

# Hybrid agile development phases: the practice in software projects as performed by software engineering team

Norzariyah Yahya<sup>1</sup>, Siti Sarah Maidin<sup>2</sup>

<sup>1</sup>Department of Computer Science, Kulliyah of Information and Communication Technology, International Islamic University Malaysia (IIUM), Kuala Lumpur, Malaysia

<sup>2</sup>Faculty of Data Science and Information Technology (FDSIT), INTI International University, Nilai, Malaysia

## Article Info

### Article history:

Received Jun 24, 2022

Revised Nov 1, 2022

Accepted Nov 7, 2022

### Keywords:

Agile

Hybrid agile

Scrum

Software engineering team

Software project

Waterfall model

## ABSTRACT

The combination of scrum and waterfall is one of the software engineering teams that preferred hybrid agile models. The purpose of combining the two models is to leverage the advantages of each also to tailor the hybrid agile model to the needs of the project. However, to what extent are the phases, stages, and features of scrum and waterfall implemented in a software project remains unclear. Additionally, which phase will employ scrum, and when will waterfall be deemed optimal is also the arising question. This research adopted a qualitative study, and interviews are used as a data collection instrument. The interview is conducted based on an interview protocol, and thematic analysis is utilized to extract the themes from the interviews. This study investigates how the scrum and waterfall models are utilized in a software project, and three themes were identified in answering the research question. The findings indicate five development phases in a hybrid agile project and that waterfall is the preferable model in planning, while development is on scrum, and project testing and deployment could be either waterfall or scrum.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Norzariyah Yahya

Department of Computer Science, Kulliyah of Information and Communication Technology

International Islamic University Malaysia (IIUM)

Kuala Lumpur, Malaysia

Email: norzariyah@iium.edu.my

## 1. INTRODUCTION

Hybrid agile is a software project model that combines a plan-driven development model with agile approaches. The aim of the hybrid project approach is to bring together the best of the agile and traditional approaches in a software project [1], [2]. Many organizations have successfully applied hybrid agile to manage a large-scale project, make it easier to prepare proper documentation, and enhance the business analysis technique [3]. Combining a plan-driven development model with an agile approach will also increase team productivity via collaboration with the stakeholders to ensure that the development process is on the right track. In addition, Špundak [4] stated that hybrid models are being used in software development projects due to the need for different methodologies for their unique characteristics and advantages and disadvantages in one software project.

One of the hybrid agile models actively used by the software engineering team is the combination of scrum with the waterfall model. There are several names to represent the combination of both models, such as scrum and waterfall [5], scrumfall [6], water-scrum-fall [7], [8] and besides waterfall and scrum, there are many more hybrid agile such as hybrid V-model [9] and agile-stage-gate model [10], [11]. This research is a continuation of a previous study which determined that the software engineering team combines scrum and waterfall in software projects. The combination has resulted in a hybrid agile model. However, it is unknown

to what degree the scrum and waterfall models would be used in the software development phases. This study investigates which phases of software development projects are carried out utilizing scrum and which phases are carried out utilizing the waterfall model.

## 2. RELATED STUDY

This section discusses the related studies on the flow of the software development process using the Waterfall model and the scrum methodology. The strengths of both models will then be addressed. This section will also explore the existing hybrid agile model studied by [3], [5], [6], [12].

### 2.1. Scrum

Scrum is one of the agile methodologies widely adopted by practitioners, with 66% of team level in a software project using scrum, followed by a hybrid agile, the scrumb an at 9% [13]. Scrum has become one of the preferable development models due to its strength of frequent communication with the product owner over continuous iterations of the evolving software [14]. Iterations, incremental development, self-managed teams, and adaptability in the face of changing needs are all characteristics of scrum that are shared with agile approaches [3]. The phrase “iterative approach” describes breaking a project’s duration into iterations or sprints, where the entire project is separated into multiple smaller initiatives [15]. Every sprint follows the same format. Scrum has become one of the preferable development models due to its strength of frequent communication with the product owner over continuous iterations of the evolving software [15].

Figure 1 depicts the scrum framework, which includes product backlogs, sprint planning, sprint backlogs, and sprint retrospectives, all completed iteratively. A large scrum project is divided into manageable sprints; during each sprint, analysis, coding, and testing are materialize [16]. As illustrated in Figure 1, scrum starts with the product backlog. Backlogs are a list of tasks allocated to and expected to be completed by a scrum team in a specific time frame, and the product backlogs will be reorganized and prioritized according to specified criteria, such as priority in a sprint backlog. The sprint backlog consists of a list of tasks that need to be completed by a scrum team in a sprint. Each team will have to update their daily progress in a standup meeting known as daily scrum. The daily scrum has been assisting scrum team members to ensure that they meet the goal for a sprint and to ensure the project stays on track. In addition, scrum meetings can help team members come up with a clear idea [16] on their assigned tasks, and each scrum member is expected to demonstrate each member’s progress during the scrum meeting [16]. The cross-functional development teams collaborate to complete the specified task and ensure a successful sprint completion, and as mentioned by Sachdeva [17], it is essential for the team to maintain quality and maximize performance over the long term, as well as to coordinate and assist one another in delivering work using various sets of skills also in scrum everyone in the team works together consistently; to achieve a common goal [18]. Then, a sprint review will be conducted at the end of each sprint to assess the project against the sprint goal determined during the sprint planning meeting. Lastly, is a sprint retrospective, a recurring meeting held at the end of a sprint to evaluate what went well and what may be improved for the following sprint cycle. The retrospective stage of an agile sprint is a critical component of the scrum methodology for creating, delivering and managing complex projects. Its goal is to identify the achievements and mistakes of the previous sprint and to connect the resulting experience to action suggestions for improvement [19].

Scrum requires communication with the product owner over continuous iterations of the evolving software [20], which is essential to ensure that the product is in line with the product owner’s goals and expectations. In addition, frequent communication with stakeholders assists the teams in identifying the arising issues and finding solutions to fix the problem on a timely basis for a continuous improvement process. Also, to identify what went well and what went wrong will assist the team in identifying the areas of improvement needed. Scrum aids in increasing a team’s productivity [20] and can be applied globally to any project size [21]. Scrum was designed to increase development speed, align individual and organizations’ gold, define a performance-driven culture, support shareholder value creation through effective performance communication at all levels, and improve individual development and quality of life [20]. In addition, scrum assists teams in completing products in a timely and consistent manner [16] and ensures that money and time are spent wisely, and customer engagement is committed to continual feedback [16]. In addition, customer engagement can be an asset to help project managers and leaders to develop suitable strategies to follow in their projects. Thus, scrum is suitable when project requirements continue to evolve, frequent feedback is needed, and the project team needs some degree of flexibility in designing their ways of delivering deliverables and exploring a new experience with a new project that the team has never done before. Regarding the working products over the long term, scrum helps to eliminate errors and saves time. Scrum builds on the advantages of extreme programming to make it more systematic to establish the direction of development.

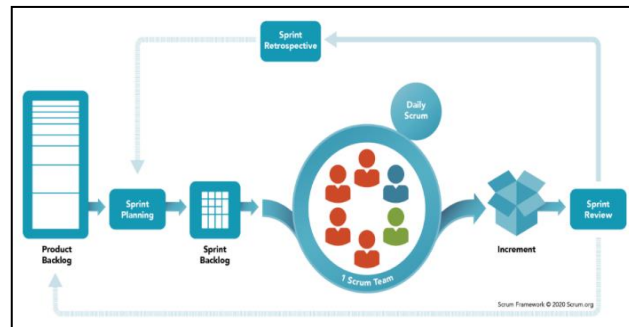


Figure 1. Scrum framework [14]

## 2.2. Waterfall

The waterfall model is one of the oldest software development lifecycles (SDLC). Winston W. Royce founded the Waterfall model in 1970, and the origin of waterfall consists of five phases: requirement, design, implementation, verification, and maintenance [22]. Waterfall has been known for its linear fashion phase development and is sometimes called the classic life cycle. Pressman and Maxim [23] stated that the waterfall model promotes a systematic and sequential approach for software development phases which begins with customer specification of requirements and progresses through planning, modelling, construction, and deployment.

Figure 2 illustrates the phases of the waterfall model. Communication is the first phase in a waterfall model. It is a phase where users' requirements are gathered, and the communication phase involves stakeholders in the requirement collection process. Following the completion of the communication phase is the planning. The planning phase establishes the project's timeline and milestones, as well as the cost estimating, scheduling, and project tracking strategy. After the planning phase is completed, the modelling phase begins. The modelling phase is where requirements are analyzed, and a software engineering team designs the system. The analyses and design will be based on the requirements gathered in the communication phase and the details obtained from the planning phase. Then, the construction takes place. The construction phase consists of code writing and testing. Finally, is the project deployment, which includes project execution, system support, and feedback. As can be seen from the flow, there is no room to revisit a phase that has been completed. As a result, no adjustments are possible, as the phase cannot be revisited [24]. This model is useful in structured systems development, where altering the software after coding is prohibited [24]. The waterfall makes software not reusable and the system not easily upgraded because the entire process will be modified for any adjustment, which is time-consuming and costly. [24]. Since the waterfall technique delivers the results at the end of a software project, the customer or developer might be in the midst of uncertainty regarding whether the current status of the project is as intended. While neither clients nor developers know when the project will be completed or received, there is a significant risk associated with the waterfall process, which typically requires too much time for damage control [25]. However, the waterfall model is easier to manage [6]. In addition, the deliverable and milestones are well-defined before the project starts [6], the project initiation and planning were adequately constructed, and the phases in the waterfall model and its activities are outlined. In addition, the waterfall model also works well on big and weak teams [6].

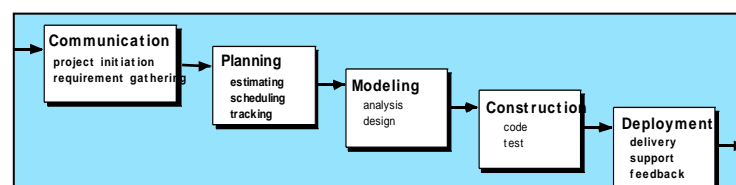


Figure 2. Waterfall model [23]

## 2.3. Scrumfall

The combination of the scrum with the waterfall model leads to a hybrid agile model known as scrumfall [6], water-scrum-fall [7], [8] and scrum and waterfall [5]. Rahim *et al.* [6] propose the scrumfall

model illustrated in Figure 3 which indicates the four software lifecycle phases: pre-game, high-level design, development, and post-game. Pre-game activities include communication, requirement elicitation, backlog creation, project planning, and cost estimation. In contrast, the post-game phase focuses on integration testing and finalizing the product for general distribution. The scrumfall model depicted in Figure 3 was created primarily to overcome the shortcomings of the scrum and waterfall models. These shortcomings are mainly because scrum model development activities are performed iteratively, with each iteration lasting one to four weeks and allowing for changes [14], [16], [17]. However, in the practical world, it is not possible to accommodate continuous changes in requirements in the later activities of a sprint. Besides, Rahim *et al.* [6] claimed that the iteration length for each sprint is too small to support large and complicated requirements. Thus, in overcoming the Scrum shortcoming, the scrumfall model is created by providing flexibility in the length of each sprint to support large and complex requirements. Rahim *et al.* [6] also reported that scrumfall holds success over large, critical systems, geographically distributed large teams where the team is combined by experienced and inexperienced personnel. In addition, scrumfall has shown effectiveness in time, cost, and economic factors.

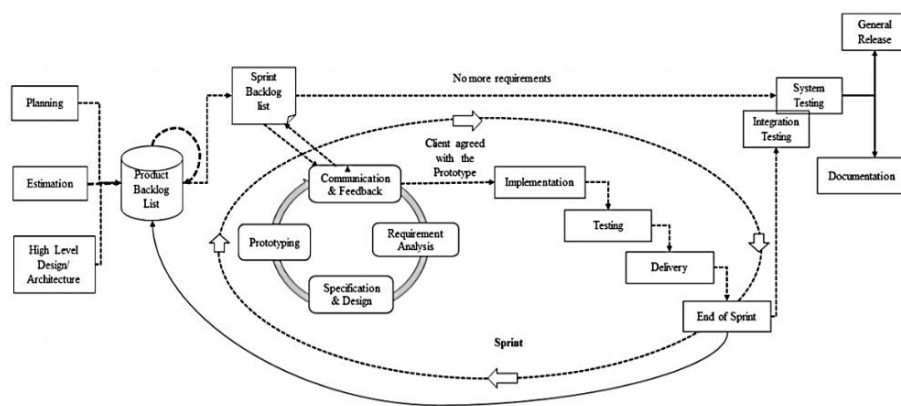


Figure 3. Scrumfall model [6]

Figure 4 illustrates the details for blending scrum and waterfall techniques in a case study conducted by Singhto and Phakdee [5]. The case study examines combining scrum and waterfall approaches to develop a set of Tailor-made software as a service (SaaS) to assist small medium enterprises (SMEs) in the east of Thailand in managing their business processes. The phases of development consist of five phases planning, analysis, design, development, and maintenance. For Singhto and Phakdee [5], the planning and analysis phases are carried out using the waterfall model. While the design and maintenance can be seen on scrum and waterfall, the development phase uses scrum.

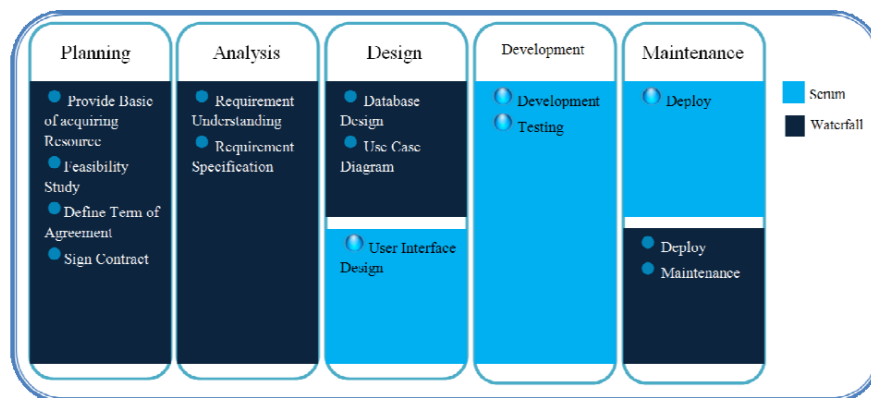


Figure 4. Blending scrum and waterfall model [5]

The model represented in Figure 5 is the model by Imani *et al.* [3]. Imani *et al.* [3] grouped the hybrid approach into two types: hybrid by phases and hybrid by agile methods. The hybrid by phases uses both traditional plan-driven and agile variants depending on the project phase. The hybrid by agile methods utilize mixed agile methods, such as scrum or XP, or using a plan-driven estimation tool in agile development. Both types have five process model phases, include the requirement, design, development, test, and project release.

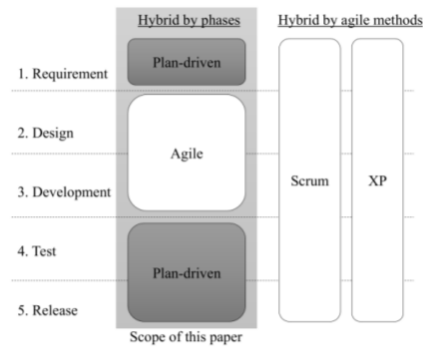


Figure 5. Hybrid approaches [3]

Imani *et al.* [3] conducted a quantitative and qualitative study to prove that hybrid approaches work better than the traditional plan-driven or agile methods. The study has quantitatively demonstrated that: the hybrid approach can be scalable for projects with high levels of requirement uncertainties, and the hybrid approach can improve project success rates, specifically in terms of cost. In comparison, the qualitative study adopted a case study as the data collection instrument on two different business organizations that used hybrid agile indicating the outcome that supports the two hypotheses in the quantitative research. The case study found that the hybrid agile can be used in a small project with high requirement uncertainty. Also, the project was successfully completed on time with a measured cost reduction by using the agile iterative development and the plan-driven test phase compared to the plan-driven approach. While, Wysocki and Orłowski [12] reported that scrumfall consists of three phases. The initial phase, the development phase and the final phase. The initial phase includes requirement analysis and planning is adapted the waterfall model. Scrum is going to be used during the development phase, which will include design, development, and implementation [12]. The integration and testing phases will be included in the final phase [12]. Figure 6 demonstrates the waterfall scrum waterfall approach diagram and the activities involved.

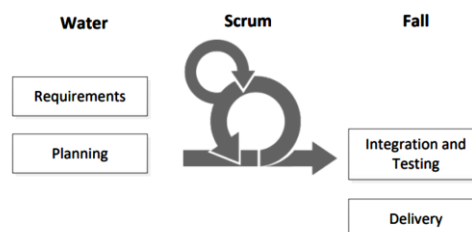


Figure 6. Waterfall scrum waterfall as software process [12]

#### 2.4. Why hybrid model

The goal of the hybrid model is to provide the greatest possible success for a software project [26], [27]. Given the advantages and pitfalls of various methods, a hybrid of the two has been offered as a viable alternative for overcoming one method's weakness by replacing it with the benefits of driven development models, which are employed as the process model in software projects. Due to the equal importance of value and timeline, a hybrid agile model using scrum and waterfall is advocated in this context. With the hybrid approach, the project's goal plan can be made and clarified incrementally, even if time, costs, and milestones

are planned for a long time [26], [28]. The focus is on the needs and benefits of the customer [29], [30]. Another advantage of the hybrid approach is that it allows people to think of solutions in more creative ways. On the contrary, changes in prioritization or new requirements can be incorporated flexibly without having to completely reschedule the project (hybrid itself). The hybrid model is appropriate for uncertain or risky projects. Brandl *et al.* [1] say the hybrid approach is best for complicated, business-critical innovation projects while Kosztyán and Szalkai [31] added that a hybrid approach makes a software development project more risk-tolerant. The hybrid approach is applicable for all projects, independent of firm size or complexity [15]. Moreover, it also benefit high-tech innovation projects [11]. Table 1 summarizes the strength and weaknesses of each approach.

Table 1. Comparison between waterfall and scrum

	Scrum	Waterfall
Strength	During the sprint review, additional features are coded and tested. Scrum can aid teams in completing the assigned deliverables in a timely and effective manner.	Suitable for a small-scale project, requirements that are well-known and unlikely to be changed in the future, fixed date of deployment, and it is ideal for a project with a significant number of interdependent tasks. It's simple to understand and manage and usually results in fewer production concerns.
Weakness	Teams may overlook software quality and accumulate a backlog of quality-related activities due to the rush of presenting the deliverables. Module integration testing cannot be monitored and managed all of the time since it takes a lot of time for development and testing.	The Waterfall approach isn't ideal for the level of complexity necessary in most present software development. It's inflexible and unsuitable for complex or long-term tasks.

Development teams of hybrid agile model can employ any approaches and methodologies best fit the needs of the problem they are solving with a flexible approach that embraces both traditional and Scrum development principles. In their day-to-day product development, many firms embrace scrum concepts and communication practices, but many still use traditional waterfall methodologies for planning, budgeting, and documenting project progress. Therefore, it is evident that a hybrid agile is a suitable approach in software development taking into consideration of the peculiarity for the strength and weaknesses of scrum and waterfall.

**3. RESEARCH METHOD**

This research adopted qualitative approaches, and the interview was selected as the data collection instrument. The interviews were conducted trailing an interview protocol discussed in [32]. This section elaborates on the research flows of this study. The research flow comprises four main phases illustrated in Figure 7. Phase one is the development of the interview protocol, phase 2 is the data collection by conducting a series of interviews, phase 3 is the data analysis, and phase 4 is the reporting.



Figure 7. The research flows

Phase 1: interview protocol. The interview protocol includes an overview of the research allowing the respondent to understand the whole picture of the study and consists of instructions on how to conduct the interview and get respondents' concerns on the data collection. The interview protocol is initiated by developing the question for the interview protocol. The questions are designed to inline with the research questions and research objectives. The questions are then composed as part of the interview protocol. The interview protocol is reviewed by a reviewer who has been involved in Software Engineering research with the aim of reviewing the questions and improving the interview protocol. Lastly, the interview protocol is pilot tested before it is used in the interviews.

Phase 2: series of interview. The second step is the data collection phase. The interview involved respondents who have been practising hybrid agile, and the respondents' experience is between three to more than ten years in software development. The interview involved five respondents who have been practising

hybrid agile and have experienced software development between three to more than ten years. The respondents are carefully selected to ensure they fulfil the stated criteria. Twenty-five emails were sent to software engineering practitioners who were our alumni who have worked in software development companies for three to ten years, inviting their participation in our research. Twelve practitioners responded to the email, and eight were chosen. Six of the eight practitioners agreed to participate; however, based on the data collected, one response has been excluded from the data analysis because the respondent has been using purely agile, extreme programming. And pure agile is not one of the respondent criteria that need to be met in our research. The questions as illustrated in Figure 8 were queried to the respondent in answering the research objective in this study.

Phase 3: data analysis. Each interview session was voice recorded, and a note was taken during each interview session. Once each interview session is completed, the interview is transcribed. The transcript was analyzed using thematic and content analyses.

Phase 4: reporting. The final phase, phase four, was the reporting. In this phase, the researcher examines the data carefully to uncover recurring themes. Each of the respondents' opinions is further elaborated based on the themes. The research report based on the identification of the themes is further elaborated and interpreted.

In a software project, a software engineering team will follow certain development phases like requirements gathering, analysis and design, coding, testing, and project deployment.

**Question 2(i):**  
 Could you please **explain the development phases** of \_\_\_\_\_ [mention the model that the respondent mentioned in *Question Number 1(i)*] model/framework/approach/method]?  
*Question Number 1(i) is the question in the interview protocol, which is not part of the research objective of this paper.*

**Question 2(ii):**  
**How** do you implement the \_\_\_\_\_ [mention the phases that the interviewee answered in *Question 2(i)* above] phase.

**Question 2(iii):**  
**When** do you implement the \_\_\_\_\_ [mention the phases that the interviewee answered in *Question 2(ii)* above, relate the phases].

**Question 3**  
 Which project development phase is crucial to execute?

i. **Which development phase is important?** Will it be on the design, coding, testing or some other phases?

ii. Do your **priorities change** when a **deadline** approaches?  
 [If the answer is **YES**, ask this question:]

1. Is there any reschedule being enforced?
2. How are the changes is being implemented?

[If the answer is **NO**, ask this question:]

1. **Why** are the priorities **not being changed**, although the deadline is approaching?

Figure 8. Interview question [32]

#### 4. RESULTS AND DISCUSSION

The findings discussed below are based on the interview sessions with five practitioners who have implemented hybrid agile in their software projects. The practitioners are those who have been involved in a software project with three to ten years of experience. The questions in the interview protocol were queried to each of the respondents. The questions from the interview, as shown in section 3 research method are divided into three themes as presented in Figure 9.

Theme 1: the development phases of an agile hybrid project implemented by the practitioners. Based on the respondents' feedback in answering question 2(i), all respondents stated that there are five phases in a hybrid agile project. The five phases are: i) project planning and requirements gathering, ii) project design, iii) development, iv) testing, and v) deployment and maintenance. Wherein each of the phases implements either waterfall or scrum approach. Theme 2 will discuss the detail of each five phases.

Theme 2: the models used in each development phase. Three respondents said that the first phase in a software project is planning. The respondents defined planning as the stage where the project manager is

responsible for gathering the projects' requirements, analysing the requirements, plan the project and designing the proposed system to meet the requirements. Then, the project manager will call the software engineering team to a meeting. For the meeting, two respondents, respondent A and respondent C called it a kick-off meeting, while others used the term meeting to represent the meeting before the development phases took place. The meeting is the platform where the project planning and the breakdown of tasks will be discussed and disseminated. The details of the requirements will be tabled, discussed, and delivered to the software engineering team to ensure that all software engineering teams understand the work delegation they were assigned with. Interestingly, all the respondents use waterfall in the early two phases, requirements gathering and planning and design.

All respondents in this research are using scrum in their development phase. In fact, the respondents highlighted that the core approach in development is agile scrum. Scrum has demanded the software engineering team meet on a regular basis to discuss their progress and share the incoming tasks that the team must complete. Similarly, in this research, all the respondents agreed that they had regular meetings when there were in the development stage, and based on the interview, three respondents will hold a progress meeting based on the necessity, while the remaining two will hold a daily standup meeting. Although a standup meeting is one of Scrum's distinctive qualities, some respondents are having a meeting based on necessity. As a result, the software engineering team has their own method to monitor their project, such as using a storyboard to update their progress, or some team has a system flow to trace and track their work and progress. However, the two respondents who having daily standup meetings stated that the daily scrum meetings are conducted for 5 to 10 minutes to update what the team has done and what they have been facing and what they will do for the current date.

While there is a mixture of approaches being used in the testing phase. Some respondents use the Waterfall, while others have been using scrum. As for testing on scrum, the development is based on modularization, wherein once specific modules or tasks are completed, the testing team will pick the tasks for testing. Software quality assurance (SQA) will adhere to an important role at this level. The quality assurance will test the feature of the completed tasks taken from the development phase, and if any bugs are found, the findings will be returned to the technical team to resolve. Quality assurance will also do the precaution test followed by the end-to-end test to ensure that the new feature does not negatively affect the existing system flow; to ensure the new feature will integrate well with the working functions. In scrum, the testing and development are done simultaneously and iteratively. While for the waterfall model, the testing will take place once the development is completed. Finally, the quality assurance team or testing team will sign off the sprint, which means everything is done then the feature will be deployed. Once again, the deployment and maintenance show all respondents are using the waterfall model. There is a team doing deployment, and the software engineers responsible for deployment will continuously monitor the system.

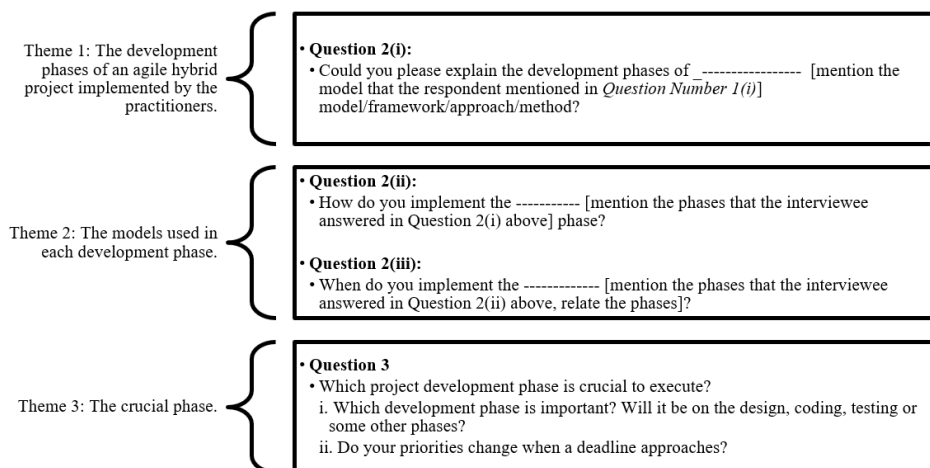


Figure 9. Themes identified to answer the questions

Theme 3: the crucial phase. This study identified from all the phases that four out of five respondents mentioned that the early stage of a software project which consists of requirements, project design, and planning are the important phase. One respondent (respondent C) stated that development is the crucial phase. Respondent A mentioned that priorities depend on two major factors. The factors are internal workload and



timeline. According to the respondent, a project is developed in modules, and later each module will be integrated. Each module will be given its priority, and the software engineering team will develop the module with the highest priority, followed by the lesser ones. In addition, if there are changes in the client's expectation or if any urgency arises, then the module's priority might also be affected. While respondent C stated that usually changes in priority happen in the middle phase of a software project. Most of the time, the changes are due to stakeholder requests because of a change of mind, or because part of the sprint needs to be fixed. It was mentioned by respondent C that fixing in a sprint is either done immediately in its active sprint or postponed depending on how severe the problem is, how it will influence the sprint, and how much money it will cost. The product manager is in charge of deciding whether to fix or postpone fixing based on the severity of the problem. On the other hand, respondents A and B came to the conclusion that the planning phase of a software project is the most important phase of a software project. Figure 10 summarizes the findings on the development methodology employed by the practitioner in a software project, which led to the hybrid agile model based on the theme identified in Figure 9.

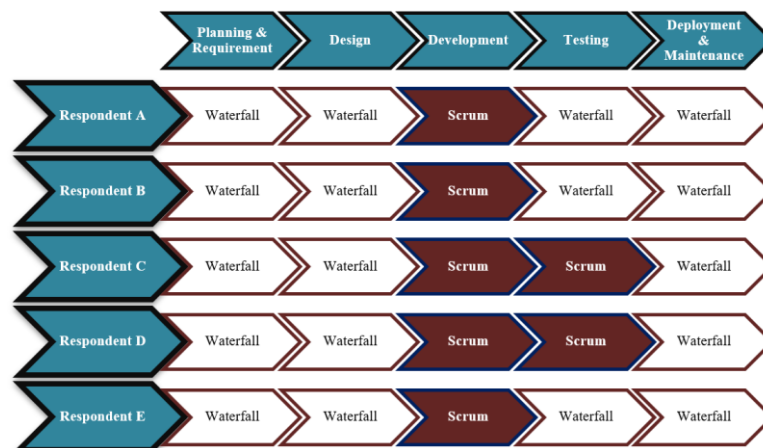


Figure 10. Software development lifecycle phases for a hybrid agile project

This research identified that software engineering teams preferred the waterfall model as their reference development model in the planning phase, gathering requirements and in the design stage. The waterfall and plan driven development as the model used in planning and requirements gathering is in line with findings by [3], [5], [6]. Project planning comprises five major activities such as estimation, scheduling, risk analyses, quality management planning and change management planning [23]. The waterfall model is seen as the ideal process model for the planning phase. Respondent A mentioned that research and planning are crucial phases in a software project. Similarly, according to respondent B, the early phase (refers to planning and design) is the crucial phase. Shylesh [33] also agrees that planning and requirement are the most vital phases in a software project. The planning phase is crucial since it is the phase where an analyst obtains data from clients, clarifies issues, and attempts to provide the best solutions [34] thus the software engineering team need to have complete control over the project at all times [35], especially in the planning phase. In addition, Singhto and Phakdee [5] stated that the Waterfall model emphasizes early-stage planning and identifies design flaws before the development phase, making the waterfall model the preferred process model in the planning stage. Unlike waterfall, in Scrum perspective, estimation is in the backlog, which caused the project's justification to be different between the benefits described in the business case and the initial estimates and plan [8].

In the design phase, respondents are using the waterfall model. However, for the design, there is a mix of a model being used by existing studies such as Rahim *et al.* [6] agreed design is waterfall model. In contrast, Singhto and Phakdee [5] use a hybrid of the waterfall and scrum models, whereas Imani *et al.* [3] reported scrum in the design phase. The mix between the waterfall model and scrum in the design phase may be due to the nature of the project itself or due to the software engineering expertise and specialities that the team has. In the design phase, the design must implement all of the explicit requirements of the analysis model [23]. It must accommodate all of the customer's implicit requirements, and the design should provide a complete picture of the software, addressing the data, functional, and behavioural domains from an implementation perspective [23]. Therefore, design needs a software engineering team with experience and resourcefulness in their domain [6], [36]. Unlike scrum, the weaker team is suitable for the waterfall model [6] because the

waterfall model is easier to understand, especially for non-developers or those new to software development [5] and inexperienced team [37]. Moreover, Alshamrani and Bahattab [37] also agree that waterfall is simpler to be implemented and, waterfall works well for software projects with an inexperienced team.

While for the development, both respondents stated that their software engineering team uses scrum by implementing modularization. Modularization is the process of dividing a software system into multiple independent modules where each module works independently. Scrum can be concluded as the ideal process model for all respondents in the development phase by [3], [5], [6]. Scrum itself mirrors the agile manifesto, which promotes a working software over comprehensive documents. Scrum strength is the iterative and the incremental approach to optimize predictability, shorter development cycle, higher customer satisfaction, lower bug rate, and quicker adaptation to changing business requirements [17]. In addition, Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed [36]. Scrum makes sure that the product is developed according to the stakeholders' needs faster and with better quality. It encourages business stakeholders and developers to work together to align the product with customer needs and company goals [5]. Thus, the practicality of SCRUM to come out with a working product gained the interest of the software engineering team.

In testing, the respondents from this research have two different preferable models, respondents A, B and E are in the waterfall model, and respondents C and D in Scrum. It is clearly stated by respondent C that the software engineering team uses scrum in testing because the testing team also refers to the kanban board. Therefore, updates on sprints, backlogs, and user stories, including the testing, are updated and referred to the kanban board. While for respondents A, B and E the testing will only take place once the development is almost completed. Lastly is the deployment and maintenance. All respondents adopted the waterfall in the deployment and maintenance phases which is similar to [3] with a plan-driven development model and [6] with the waterfall model, while [5] mix the scrum and waterfall model.

The software engineering team favours the combination of the scrum and waterfall model because the combination holds success over large, critical systems, and geographically distributed teams where the team is comprised of both experienced and inexperienced personnel [6]. Additionally, scrum and waterfall has demonstrated efficacy in terms of time, cost, and economic factors [6]. The hybrid agile with the combination of scrum and waterfall offers major advantages such as greater user and customer satisfaction with information technology (IT) services, financial savings, reduced time requirements, and improved alignment with company objectives [5]. When opposed to solely plan-driven methodologies, hybrid agile demonstrates the advantage in the context of increasing the possibility of improving the cost-benefit ratio [3]. Moreover, Kuhrmann *et al.* [38] highlighted that the selection of the models needs to be aligned with the project needs instead of individual preferences. The respondents agreed that hybrid agile is the preferred model in the development due to the needs of the project itself. Furthermore, Kuhrman *et al.* [38] shows that hybrid agile is motivated by necessity rather than mentality.

## 5. CONCLUSION

It is clear from this finding that no one single model is suitable for all projects; rather, both agile and non-agile approaches need to be adapted and combined in order to accomplish a wide range of objectives. This research shows that planning, requirement gathering, design, deployment and maintenance phases need a more classic approach like the waterfall model. While development requires more flexibility in iteration and promotes the involvement of the stakeholders in the development phases, such as scrum. On the other hand, testing shows different preferences among the respondents, wherein some respondents have been using the waterfall model while some is scrum. In a nutshell, it can be concluded that the preferences of model selection are regardless of the project size or personal preference by the software engineering team. Instead, it needs to be aligned with the project objective and needs. The evidence indicates that respondents agreed that the size of a project, regardless of its scale, either big, medium, or small scale project, had no weightage on the process model preferred for a software project. The emphasis is given on how the selected model would assist the software engineering team in achieving the project goal. Therefore, it can be concluded that hybrid agile is the one of the best solution for the software development team in achieving its project objectives, accelerate the development process and increase the team productivity.

## ACKNOWLEDGEMENTS

This work is supported by the Ministry of Education (MOE) through Fundamental Research Grant Scheme (FRGS/1/2019/ICT01/UIAM/03/2).

## REFERENCES

- [1] F. J. Brandl, M. Kagerer, and G. Reinhart, "A hybrid innovation management framework for manufacturing – enablers for more agility in plants," *Procedia CIRP*, vol. 72, pp. 1154–1159, 2018, doi: 10.1016/j.procir.2018.04.022.
- [2] J. Binder, L. I. Aillaud, and L. Schilli, "The project management cocktail model: an approach for balancing agile and ISO 21500," *Procedia - Social and Behavioral Sciences*, vol. 119, pp. 182–191, Mar. 2014, doi: 10.1016/j.sbspro.2014.03.022.
- [3] T. Imani, M. Nakano, and V. Anantatmula, "Does a hybrid approach of agile and plan-driven methods work better for IT system development projects?," *International Journal of Engineering Research and Applications*, vol. 07, no. 03, pp. 39–46, Mar. 2017, doi: 10.9790/9622-0703043946.
- [4] M. Špundak, "Mixed agile/traditional project management methodology – reality or illusion?," *Procedia - Social and Behavioral Sciences*, vol. 119, pp. 939–948, Mar. 2014, doi: 10.1016/j.sbspro.2014.03.105.
- [5] W. Singhto and N. Phakdee, "Adopting a combination of scrum and waterfall methodologies in developing tailor-made saas products for Thai service and manufacturing SMEs," in *2016 International Computer Science and Engineering Conference (ICSEC)*, Dec. 2016, pp. 1–6, doi: 10.1109/ICSEC.2016.7859882.
- [6] S. Rahim, A. E. Chowdhury, D. Nandi, and M. Rahman, "Scrumfall: a hybrid software process model," *International Journal of Information Technology and Computer Science*, vol. 10, no. 12, pp. 41–48, Dec. 2018, doi: 10.5815/ijitcs.2018.12.06.
- [7] G. Theocharis, M. Kuhrmann, J. Münch, and P. Diebold, "Is water-scrum-fall reality? on the use of agile and traditional development practices," in *Product-Focused Software Process Improvement. PROFES 2015. Lecture Notes in Computer Science()*, Springer, 2015, pp. 149–166.
- [8] D. West, M. Gilpin, T. Grant, and A. Anderson, "Water-scrum-fall is the reality of agile for most organizations today," 2011. [Online]. Available: <https://www.forrester.com/report/WaterScrumFall-Is-The-Reality-Of-Agile-For-Most-Organizations-Today/RES60109>.
- [9] T. Haya[1] T. Hayata and J. Han, "A hybrid model for IT project with Scrum," in *Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics*, Jul. 2011, pp. 285–290, doi: 10.1109/SOLI.2011.5986572.ta and J. Han, "A hybrid model for IT project with scrum," in *Proceedings of 2011 IEEE International Conference on Service Operations, Logistics and Informatics*, Jul. 2011, pp. 285–290, doi: 10.1109/SOLI.2011.5986572.
- [10] R. G. Cooper and A. F. Sommer, "Agile–stage–gate for manufacturers," *Research-Technology Management*, vol. 61, no. 2, pp. 17–26, Mar. 2018, doi: 10.1080/08956308.2018.1421380.
- [11] E. C. Conforto and D. C. Amaral, "Agile project management and stage-gate model—a hybrid framework for technology-based companies," *Journal of Engineering and Technology Management*, vol. 40, pp. 1–14, Apr. 2016, doi: 10.1016/j.jengtecman.2016.02.003.
- [12] W. Wysocki and C. Orłowski, "A multi-agent model for planning hybrid software processes," *Procedia Computer Science*, vol. 159, pp. 1688–1697, 2019, doi: 10.1016/j.procs.2019.09.339.
- [13] "15th state of agile report: agile adoption that accelerates across the enterprise," 2022. [Online]. Available: <https://info.digital.ai/rs/981-LQX-968/images/SOA15.pdf>.
- [14] "What is scrum? a better way to work together and get work done." <https://www.scrum.org/resources/what-is-scrum> (accessed Jun. 01, 2021).
- [15] M. Bogdanova, E. Parashkevova, and M. Stoyanova, "Agile project management in governmental organizations – methodological issues," *IJASOS- International E-journal of Advances in Social Sciences*, vol. 6, no. 16, pp. 262–275, Apr. 2020, doi: 10.18769/ijasos.616037.
- [16] V. Hema, S. Thota, S. Naresh Kumar, C. Padmaja, C. B. Rama Krishna, and K. Mahender, "Scrum: an effective software development agile tool," *IOP Conference Series: Materials Science and Engineering*, vol. 981, no. 2, p. 022060, Dec. 2020, doi: 10.1088/1757-899X/981/2/022060.
- [17] S. Sachdeva, "Scrum methodology," *International Journal of Engineering and Computer Science*, vol. 5, no. 6, pp. 16792–16800, 2016.
- [18] A. Khalid, S. A. Butt, T. Jamal, and S. Gochhait, "Agile scrum issues at large-scale distributed projects," in *Research Anthology on Agile Software, Software Development, and Testing*, IGI Global, 2022, pp. 388–398.
- [19] M. Wawryk and Y. Y. Ng, "Playing the sprint retrospective," Sep. 2019, pp. 871–874, doi: 10.15439/2019F284.
- [20] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, May 2017, pp. 864–869, doi: 10.1109/CCAA.2017.8229928.
- [21] D. Ma'arif, M. Yusnorizam, M. F. H. Yusof, and N. S. M. Satar, "The challenges of implementing agile scrum in information system's project," *Journal of Advance Research in Dynamical & Control Systems*, vol. 10, no. 9, pp. 2357–2363, 2018.
- [22] "Comparing traditional systems analysis and design with agile methodologies," 2009. <https://www.umsl.edu/~hugheyd/is6840/waterfall.html> (accessed Jun. 01, 2021).
- [23] R. S. Pressman and B. R. Maxim, "Process model," in *Software Engineering: A Practitioner's Approach*, New York: McGraw-Hill Education, 2019, pp. 19–47.
- [24] A. A. Adenowo and B. A. Adenowo, "Software engineering methodologies: a review of the waterfall model and object-oriented approach," *International Journal of Scientific & Engineering Research*, vol. 4, pp. 427–434, 2013.
- [25] R. Mokhtar and M. Khayyat, "A comparative case study of waterfall and agile management," *SAR Journal - Science and Research*, pp. 52–62, Mar. 2022, doi: 10.18421/SAR51-07.
- [26] K. Edwards, R. G. Cooper, T. Vedsmand, and G. Nardelli, "Evaluating the Agile-stage-gate hybrid model: experiences from three SME manufacturing firms," *International Journal of Innovation and Technology Management*, vol. 16, no. 08, Dec. 2019, doi: 10.1142/S0219877019500482.
- [27] M. R. Farokhad, J. R. Otegi-Olaso, L. S. Pinilla, N. T. Gandarias, and L. N. L. de Lacalle, "Assessing the success of R&D projects and innovation projects through project management life cycle," in *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Sep. 2019, pp. 1104–1110, doi: 10.1109/IDAACS.2019.8924298.
- [28] J. Reiff and D. Schlegel, "Hybrid project management – a systematic literature review," *International Journal of Information Systems and Project Management*, vol. 10, no. 2, pp. 45–63, Jul. 2022, doi: 10.12821/ijispm100203.
- [29] A. Gemino, B. Horner Reich, and P. M. Serrador, "Agile, traditional, and hybrid approaches to project success: is hybrid a poor second choice?," *Project Management Journal*, vol. 52, no. 2, pp. 161–175, Apr. 2021, doi: 10.1177/8756972820973082.
- [30] Z. T. Kosztyán, R. Jakab, G. Novák, and C. Hegedűs, "Survive IT! survival analysis of IT project planning approaches," *Operations Research Perspectives*, vol. 7, p. 100170, 2020, doi: 10.1016/j.orp.2020.100170.




- [31] Z. T. Kosztyán and I. Szalkai, "Multimode resource-constrained project scheduling in flexible projects," *Journal of Global Optimization*, vol. 76, no. 1, pp. 211–241, Jan. 2020, doi: 10.1007/s10898-019-00832-8.
- [32] N. Yahya, S. S. Maidin, and M. S. Safari, "The development of interview protocol to explore hybrid agile software development phases," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 3, pp. 1639–1645, Jun. 2021, doi: 10.30534/ijatcse/2021/221032021.
- [33] S. Shylesh, "A study of software development life cycle process models," in *National Conference on Reinventing Opportunities in Management, IT, and Social Sciences*, 2017, pp. 534–541.
- [34] M. Suhaib, "Conflicts identification among stakeholders in goal oriented requirements engineering process," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 4926–4930, Oct. 2019, doi: 10.35940/ijitee.L3557.1081219.
- [35] A. Mishra and D. Dubey, "A comparative study of different software development life cycle models in different scenarios," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 5, pp. 64–69, 2013.
- [36] "The scrum guide." <https://www.scrum.org/resources/scrum-guide> (accessed Jun. 01, 2021).
- [37] A. Alshamrani and A. Bahattab, "A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model," *International Journal of Computer Science Issues (IJCSI)*, vol. 12, no. 1, pp. 106–111, 2015.
- [38] M. Kuhrmann *et al.*, "Hybrid software development approaches in practice: a European perspective," *IEEE Software*, vol. 36, no. 4, pp. 20–31, Jul. 2019, doi: 10.1109/MS.2018.110161245.

## BIOGRAPHIES OF AUTHORS



**Norzariyah Yahya**    is an Assistant Professor at Kuliyyah of Information Communication and Technology, International Islamic University Malaysia (IIUM). Her research interest is Software Engineering in Agile development, software testing and quality. She is currently exploring Blockchain technology. She can be contacted at email: [norzariyah@iium.edu.my](mailto:norzariyah@iium.edu.my).



**Siti Sarah Maidin**    received the B.Sc. degree in computer science from the University Teknikal Malaysia Melaka, Malaysia, the M.Sc. degree in Information Technology from the University Teknologi MARA, Malaysia and the Ph.D. degree in Information and Communication Technology from Universiti Tenaga Nasional, Malaysia. She used to hold several administrative posts with the Centre for Foundation Studies, International Islamic University Malaysia (IIUM), from 2006 to 2017. She is currently a Senior Lecturer with the Department of Data Science and Information Technology, INTI International University, Malaysia holding post as Head of Postgraduate Studies. She has supervised numerous undergraduates and postgraduates' students. Her research interests include soft computing, data analytics, and system development. She can be contacted at email: [sitisarah.maidin@newinti.edu.my](mailto:sitisarah.maidin@newinti.edu.my).