displayed to the customer and important stakeholders [11]. The agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided into time boxes (small time frames) to deliver specific features for a release. The iterative approach is taken and a working software build is delivered after each iteration [12]. Each build is incremental in terms of features; the final build holds all the features required by the customer. Here is a graphical illustration of the Agile Model The Agile thought process had started early in software development and started becoming popular with time due to its flexibility and adaptability. Following are the Agile Manifesto principles – Individuals and interactions – In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming. Working software – Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation. Customer collaboration – As the requirements cannot be gathered completely at the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements [13]. Responding to change – Agile Development is focused on quick responses to change and continuous development. Agile is based on adaptive software development methods, whereas the traditional SDLC models like the waterfall model are based on a predictive approach. Predictive teams in the traditional SDLC models usually work with detailed planning and

have a complete forecast of the exact tasks and features to be delivered in the next few months or during the product life cycle. Predictive methods entirely depend on the requirement analysis and planning done at the beginning of a cycle. Any changes to be incorporated go through strict change control management and prioritization. Agile uses an adaptive approach where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature-driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in the future. Customer Interaction is the backbone of this agile methodology, and open communication with minimum documentation are the typical features of the Agile development environment [14]. The agile teams work in close collaboration with each other and are most often located in the same geographical location. Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the agile model. The advantages of the Agile Model are as follows – Is a very realistic approach to software development. Promotes teamwork and cross-training. Functionality can be developed rapidly and demonstrated. Resource requirements are minimum. Suitable for fixed or changing requirements. Delivers early partial working solutions. A good model for environments that change steadily. Minimal rules, documentation easily employed. Enables concurrent development and delivery within an overall planned context. Little or no planning is required. Easy to manage. Gives flexibility to developers [15].



**Fig. 1.1.** Proposing System Work Flow (Agile)

#### 4. Agile Methodology (SCRUM)

Agile is a software development methodology to build software incrementally using short iterations of 1 to 4 weeks so that the development is aligned with the changing business needs [16]. This simple tutorial uses appropriate examples to help you understand agile development in a general and quick way. Agile is a software development methodology to build software incrementally using short iterations of 1 to 4 weeks so that the development process is aligned with the changing business needs [17]. Instead of a single-pass development of 6 to 18 months where all the requirements and risks are predicted upfront, Agile adopts a process of frequent feedback where a workable product is delivered after 1 to 4-week iteration. An Agile team works in iterations to deliver user stories where each iteration is of 10 to 15 days. Each user story is planned based on its backlog prioritization and size. The team uses its capacity – how many hours are available with a team to work on tasks – to decide how much scope they have to plan [18].

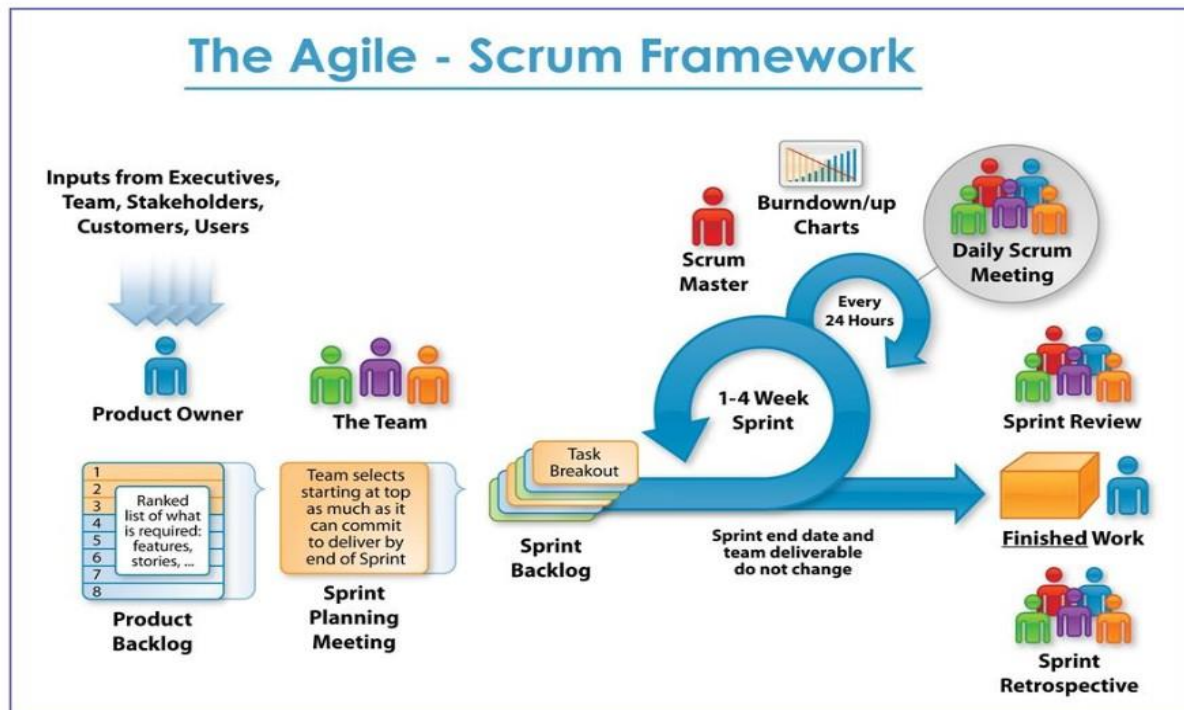


Fig. 1.2. Agile Scrum Framework Architecture

#### 5. Agile Scrum Components

Agile Scrum Model Process can be classified into three categories of components such as rocks, Ceremonies (Meetings) & Artifacts (Documents).

Table 1.0. Agile Scrum Components

Roles	Ceremonies (Meetings)	Artifacts (Documents)
Client	Kick of Meeting	PIN (Product Initiation Note)
SHS/CEO	Product Backlog Grooming Meeting	Product Backlog call user stories
PO	Sprint Planning Meeting	Sprint Backlog (Some user Stores)
SM	Daily Scrum Meeting	HLD & LLDS (high-level designs low-level designs. unit test cases Integration test cases.
ST	Sprint Review Meeting	Test Scenarios, Test cases, Test Data, Default Report
CCB	Sprint Retrospective Meeting Product Backlog Refinement Meeting	Automation Scripts, Sprint Burndown chart, Traceability matrix

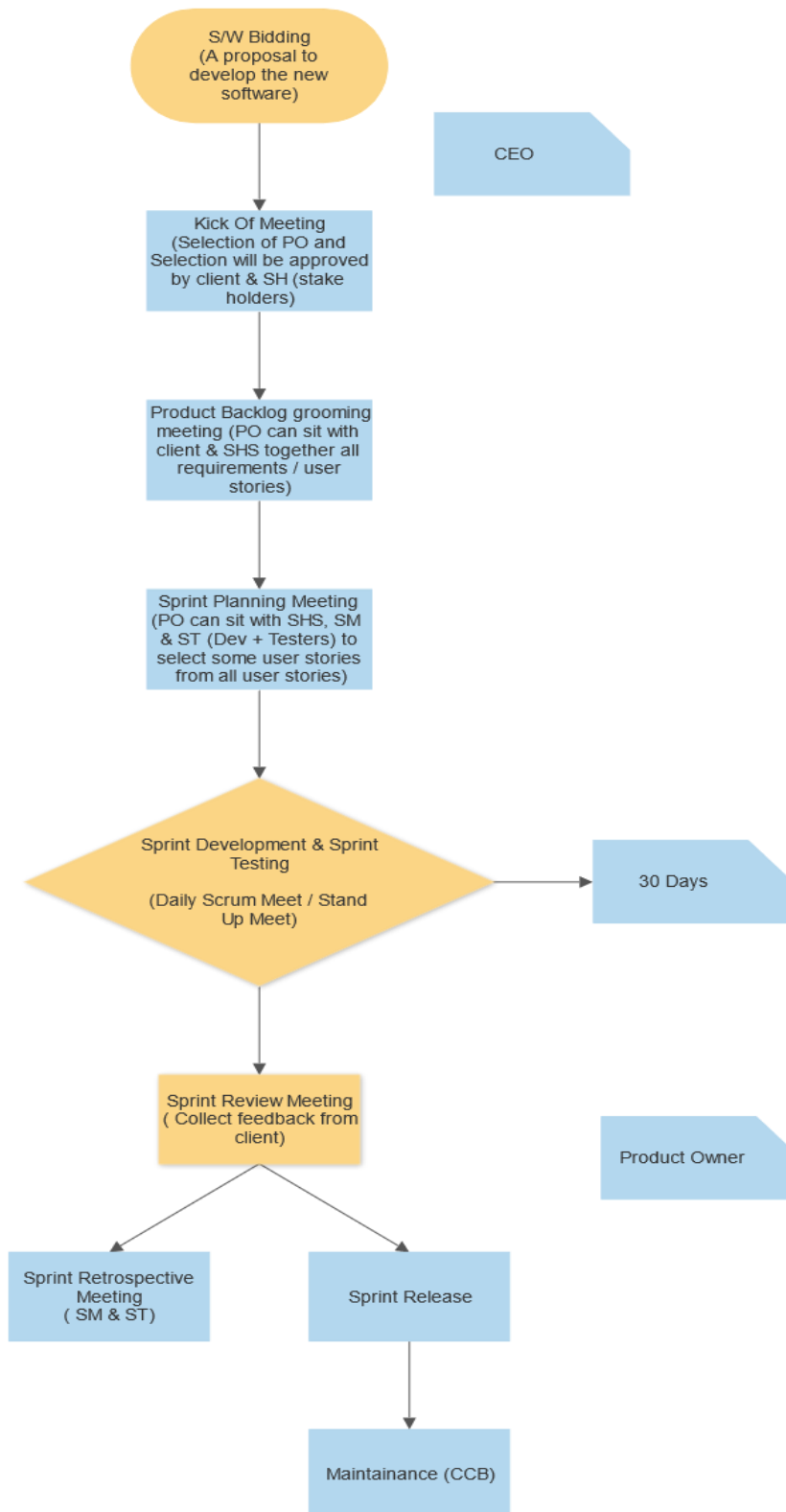


Fig. 1.3. Agile Scrum Components System Work Flow

## 6. Roles in Agile Scrum

### 6.1. Client Responsibilities in an Agile Scrum

- Responsible to approve the selection of Product Owner.
- Responsible to share all requirements with the Product Owner.

- Responsible to involve in every sprint review meeting to my feedback.
- Responsible to involve in product backlog refinement for changes in user stories. If meeting required [19].

## **6.2. Stakeholders Responsibilities in an Agile Scrum**

- Working under the guidance of the client.
- Involve in the kick of meeting to approve the selection of Product Owner.
- Help to Product Owner in product backlog grooming meeting for all user stories preparation.
- Involve Planning meeting along with Product Owner, Scrum Master & Scrum Team to Select Some user stories from all user stories.
- Involve in daily scrum meet along with Product Owner, Scrum Master & Scrum Team for status reporting.
- Participate in a sprint review meeting to collect feedback from a client on the corresponding sprint.
- Participate in product backlog refinement meetings for changes in user stories if required.

## **6.3. CEO (Higher Management) Responsibilities**

- Responsible to receive a proposal from the client in the project / Responsible to prepare own proposal for a product.
- Conduct kick of meeting to select product owner.
- Get approval on Product Owner selection from clients & stakeholders.

## **6.4. Product Owner Responsibilities**

- The product owner is responsible to gather all requirements by talking with the client of Stakeholders.
- Product Owner can take the help of SME ( Subject Matter Expert) while preparing user stories.
- PO an organize sprint planning meeting with scrum master, scrum team of stakeholders to select some user stories from all user stories depends on size, importance & complexity.
- Participate in daily scrum meet along with scrum master, scrum team, stakeholders for status reporting.
- Organize sprint review meetings to collect feedback from a client on sprint.
- Taking care of sprint release and guide Change Control Board for sprint maintenance.
- Organize product backlog refinement meeting along with stakeholders, clients, scrum master, and scrum team for a change in user stories if required.
- Working as the decision-maker and responsible for quality.

## **6.5. Scrum Master Responsibilities**

- Working as the facilitator.
- Responsibilities for scrum team attributes.
- Remove team-level issues.
- Responsibilities to motivate scrum team to finish work at right home.
- Protect team external issues.
- Responsible to provide facilities and sending the invitation to different roles for meetings, spent planning making, daily scrum meeting, sprint review meeting, sprint retrospective meeting & product backlog refinement meeting.

## **6.6. Scrum Team Developers Responsibilities**

- Working under the providence of scrum master and reporting to Product Owner and Stakeholders.
- Study user stories in sprint backlog to understand sprint.
- Prepare HLD and LLD for a sprint.
- Responsible for coding & unit testing.
- Integration and Integration testing.
- Sprint releaSe to testers for testing.
- Responsible for buys fixing reported by testers.

- Participate in the sprint planning meeting, daily scrum meeting, sprint review meeting, sprint retrospective meeting, product backlog refinement meeting.

#### **6.7. Scrum Team Testers Responsibilities**

- Working under the guidance of scrum master and reporting to Product Owner and SHS.
- Study user stories to understand sprint requirements and expectations.
- Prepare test scenarios, test cases, and test data including automation programs.
- Executive tests on sprint to detect defects.
- Report defect to developers.
- Participate in defect tracking.
- Receive modified sprint after fixing bugs to conduct retesting.
- Actively participate in sprint planning meetings and daily scrum meetings and sprint review meetings and sprint retrospective meetings and product backlog refinement meetings.

#### **6.8. CCB (Change Control Board)**

- CCB Responsibilities.
- Working under the guidance of PO.
- Responsible to handle change requests in released sprints.
- 

### **7. Ceremonies in Agile Scrum (Meeting)**

#### **7.1. Kick of Meeting**

- Conducted by CEO (hyper Management).
- To select PO.
- The time box is optional (meeting time).

#### **7.2. Product Backed Grooming Meeting**

- Conducted by PO along with clients and stakeholders (SMES are optional).
- Prepare product backlog with all requirements as user stories.
- The time box is optional.

#### **7.3. Sprint Planning Meeting**

- Conduct by PO along with SHS, SM, & ST.
- To select some user stories from all user stories in a product backed concerning user stories importance, size, and complexity.
- If any user story is big (epic) PO can divide that user story into multiple user stories [20].
- The time box is 3 hours.

#### **7.4. Daily Scrum Meeting**

- Conducted by scrum master along with PO, SHS, and ST.
- To know the work status of problems, but not to solve problems.
- The time box is 15 minutes.

#### **7.5. Sprint Review Meeting**

- Conducted by PO along with the client, stakeholders, scrum master, and scrum team ( developers & testers).
- To collect feedback from a client.
- The time box is 3 hours.

#### **7.6. Sprint Retrospective Meeting**

- Conduct by scrum master along with scrum team.



- To get scrum team feedback on sprint.
- The time box is 2 hours.

### 7.7. Product Backlog Refinement Meeting

- Conducted by PO along with the client, stakeholders, SM, ST.
- To Perform changes in user stones yet to be developed if required.
- The time box is 3 hours.

## 8. Artifacts in Agile Scrum

### 8.1 PIN

- Project or product initiation note.
- Prepared by PO.
- Specify overall plan informs of time & people.

### 8.2 Product Backlog (PBC)

- Prepared by PO like shown below.

User story id	Description	AS a	I Would like to	So that	Acceptance criteria
US – 1	Login	Login Registered user	Launch S/W Enter user-id Enter Password Click ok	The Next Page will be displayed	Need to run on windows/ Linux/ Mac computers and android is & windows mobiles.
			Requirements	Requirements	Expectations.

- Excel documents (Table wise).
- What to develop (requirement).
- Where to run (Expectations).

### 8.3 Sprint Backlog

- SBC is having some user stories which are selected from all user stories in the product backlog [21].

### 8.4 Sprint Burndown Chart

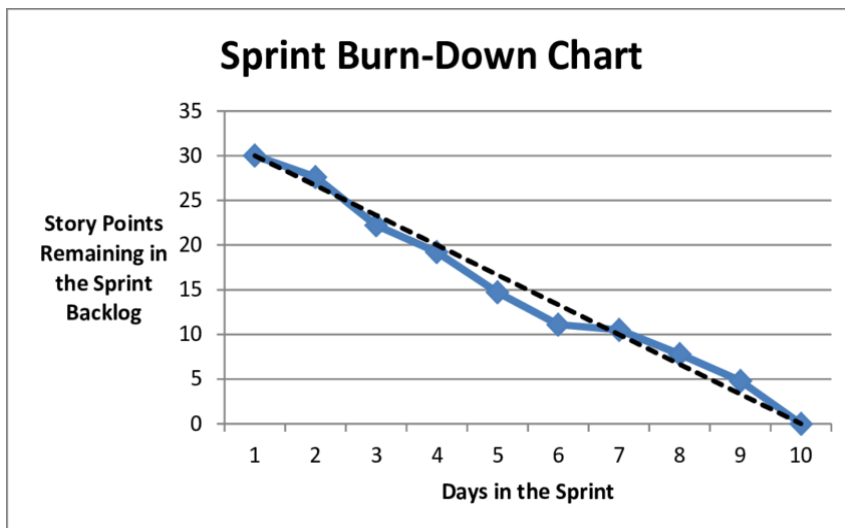


Fig. 1.4. Sprint Burn Down Chart

- The scrum master is responsible to prepare.
- To estimate the status of work. 10 members – Scrum team, 8 hours – Working hours in the day, 30 days – Total working days.

### 8.5 Traceability Matrix

- Preparing by the scrum master.
- To know the status at work.

Feature user stories	Test cases	Passed/failed	Defeat Report	Closed/Deferred	Comments
Login	User id test	Passed	-	-	-
	password test	Passed	-	-	-
	ok button test	Failed	DR	Closed	-

## 9. Agile Manifesto

### 9.1 Customer Satisfaction

The highest need is given to fulfill the prerequisites of clients through right on time and constant conveyance of important programming.

### 9.2 Welcome Change

Changes are unavoidable during programming improvement. Consistently changing necessities ought to be gladly received, even late in the improvement stage. Agile cycles should attempt to build clients' upper hand [22].

### 9.3 Deliver Working Software

Deliver functioning programming much of the time, going from half a month to a couple of months, considering a more limited time scale.

### 9.4 Collaboration

Business individuals and engineers should cooperate during the whole existence of a venture.

### 9.5 Motivation

Projects ought to be worked around roused people. Give a climate to help singular colleagues and trust them to cause them to feel dependable to take care of business.

### 9.6 Face-to-confront Conversation

Face-to-confront discussion is the most proficient and successful strategy for passing on data to and inside an advancement group [23].

### 9.7 Measure the Progress according to the Working Software

Working programming is the key and it ought to be the essential proportion of progress.

### 9.8 Maintain Constant Pace

Agile cycles point towards a practical turn of events. The business, the designers, and the clients ought to have the option to keep a consistent speed with the venture.

### **9.9 Monitoring**

Pay standard thoughtfulness regarding specialized greatness and a great plan to improve readiness.

### **9.10 Simplicity**

Keep things straightforward and utilize basic terms to quantify the work that isn't finished.

### **9.11 Self-coordinated Teams**

The agile group ought to act naturally coordinated and ought not to rely vigorously upon different groups because the best structures, necessities, and plans rise out of self-coordinated groups.

### **9.12 Review the Work Regularly**

Review the work done at standard spans so the group can think about how to turn out to be more compelling and change its conduct likewise.

## **10. Agenda for Release**

### **10.1 Opening Function**

Welcome message, audit reason and plan, putting together instruments and prologue to business supports.

### **10.2 Product Vision, Roadmap**

Show the huge image of the item.

### **10.3 Review Past Discharges**

Discussion on anything which can affect the arrangement [24].

### **10.4 Release name/topic**

Inspect the current status of guide subjects and do the necessary changes, assuming any.

### **10.5 Velocity**

Present the speed for the current delivery and of past discharges.

### **10.6 Release Plan**

Review key achievements and choice on time boxes for delivery and emphases inside discharge.

### **10.7 Issues and Concerns**

Check any worries or issues and record them [25].

### **10.8 Review and Update the Definition of Done**

Review the meaning of done and roll out suitable improvements dependent on innovation, expertise, or changes in colleagues since the last cycle/discharge.

### **10.9 Stories and Things to be thought of**

Present the client stories and highlights from the item build-up to be considered for booking in the current delivery.

#### **10.10 Determine Estimating Values**

If the speed is obscure, at that point plan the measuring esteems to be utilized in the delivery arranging [26].

#### **10.11 Coarse the Size of Stories**

The conveyance group decides the fitting size of the accounts viable and divides the tales into various cycles if a story is excessively huge. The item proprietor and the topic specialists explain the questions, elaborate the acknowledgment models and make appropriate story parts. The scrum ace works with the joint effort.

#### **10.12 Map Stories to Cycles**

The conveyance group and the item proprietor move the narratives/abandons in the emphases dependent on the size and speed. The scrum ace works with the coordinated effort.

#### **10.13 New Concerns or Issues**

Check any new issues dependent on experience and record something very similar.

#### **10.14 Dependencies and Suppositions**

Check any conditions/suspicious arranged during the delivery arranging.

#### **10.15 Commit**

The scrum ace requires the arranging. Conveyance group and Product proprietor signal it as the best arrangement and afterward resolve to move to a higher degree of preparation, that is, emphasis arranging.

#### **10.16 Communication and Coordination's Arranging**

Review/Update the correspondence and co-ordinations anticipating the delivery.

#### **10.17 Parking Parcel**

Process parking area implies all things ought to be either settled or set as things to do.

#### **10.18 Distribute Action Things and Activity Plans**

Distribute the things to do among their proprietors, measure the activity plan [27].

#### **10.19 Retrospect**

Solicit criticism from members to make the gathering effective.

#### **10.20 Close**

Celebrate the achievement.

### **11. Conclusion**

Agile models are chosen for the development of the software. Each has its advantages and disadvantages, it depends upon the organization which model to choose. If a customer changes its requirements frequently and is asked to deliver the Software in a short period with skilled resources then it is beneficial to choose Agile based testing model while in other hand Waterfall model can be used when a requirement is clear, projects are large and developers have sufficient time. If projects are very large, requirements change and proper validation is required in each phase then in that case tester uses "V-Model". But, we can say due to tight timelines and customer high expectations, now a day's Agile Testing Methodology is the best methodology which industry should adopt for delivering the projects.

## References

1. L. Rising and N. S. Janoff, The Scrum software development process for small teams, *IEEE Software*, Issue 17, pp. 26-32, 2000.
2. K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, Upper Saddle River, NJ, Prentice - Hall, 1st Edition, Oct 2001.
3. Julian, Brendan, James Noble, and Craig Anslow. "Agile Practices in Practice: Towards a Theory of Agile Adoption and Process Evolution." *International Conference on Agile Software Development*. Springer, Cham, 2019.
4. Puleio, Michael. "How not to do agile testing." *In Agile Conference*, 2006, pp. 7-pp. IEEE, 2006.
5. Abrahamsson, Pekka, et al. "Agile software development methods: Review and analysis." arXiv preprint arXiv:1709.08439 (2017).
6. Pawlak, Michał, and Aneta Poniszewska-Marańda. "Software Testing Management Process for Agile Approach Projects." *Data-Centric Business and Applications*. Springer, Cham, 2020. 63-84.
7. Padmini, K.V.J. & Kankanamge, P.S. & Bandara, Dilum & Perera, Indika. (2018). Challenges Faced by Agile Testers: A Case Study. 431-436. 10.1109/MERCon.2018.8421968.
8. Rajasekhar, P. & Yadav, Dr. (2013). Critical Issues in Software Testing During Agile Development.
9. Al-Zewairi, Malek, et al. "Agile software development methodologies: survey of surveys." *Journal of Computer and Communications* 5.05 (2017): 74.
10. Stolberg, Sean. "Enabling agile testing through continuous integration." *In Agile Conference*, 2009. AGILE'09., pp. 369-374. IEEE, 2009.
11. Rajput, Barkha. "Software Quality Assurance Using Agile Software Methodology in Education Assessment Industry." (2016). *Authorized licensed use limited to NUST School of Electrical Engineering and Computer Science (SEECs)*. Downloaded on February 12, 2021, at 07:10:50 UTC from IEEE Xplore. Restrictions apply. 2020 14th International Conference on Open Source Systems and Technologies (ICOSST) 978-1-7281-9050-1/20/\$31.00 ©2020 IEEE
12. Collins, Eliane, Arilo Dias-Neto, and Vicente F. de Lucena Jr. "Strategies for agile software testing automation: An industrial experience." *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops. IEEE*, 2012.
13. Stolberg, Sean. "Enabling agile testing through continuous integration." *In Agile Conference*, 2009. AGILE'09., pp. 369-374. IEEE, 2009.
14. Agile Marketing Documentation. [www.agileherpas.com/blog/state-of-agile-marketing-2020](http://www.agileherpas.com/blog/state-of-agile-marketing-2020).
15. Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). "Empirical study of agile software development methodologies: A comparative analysis". *ACM SIGSOFT Software Engineering Notes*, 40(1), 1-6. <https://doi.org/10.1145/2693208.2693233>
16. Mammalamadaka, K., & Krishna, V. R. (2013). "Agile software development and challenges". *International Journal of Emerging Technology and Advanced Engineering*, 3(6).
17. Rodríguez, P., Mäntylä, M., Oivo, M., Lwakatare, L. E., Seppänen, P., & Kuvaja, P. (2018). "Advances in Using Agile and Lean Processes for Software Development". *Advances in Computers* (Vol. 113, pp. 135-224). Elsevier.
18. Erickson, J., Lyytinen, K., & Siau, K. (2005). "Agile modeling, agile software development, and extreme programming: the state of research". *Journal of Database Management (JDM)*, 16(4), 88-100. <https://doi.org/10.4018/jdm.20051001>
19. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). "Agile software development methods: Review and analysis". arXiv preprint arXiv:1709.08439.
20. Schmidt, C. (2016). *Agile software development teams*. Springer International Publishing.
21. Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum* (Vol. 1). Upper Saddle River: Prentice Hall
22. Williams, L. (2010). "Agile software development methodologies and practices". *Advances in Computers* (Vol. 80, pp.1-44). Elsevier.
23. Chopade, M. R. M., & Dhavase, N. S. (2017). "Agile software development: Positive and negative user stories". *2nd International Conference for Convergence in Technology (I2CT)* (pp. 297-299). IEEE. <https://doi.org/10.1109/i2ct.2017.8226139>
24. Cohen, D., Lindvall, M., & Costa, P. (2003). *Agile software development*. DACS SOAR Report, 11, 2003.
25. Franková, P., Drahošová, M., & Balco, P. (2016). "Agile project management approach and its use in big data management". *Procedia Computer Science*, 83, 576-583. <https://doi.org/10.1016/j.procs.2016.04.27>
26. Rijwan Khan, Akhilesh Kumar Srivastava, Dilleshwar Pandey. *Proceedings of the SMART-2016, IEEE Conference ID: 39669 5th International Conference on System Modeling & Advancement in Research*

- Trends*, 25th\_27'h November 2016 College of Computing Sciences & Information Technology, Teerthanker Mahaveer University, Moradabad, India Agile Approach for Software Testing Process.
27. Samar Al-Saqa, Samer Sawalha, Hiba Abdel-Nabi. Agile Software Development: Methodologies and Trends Article. *International Journal of Interactive Mobile Technologies (iJIM)* July 2020.
28. Attique Ur Rehman, Ali Nawaz, Mohammad Tahir Ali. (2020) *14th International Conference on Open Source Systems and Technologies (ICOSST)* 978-1-7281-9050-1/20/\$31.00 ©2020 IEEE A Comparative Study of Agile Methods, Testing Challenges, Solutions & Tool Support.