

# Current Trends in Methodology for Software Development Process



Somya Goyal, Ayush Gupta, and Harshit Jha

**Abstract** Software development processes are heart core to finish a new project successfully. A wide range of software development methodologies is available from literature. This study analyzes the available methodologies for software methodology and finds out the most suitable methodology for a project. A survey-based research is carried out among developers and managers with varying experience in their field. For this study, agile and waterfall methodologies are focused as these are the popular methodologies to find our results. The results from our survey show that there is no “perfect” method to for a project as one needs to consider factors like resources, duration of project, etc., to select methodology to follow. It infers that waterfall-based process is popular among small projects in which requirements are well defined, whereas agile methodology is preferred with all kinds of projects as it provided better flexibility and continuous feedbacks, and delivery was provided to deliver a project with higher satisfaction rate.

**Keywords** Software development · Waterfall · Agile · Scrum · Kanban

## 1 Introduction

Software development process follows steps such as initiation, planning, execution and closing the project which are essential for development of a project and are discussed in the section below. There are various number of software development process models that have been created to make the project efficient [1, 2] such as waterfall, Six Sigma Lean, Spiral, V-Shape and agile. Waterfall model is oldest of all and one of the most followed models, it is performed in sequence of stages

---

S. Goyal (✉) · A. Gupta · H. Jha  
Manipal University Jaipur, Jaipur, Rajasthan 303007, India  
e-mail: [somyagoyal1988@gmail.com](mailto:somyagoyal1988@gmail.com)

A. Gupta  
e-mail: [ayush20118@gmail.com](mailto:ayush20118@gmail.com)

H. Jha  
e-mail: [harshitjha7777777@gmail.com](mailto:harshitjha7777777@gmail.com)

and only after completing one stage one can move to second. Six Sigma Lean is one of the newest models, in this both Lean and Sigma are merged to improve the quality of result and taking out the reasons to reduce variability in whole process. Spiral model emphasis risk analysis of project by repeatedly passing project through iteration. The V-Shape model executed in a sequence based on testing after each step. Agile software development model is a group of various methodology such as Scrum, Kanban, Extreme programming (XP), Lean, Crystal, who focused on satisfaction of stakeholders by rapid delivery after every stage which results in a better-quality product, and it is considered as one of the best methods and widely used in organizations presently.

The purpose of this study is to practically analyze the use of methodologies by developers using a survey. We focused more on agile and waterfall method as they are the most used methodologies used by product managers.

In agile model, we will focus on Kanban and Scrum for our study as these are one of the most used methodologies used by developers according to the survey. Kanban method gives visuals of the entire project to whole team and stakeholders which makes it more transparent and develops better coordination between team. Scrum framework developed by Ken Schwaber and Jeff Sutherland Inn 1993 [3], as explained [4] its implementation is done in a cross functional team by ensuring sprint goal is matched.

## 2 Related Work

In the field of project management, there are many methodologies to deal with software development life cycle (SDLC) after studying and evaluating many theories [5–7], with their advantages and disadvantages, we concluded to get a closer look at waterfall and agile. These methodologies and approach have been studied and are continuously improved like introduction of DevOps by Patrick Debois which helped the agile development by making it more efficient and providing continuous delivers to make improvement in every step of project [16, 24]. This also helped with more flexibility in project and minimizing errors.

There are many other studies on various models of SDLC, as we know, both waterfall and agile [8] are most used SDLC methodology and as agile replacing waterfall in industry is the reason why we selected those as our survey.

All models have their weakness and strengths [9, 10] but depending on time, resources [11], and nature of project, an organization work with the most suitable method to deliver an efficient project.

### 3 Proposed Approach

The proposed approach to carry out this study is given as Fig. 1. For this study, we took survey from a group of people including entry level developers and project managers, while finding which one is the most preferred methodology in software development life cycle. We will first compare waterfall with agile and then most preferred agile method between Scrum and Kanban.

#### 3.1 Comparison Between Agile and Waterfall

Waterfall is much organized than Agile as it is sequential process, and hence, the project can only go in next phase when previous one is completed, and testing is done after completion of project, whereas in agile, we continuously test the process and improves it if required [12]. On the other hand, agile is more flexible than waterfall [13] as stakeholders can provide their review after each phase and ask for a change if required. From all the study following points were noted:

- In waterfall model, it is difficult to make changes after completion of a process, whereas we can do it in agile.
- In waterfall, we need to define all the requirements and it should not be changed in between project.
- In agile, there is high chance of interaction with stakeholders.

#### 3.2 Comparison Between Kanban and Scrum

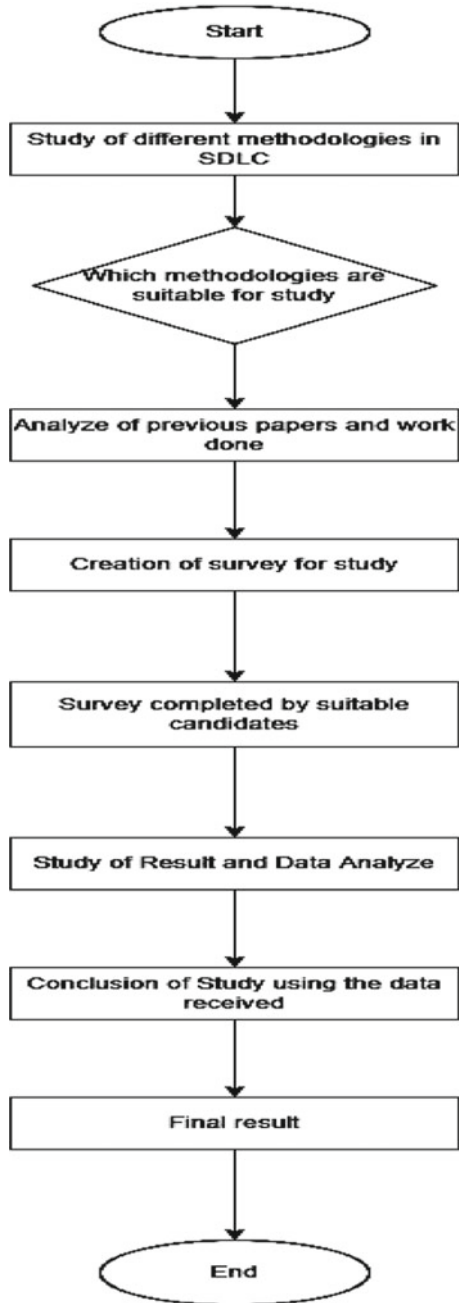
Scrum and Kanban both are agile methods but are different from each other. It is team's decision which methodology they want to follow for their project as Kanban gives more freedom than Scrum, whereas Scrum allows team to work in sprint and teams are not advised to change decision during sprint but in Kanban change at any stage is possible [14]. Here are some points which are useful in our study:

- In Kanban, a team can work without assigning a particular role.
- Meetings are necessary in sprints when Scrum is followed.
- Strict estimation of phase is required in Scrum.

As discussed above, these both are part of agile model, and hence, there are many similarities too, such as:

- Testing of both the processes is required to find which one is more compatible with team.
- Both processes are focused toward delivering project fast.
- In both, the methodologies feedbacks and reviews are taken frequently to deliver a successful product.

**Fig. 1** Flowchart of the steps taken to complete the study



- Scrum and Kanban both are transparent.

It should be noted, the final decision of selecting the methodology should be of team as per their work culture and atmosphere [15].

### 3.3 Study Preferences

To derive the results of this study, we created a survey and distributed it among the developers and managers. The survey helped us with the preference of methodology by a particular individual or team based on their requirements and resources.

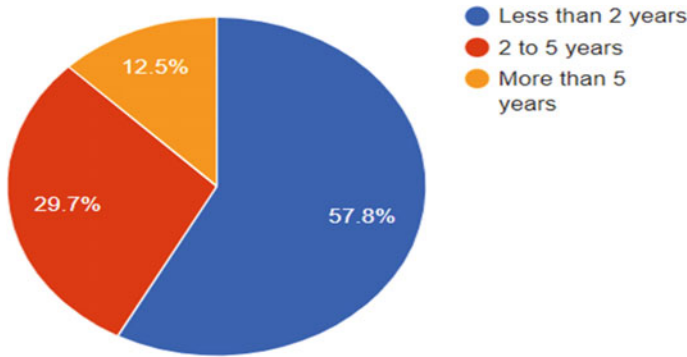
We created a section where we asked about the experience of individual as it will help us to determine which methodology is more preferred by an entry level developer and someone more experienced.

In next section, we asked people about the method they preferred and according to the method we asked them different question to get better results. Some important questions which are important for our study like:

- Time project took for completion as it helped with determining which project will be better for long term or short term.
- Satisfaction of the person who took part in research with the methodology their team preferred and what they felt about the project to get better picture of advantages and disadvantages of a particular methodology.
- Success of project was also an important factor for this study as it helped with the efficiency of use of resources in a particular methodology.
- Delivery of project was successful or not as it helped us with determining a particular methodology failed or passed.
- Size of team was also an important factor as it helped us to assess which method is suitable for big team and which works best for a small team.

## 4 Study Results

We circulated the survey among 64 candidates to get data for the required methodologies. We presented them with some question for waterfall, Scrum and Kanban. In this study, we focused on people with different experiences in their domain, and we categorized them into different categories like less than 2 years, 2–5 years and more than 5 years. We categorized them into these categories to get better idea of methodology preferred by people varying their experiences [16], and hence, we can get a better view at what methodology a person with less experience prefer [17] and what is preferred by someone with more experience. As we can see in Fig. 2, maximum number of people belonged to category with less than 2 years of experience, i.e., entry level developers, we also got a fair number of participants from people with more than 2 years of experience to generalize what method they prefer.



**Fig. 2** Experience of candidates

In our study, total 7 number of projects participated with a team of 3–20 people. Teams with members ranging from 3 to 4 preferred waterfall model, teams with 6–8 members used Kanban, while teams with 10–20 people used Scrum for their projects. As we can see in Fig. 3, 28.6% teams who preferred waterfall model were of small size, 28.6% teams who preferred Kanban were of medium size and teams who used scrum were large teams.

Most projects in our study were short-term projects as we can see in Fig. 4, out of 4 short-term projects, 2 followed Kanban, 1 followed waterfall and 1 followed Scrum. From 2 mid-term projects, 1 followed waterfall and the other followed Scrum and 1 long-term project preferred Scrum.

From Fig. 5, we derived 54 candidates prefer Scrum for their projects, 23.6% prefers Kanban and 27.3% prefer waterfall to complete their project. The preference of methodology depends on the size of team and duration of project [18, 19].

Users who preferred Scrum outlined that communication between team was good and nobody felt they are doing any work under burden, but they were not satisfied with the strict routine they followed.

**Fig. 3** Methodology used in projects

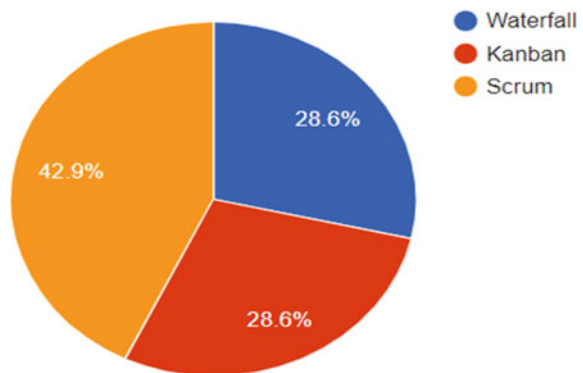


Fig. 4 Duration of project

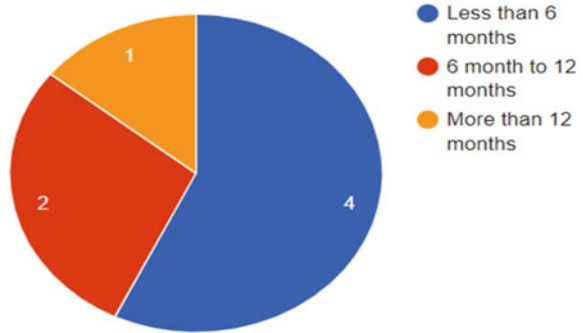
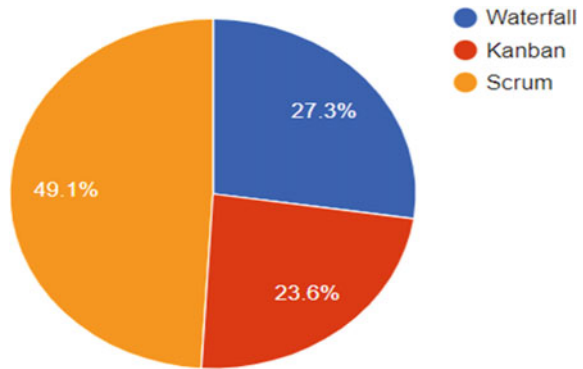


Fig. 5 Candidate satisfaction



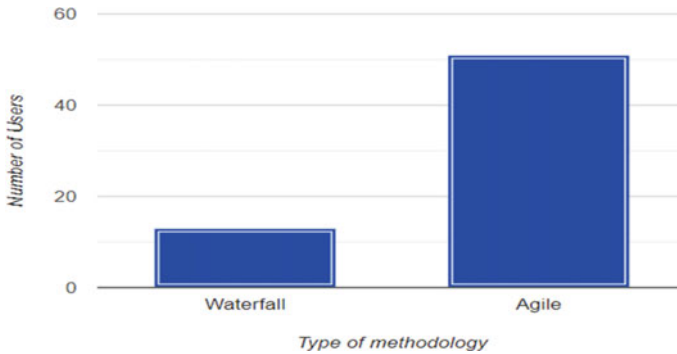
Waterfall users responded with satisfaction of the clarity and strictness toward the requirements which were given at start of project, but they faced problems with measurement of progress throughout the project.

Kanban users were happy with the flexibility the methodology offered them but frustrated with updating the cards again and again.

In our study, Scrum provided with the greatest number of successful projects with Scrum which was very close to Kanban. Candidates who participated in the survey preferred agile over waterfall because of the flexibility and synchronization it provided.

## 5 Conclusions

After the study, we can conclude that every method works different for every project depending on project's requirement, team size and resources. Every methodology has its pros and cons, and hence, everything about the project must be considered while selecting methodology. In Fig. 6, we can see 51 users in our survey preferred agile method and 13 users preferred waterfall method for their project.



**Fig. 6** Preferred methodology by participants

Hence, we found out agile was more popular among users and preferred for projects of all size as it provides more flexibility and synchronization between team, whereas waterfall worked well with short-term projects as and with a small size team having less than 10 members. Therefore, the choice depends on the need of project as there is no perfect method [20]. Waterfall was good with small team and small projects as the requirements were clear before starting of project, while agile worked well with all kinds of team and project as it offered flexibility and continuous review on the process.

## References

1. Alshamrani A, Bahattab A (2015) A comparison between three SDLC models Waterfall model, Spiral model, and Incremental/Iterative model. *Int J Comput Sci Issues* 12(1):106–111
2. Goyal S, Bhatia PK (2019) A non-linear technique for effective software effort estimation using multi-layer perceptrons. In: 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), Faridabad, India, pp 1–4. <https://doi.org/10.1109/COMITCon.2019.8862256>
3. Thummadi BV, Lyytinen K (2020) How much method-in-use matters? A case study of agile and waterfall software projects and their design routine variation. *J Assoc Inf Syst* 21(4). <https://doi.org/10.17705/1jais.00623>
4. Goyal S (2022) FOFS: firefly optimization for feature selection to predict fault-prone software modules. In: Nanda P, Verma VK, Srivastava S, Gupta RK, Mazumdar AP (eds) *Data engineering for smart systems. Lecture Notes in Networks and Systems*, vol 238. Springer, Singapore. [https://doi.org/10.1007/978-981-16-2641-8\\_46](https://doi.org/10.1007/978-981-16-2641-8_46)
5. Chowdhury AZM, Bhowmik A, Hasan H, Rahim MS (2018) Analysis of the veracities of industry used software development life cycle methodologies. [arXiv:1805.08631](https://arxiv.org/abs/1805.08631)
6. Matković P, Tumbas P (2010) A comparative overview of the evolution of software development models. *J Ind Eng Manage* 1:163–172
7. Petersen K, Wohlin C, Baca D (2009) The waterfall model in large-scale development. In: Bomarius F, Oivo M, Jaring P, Abrahamsson P (eds) *Product-focused software process improvement. PROFES 2009. Lecture Notes in Business Information Processing*, vol 32. Springer, Berlin, Heidelberg



8. Cocco L, Mannaro K, Concas G, Marchesi M (2011) Simulating kanban and scrum vs. waterfall with system dynamics. In: International conference on agile software development
9. Balaji S, Murugaiyan MS (2012) Waterfall vs v-model vs agile: a comparative study on SDLC. *Int J Inf Technol Bus Manage* 2(1)
10. Gandomani TJ (2016) Agile transition and adoption human-related challenges and issues: a grounded theory approach 62. <https://doi.org/10.1016/j.chb.2016.04.009>
11. Goyal S (2021) Handling class-imbalance with KNN (neighbourhood) under-sampling for software defect prediction. *Artif Intell Rev*. <https://doi.org/10.1007/s10462-021-10044-w>
12. McCormick M (2012) Waterfall vs agile methodology. MPCSC Inc. Revised edn. 8 Sept 2012
13. Petersen K (2010) Doctoral research in Sweden implementing lean and agile software development in industry
14. Mountain Goat Software (2016) Scrum overview: agile software development. Mountain Goat Software. Web. 22 Oct 2016
15. Forsberg K (1991) The relationship of systems engineering to the project cycle. In: First annual symposium of the national council on systems engineering (NCOSE)
16. Goyal S (2021) Effective software defect prediction using support vector machines (SVMs). *Int J Syst Assur Eng Manag*. <https://doi.org/10.1007/s13198-021-01326-1>
17. Goyal S (2022) 3PcGE: 3-parent child-based genetic evolution for software defect prediction. *Innovations Syst Softw Eng*. <https://doi.org/10.1007/s11334-021-00427-1>
18. Goyal S (2021) Predicting the defects using stacked ensemble learner with filtered dataset. *Autom Softw Eng* 28:14. <https://doi.org/10.1007/s10515-021-00285-y>
19. Goyal S (2022) Genetic evolution-based feature selection for software defect prediction using SVMs. *J Circ, Syst Comput* 31(11):2250161. <https://doi.org/10.1142/S0218126622501614>
20. Goyal S, Bhatia PK (2020) Feature selection technique for effective software effort estimation using multi-layer perceptrons. In: Proceedings of ICETIT 2019. Lecture Notes in Electrical Engineering, vol 605. Springer, Cham, pp 183–194. [https://doi.org/10.1007/978-3-030-30577-2\\_15](https://doi.org/10.1007/978-3-030-30577-2_15)