

Agile, waterfall and iterative approach in information technology projects

C Fagarasan¹, O Popa¹, A Pislă¹ and C Cristea²

¹ Design engineering & Robotics Department, Machine Building Faculty, Technical University of Cluj-Napoca, 103-105 Muncii Avenue, Cluj-Napoca, Romania

² Electrical Machines and Drives Department, Faculty of Electrical Engineering, Technical University of Cluj-Napoca, 26-28 G. Baritiu Street, Cluj-Napoca, Romania

office.cristif@gmail.com

Abstract. The aim of this paper is to analyse the Agile methods implementation capability within software projects as a more viable alternative to the waterfall model and determine the key factors that guarantee the continuous delivery of software to market at the right and agreed quality level. Two Agile implementation methods were considered: Scrum and Kanban, which were evaluated based on factors like predictability, practicability and complexity. The conclusions show the selected methodologies advantages and disadvantages, and project key attributes can be used to choose the suitable methodology for a project. Regarding the Agile implementation methods, the most straightforward and predictable one was proven to be Scrum. This framework supports frequent delivery of software products that can address complex problems and adapt to the rapid market changes.

1. Introduction

The Information Technology (I.T.) industry is continually growing and expanding, forcing the people who work in this domain to enhance their workflow and improve the software delivery methods, even those considered to be well-known within the industry constantly adapting through learning.

In conformity with the report, edited by the United States Bureau of Labor Statistics [1], the software industry is one of the largest in the world economy in 2019, displaying the most significant increase anticipating that until 2022 will perform a 20% increase in the industry working places.

Considering the I.T. industry in Romania, there are about 17.000 active companies with over 100.000 registered employees and an annual income of over 4.8 billion Euros, as is presented in a study run by the Romanian Association for Electronic Industry and Software (ARIES) [2]. The analysis presented in the paper is interested in the development of I.T. processes and sustainability of the software companies, starting from now when this industry is in constant growth. It is essential to achieve a certain level of standardisation at the organisational level that enables to align the software products delivery process at the desired quality, and in the meanwhile to be able to measure the performance of each version that is accepted to be directed to the production teams. In this sense, various measuring instruments and methodologies like Life Cycle Management Index Tool (LY-MIT) [3] are developed.

It is accepted that complex software projects may take years to complete, so the tendency in the last twenty years was to migrate from a waterfall mindset, which means to deliver a working product at the end of the project, to an Agile mindset which implies to provide small increments of working software through time-boxed iterations. In [4], it is stated that the waterfall method is suitable for a predictable environment, whereas Agile works better in a change-driven climate, given the fact that the

software projects are continually evolving and the final shape of the product is practically far from clear in the beginning. Agile seems a rational option when talking about the implementation of software projects.

In [5], it is presented how the Lean methodology inspired the Agile methodology creation. It is argued that through this method, customer value is created by eliminating waste, amplifying learning, empowering the team, delivering fast and optimising the product as a whole. A known implementation method used in I.T. that derived from Lean is Kanban, a Japanese system that helps to visualise the workflow, maximise efficiency and continually improve. Another well-known methodology is the Agile implementation methods, detailed in [6], where the terminology and processes belonging to each one is described. The methodologies that organisations use and which are worldwide spread are Scrum, Kanban and Extreme Programming.

Starting from Peter Drucker's saying, "What gets measured, gets managed" [7], when a new software project is initiated, the project team, together with the project manager, choose between different methodologies available, deciding about the one that will be used for the project implementation. This happens during the discovery phase that is part of the project planning stage.

After the methodology is selected, the team might explore the various implementation methods frequently used to develop software projects and choose the best one to fulfil the project's needs [8]. Understanding the complexity of the future project and the environment in which the project will be situated will help the project manager select the potential implementation methods that can deliver the project's product. As mentioned before, this is done during the planning stage. However, there are a total of five stages through which each project will find itself in a moment in time: initiating, planning, executing, monitoring and controlling, and closing, according to [8].

These stages are usually included in traditional projects, and only when the objectives of a phase are achieved, the next one is started.

When an Agile methodology or iterative implementation drives the projects, most of the stages are repeated in each iteration. An advantage in applying this method is that the project's scope can be adapted during the implementation phase based on early feedback from customers or end-users. If the made assumptions during the requirements specification stage (identification stage) prove faulty, the adaptive implementation methods introduce flexibility into the delivery model, enabling the development teams to reprioritise and introduce changes that can increase the final product's value. Depending on the used framework, a different number of rules and principles will have to be applied, in correspondence and proportional with the project's agility level.

This paper aims to compare different methodologies used to deliver information technology projects. Moreover, based on [5], [9], and [10], Scrum and Kanban frameworks have been analysed with an emphasis on the differences between the frameworks and the key points that can help software development teams to select the best implementation method for their projects by leveraging the foreseen advantages and disadvantages.

2. Agile and Waterfall methodologies

Agile software implementation methods become three times more prevalent than Waterfall. In the last twenty years, statistics show that they have three times as many successful outcomes. That does not mean Waterfall is outdated, but particular projects may call for different solutions. Alternative methodologies need to be well-understood to make the right choice, so an appropriate decision is taken when starting a software project. The Waterfall model was named from the analogy to the finish-to-start relationship that exists between the project phases. Each stage involves different skills and has a formal acceptance and approval at the end. Figure 1 presents the Waterfall software development lifecycle.

When talking about Agile methodology, the product scope is divided into smaller shippable increments delivered at the end of each iteration. As presented in Figure 2, the phases are adapted at a smaller scale in the Agile methodology and are executed for each product increment. The increments are then split over iterations and then implemented. At the end of each iteration, will be delivered a functional product increment. Agile software teams must possess all the skills necessary to produce a working product at the end of each sprint, a term used in Scrum, and it refers to one iteration.

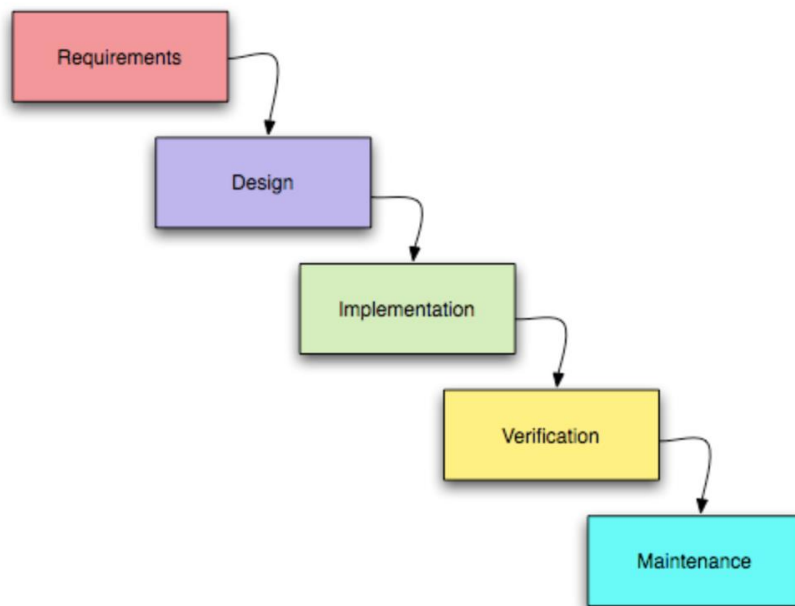


Figure 1. Waterfall software development lifecycle

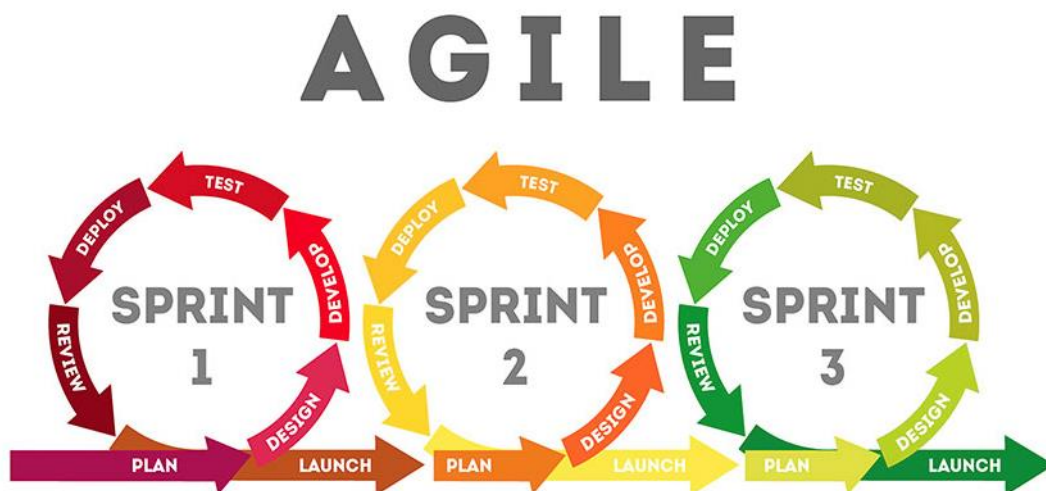


Figure 2. Iterative software development in the Agile methodology

In Table 1, some of the key metrics were taken as an example to explain the differences between these two software implementation methodologies. Figure 3 shows how using the iterative implementation reduce the overall project risk.

The metrics are related to the project management triangle: scope, time and cost. Other factors like time-to-market and return of investment (ROI), project risk, flexibility focus, and adaptability to change were considered. By following these metrics, the differences between Agile methodology and Waterfall methodology are more evident and can help make an informed decision when starting a new software development project. Choosing the suitable methodology for a project is crucial for its success.

The metrics presented can be used as a checklist in the project initiating phase. They can help the project managers understand the advantages and disadvantages of the analysed methodologies.

Table 1. Comparison between the Waterfall and Agile methodologies

Metric	Waterfall methodology	Agile methodology
Time spent on planning the project	Long. Scope and processes need to be defined before starting the actual product development	Short. There is limited time spent on planning before the first iteration is started
Time needed to get the implementation started	Long. The initiating and planning stages are formal and take a lot of time to complete	Short. Agile values customer collaboration over contract negotiation
Development direction	Plan-driven	Change-driven
Built-in flexibility to discover problems	None	Possibility to adapt during and after each iteration
Built-in flexibility to discover problems	None	Possibility to adapt during and after each iteration
Delivery time for a usable product	Long, at the end of the project	Short, at the end of each iteration
Time-to-market and Return of Investment	Fixed, at the end of the project	Adjustable, early increments might start bringing revenue
Project Implementation Cost	Higher due to fixed-price contracts, which can include premium risks	Time and Materials contracts usually offer more flexibility and can be value-dependent
Requirements	They are defined at the beginning of the project. These are unlikely to change during the implementation phase	Likely to change during the implementation phase based on end-user feedback
Scope changes	There is a lengthy process to follow if scope changes are introduced during execution	Scope changes are welcomed
Project delivery	One delivery, at the end of the project	Incremental deliveries, at the end of each iteration
Project Risk	High at the end of the project since this is the point when the product is delivered to the market.	Low at the end of the project because a shippable increment is delivered with each iteration.
Project Focus	Processes	End-user value
Collaboration between the customer and product development teams	Low	High, they can be part of the project team
Communication modes	With the Project Manager	With the product development team
Customer-vendor relationship	Contract negotiation before anything else	Customer collaboration before anything else

In the Waterfall methodology, the final product is delivered to the customer at the end of the project. The product development team does not receive any feedback during the implementation stage, and the focus is only on the requirements established in the planning stage. In contrast, in Agile methodology, the customer receives a working product increment after each iteration, and the customers' feedback can be implemented in the subsequent iterations. This is the main reason why most software organisations prefer the Agile methodology for software products development.

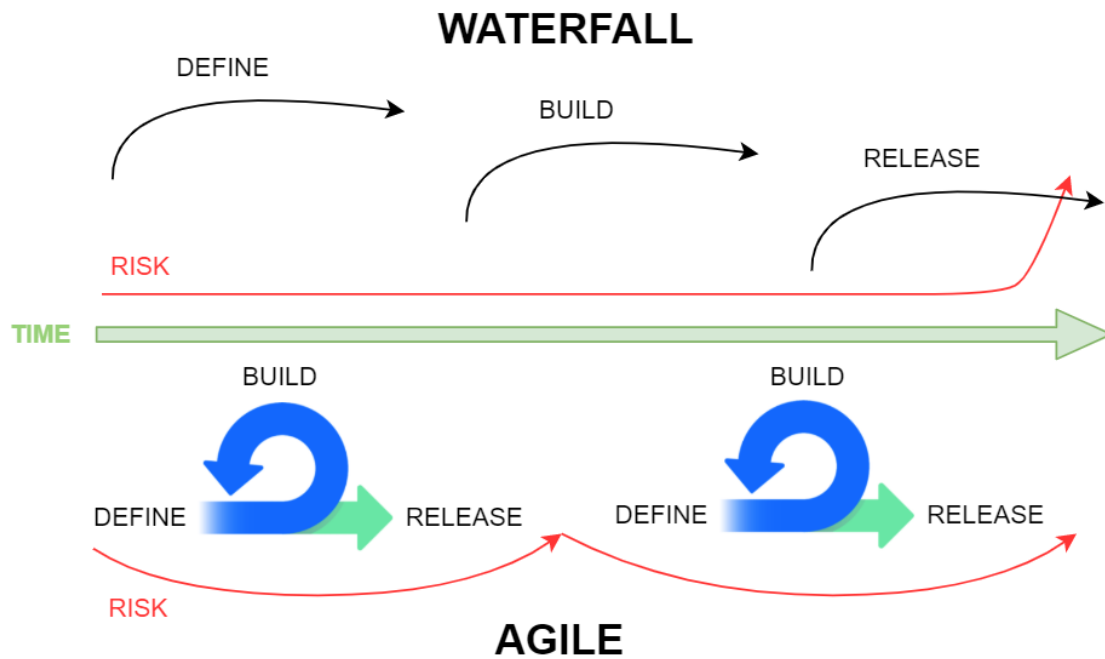


Figure 3. The project risk in Agile and Waterfall methodologies

As shown in Figure 3, by delivering software projects using the Waterfall methodology, the risk is high at the end of the project when the delivery of the project's outcome happens. During the project, the team focuses on implementing the requirements established during the planning phase. The risk is lower at the end of the project when using the Agile methodology because a shippable increment is delivered with each iteration. The customers' feedback can be implemented in subsequent iterations. However, some key project attributes can be evaluated to determine how they comply with the two methodologies presented in Table 2.

Table 2. Project attributes in the Waterfall and Agile methodologies

Project Attribute	Waterfall methodology	Agile methodology
Magnitude	Small or Medium-sized projects.	Any size projects.
Requirements	Clear from the beginning of the project, no room for change.	Not clear, they will become clearer after analysing the feedback received on the implemented increments.
Scope	Fixed scope, mentioned in the contract.	Likely to change during the implementation of the software product.
Budget	Fixed-budget mentioned in the contract. It has to be approved before the project is started.	The project can be started if a rough order of magnitude budget exists.
Delivery	The product has to be delivered in one piece.	The product can be delivered through iterations
Success	It is determined by the project plan implementation.	It is determined through customer feedback and end-user value.

3. Software projects development lifecycle – in Agile methodology

Figure 4 presents the popular Agile iterative development lifecycle that has been adapted for the software industry, given the flexibility needed during the software projects' implementation stage. Every organisation that is active within the I.T. industry can have specific processes and procedures. Generally, the same stages are present in the Agile methodology project lifecycle, as displayed in Figure 4. The project initiation phase, also known as the discovery phase, happens before starting the project. Usually, the team that will complete this stage is not the same as the one that will implement the project. Software development organisations have a separate department, called the "Presales department", which handles all business initiatives.

After the presales department receives a request for the proposal, a small team of experts is gathered for a short time to elicit the requirements, estimate the effort, and negotiate a contract. The period dedicated to this activity may vary for different reasons, but it is usually between two to four weeks.

After a contract is signed, a project manager is assigned to the project, and the planning stage is started. The first iteration is light in terms of scope because, during this iteration, the team will begin to set up the project, refine the requirements for the next iteration and build the backlog. Once these elements are completed, the development work can be started. In the meantime, the project manager will secure the necessary allocations, gather the project team and starts with the next iteration. The product development team should have all the skills required to deliver the first product increment.

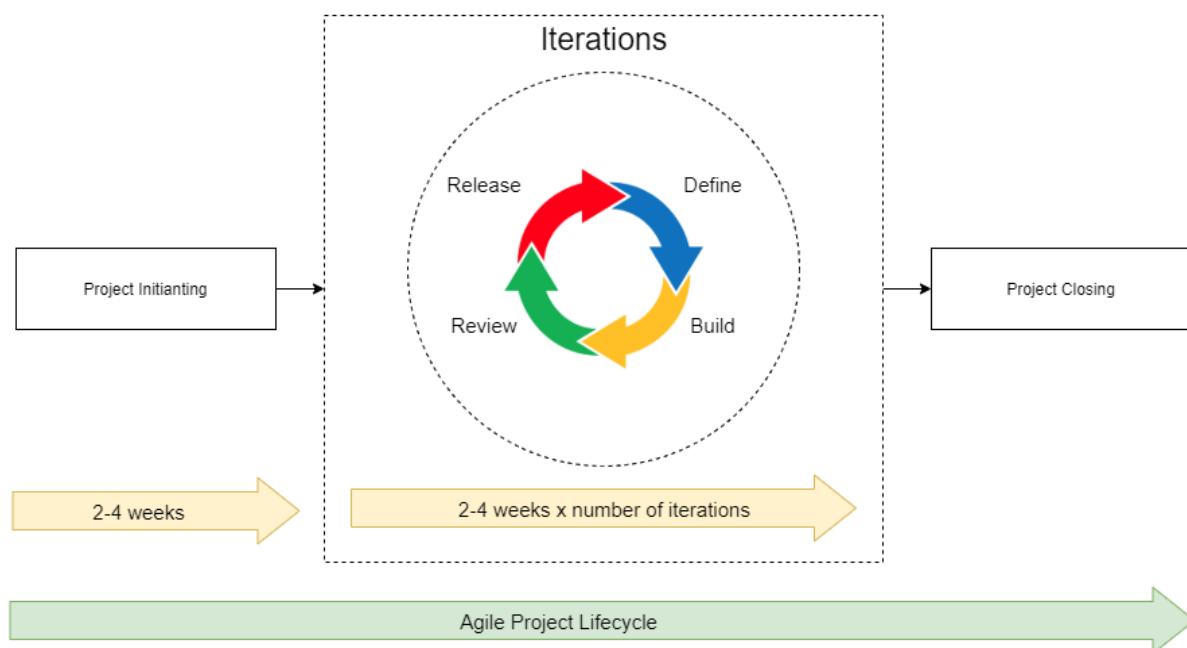


Figure 4. Agile software development lifecycle

The cycles Define, Build, Review, and Release will continue until all product increments are delivered and the scope of the project has been completed. The project will find itself in the Project Closing stage, where the attention is focused on the formal acceptance and the reallocation of team members to other projects. However, even if the review and retrospective sessions are organised within each iteration, it is recommended to schedule a "lessons' learned meeting" in which the product development team discusses the project challenges and how they were overcome.

4. Agile implementation methods: comparison between Scrum and Kanban

As it was mentioned, no implementation method will guarantee the project success. These are frameworks that can support the product development team, and their tools can be leveraged to perform

at the highest efficiency. Both of the considered methods have their roots in Agile methodology, and they focus on delivering frequently and continuous improvement [12]. Their best practices are similar; the total scope is divided into small manageable pieces and delivered. Incrementally, a synthetic comparison between them is depicted in Table 3.

Table 3. Comparison between Kanban and Scrum

Kanban	Scrum
There are no pre-defined project roles, and the project manager is optional.	All roles are pre-defined, and a Scrum Master role is required.
The project tasks are shared by the team as a whole.	The project tasks have dedicated owners as part of the product development team.
Timelines are not fixed; they evolve on a need basis, making it difficult to predict a product release.	Timelines are time-boxed into sprints.
Changes can be made at the customers' discretion.	Changes can only be made when a new sprint starts.
Cannot guarantee the meeting of deadlines due to the lack of predictability.	It has a more organised structure with defined procedures that can keep the product development on track.
Focused on changing expectations of the client, the team can adapt during the product iterations.	It is focused on customer-driven development and shipping value at the end of each iteration.
Project priorities might change daily.	Project priorities might change with every iteration.
It works better for large and distributed teams.	A scrum team should have a maximum of 9 members, and they should be collocated.
The project work can start immediately.	A Scrum master is needed to coach the team in order to make sure that the Scrum framework is understood and implemented.
Used for small pieces of work, defect fixes or enhancement requests.	It is used for feature development tasks with clear goals.
It is concentrated on work items.	It is concentrated on iterations.
The productivity is measured through cycle time.	The productivity is measured through team velocity.
The project team is specialised, and its team members are specialised human resources.	The project team is cross-functional and has all the skills to deliver a working increment.

It may seem that the Waterfall methodology can deliver the final product of the project faster, specifications of the project being detailed before the executing phase starts. In contrast, in Agile methodology, there is a tendency to go in a spiral, given the high number of changes that can appear. Despite that, the product starts bringing revenue earlier by implementing Agile methodology due to the significantly reduced time-to-market. Even if it has a minimum set of features, the product brings value to the customer in the beginnings. It is developed in parallel with end-user feedback, which can be incorporated in future product increments.

According to the statistics presented in the Annual State of Agile Report [13], developed by Digital.ai, 58% of the software projects that were implemented by using the Agile methodologies chose Scrum as an implementation method. In contrast, only 7% of the total software projects were implemented using the pure Kanban development method, as shown in Figure 6. Nevertheless, this

statistic should not influence a project managers' decision when choosing the suitable implementation method.

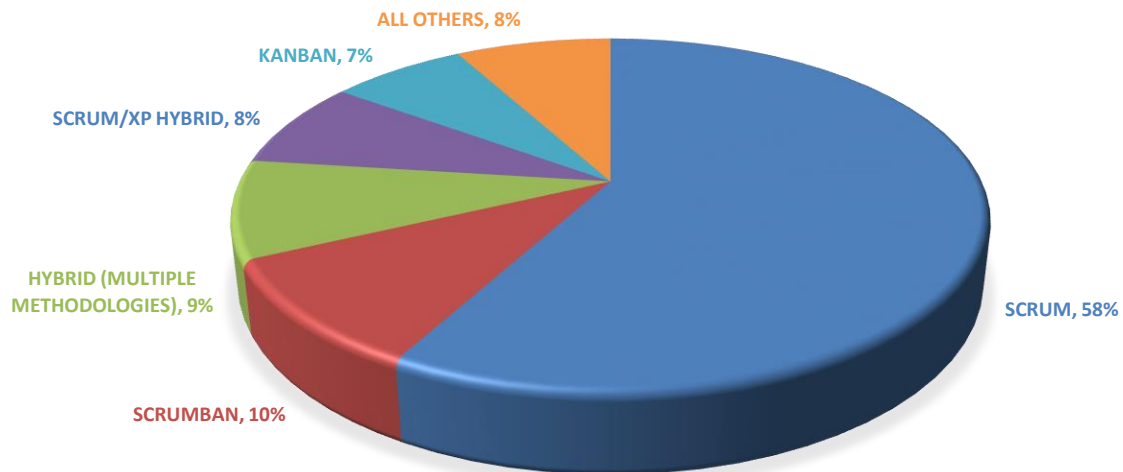


Figure 5. Agile methodologies used for implementing software projects

5. Case study: delivering a small-sized project by using Scrum

This paragraph presents the use of the Scrum methodology applied to deliver a small-sized project with a three-month estimated timeline case study. There are several steps to follow in order to reach a project timeline. Firstly, the client's requirements need to be clarified to ensure that the scope is understood and the questions that the product development team raised are entirely answered.

Secondly, a breakdown is created, containing all the features and functionalities that the customer wants to include in the product's first version. Those are negotiated until the team and the customer reaches an agreement on the scope, timeline and budget allocated for the first delivery. These features are estimated, and after considering the relationships and dependencies between the project activities and project team, a high-level timeline is created, as shown in Figure 6. This timeline is used as a baseline, together with the breakdown structure that contains detailed information about the features, documented risks, assumptions and constraints.

The dependencies play a crucial role when creating a new project roadmap. They need to be assessed, and any risks or issues should be flagged and documented in a registry so they can be tracked.

In this case study, two-week iterations (sprints) were considered. These are short periods, which are time-boxed, during which a product increment is created [11]. As presented in Figure 6, the first delivery is scheduled at the end of Sprint 2 when the first product increment is deployed on a User Acceptance Testing environment (UAT) and presented to the customer to decide if the product is ready to be released to a broader audience or not. Even if an increment is delivered at the end of each sprint, this does not mean that the product will be made available to the public by deploying it in a production environment. This decision lies with the Product Owner, which usually is the customer or a customers' representative.

The deliverable of Sprint 1 is a backlog of work items that can be picked up in Sprint 2 by the product development team and the project setup activities, which enable the team to start the work on the project. During the discovery phase and Sprint 1, the project manager will ensure that all needed project physical resources and project allocations are secured so that the work can be started immediately. At the end of the project roadmap, there is a formal acceptance phase for the product as a

whole. This usually happens in the User Acceptance Testing environment, and after the sign-off is received, a final deployment of the product in the production environment is made. Usually, after this point, the project goes into a maintenance phase.

Name	Month 1				Month 2				Month 3			
	w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4
	Sprint 1		Sprint 2		Sprint 3		Sprint 4		Sprint 5		Sprint 6	
Project Setup	Yellow	Yellow	Yellow									
Feature 1			Yellow	Yellow	Yellow	Yellow						
Feature 2					Yellow	Yellow	Yellow	Yellow	Yellow	Yellow		
Feature 3					Yellow	Yellow	Yellow	Yellow				
Quality Assurance				Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Acceptance											Pink	Pink
Delivery				Pink		Pink		Pink		Pink		Pink

Figure 6. High-level timeline for an Agile project delivery

6. Conclusions

There can be as many delivery methods as project types, which are called hybrid. It is known that the applied methodology can help deliver a project, but depending on the projects' specificity, the delivery model can be adapted to work for each specific project. The frameworks and methodologies available are seen as guidelines that can be used as tools by the project managers and product development teams to deliver the project's desired outcome.

Regarding the Agile methodology implementation methods and the Agile Toolkits [14], Scrum has proven to offer a stable framework that product development teams can quickly adapt if a Scrum Master exists to coach and offer support during the implementation process. It has been established that Scrum can offer predictability while leveraging flexibility regarding scope changes during the project executing stage. There are hundreds of projects that are implemented through hybrid models, influenced by both Scrum and Kanban. The project management software systems have incorporated these methodologies, and they can help get any project started. The templates provided can be adjusted by learning after several iterations. Regardless of the selected implementation method, the product development team should stick with it for several iterations. The implementation process should be adapted at the teams' discretion through the inspect and the adapt sessions. The Agile methodology is a practical implementation approach that offers efficacious instruments which can improve the performance of any software organisation's delivery model. Its principles can be adopted within the Waterfall projects. By leveraging the iterative implementation concept, the phases of a project can be delivered incrementally rather than having just one delivery at the end of each stage.

A typical hybrid model encountered within the information and technology industry is when the initiating and planning phases are implemented using the Waterfall approach. However, the implementation activities are done through the iterative delivery model, which Agile promotes. There are two ways in which it can be used and adopted at an organisational level so that the delivery practices can be improved:

- For junior organisations, which is new within the software delivery market, it can be quickly implemented by following the methodology's principles and adopting the Agile mindset at the organisation level to define the delivery model.
- For existing corporations, which already have a determined delivery model, a more hybrid model is recommended to blend the traditional approaches that made the organisation successful to date with the Agile principles, enhancing the value of the delivered software products.

Although the Agile methodology started to become the standard methodology in software projects implementation, several organisations are currently employing Waterfall methodology because it simply works, and it has a proven track record. Using this system, the managers feel comfortable knowing that everything is extensively documented and there is a centralised decision-making process that reduces human error. All companies will become software companies as technology advancement is focused on digitalising and automating all business processes. The performing companies need to have the capability to deliver software at a large scale. Currently, the digital age is at its turning point and it is recommended for the organizations that are using the existing Waterfall software delivery model to incorporate the Agile principles.

7. References

- [1] U. S. Bureau of Statistics 2021 *Labor f. stat. f. the cur. pop. sur.* (<https://www.bls.gov/cps/cpsaat18.htm>)
- [2] ARIES 2019 *Market survey*
- [3] Niemann J and Pisla A 2018 *Sust. poten.and risks assess in autom. and robot. using the l. c. manag. ind. tool—LY-MIT* **10** 21 (<https://doi.org/10.3390/su10124638>)
- [4] Kisielnicki J and Misiak A M 2017 *Effect. of A. comp. to W. implem. meth. in I.T. proj. anal. bas. on b. i. proj.* **3** 14
- [5] Llahm O, Faraj B, Dalla and Llahm O B D 2020 *Lean soft. dev. prac. and princ. in ter. of obs. and evol. meth. to incr. work env. produc.* **7** 24
- [6] El-Sseid A M, El-Sseid H and Llahm O B D 2020 *Scrum Dev. Proc.* **13** 31
- [7] Prusak L 2010 *What can't be measured* (<https://hbr.org/2010/10/what-cant-be-measured>)
- [8] PMI *A Guide to the Project Management Body of Knowledge*, 6th Edition
- [9] Schwaber K and Sutherland J 2020 *The Scrum Guide* **3-7** 14
- [10] Beck K, Beedle M, Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R *et al* 2001 *Manifesto for Agile Software Development* (<https://agilemanifesto.org/>)
- [11] Mahnic V and Drnovscek S 2005 *Agile soft. dev. with scrum* **1-4** 7
- [12] Moniruzzaman A B M and Hossain S A 2013 *Comp. st. on a. soft. dev. meth.* **2-4** 15
- [13] Digital.ai *The 14th Annual State of Agile Report 2020* (<https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494>)
- [14] Poppendieck M and Poppendieck T 2003 *Lean Software Development: An Agile Toolkit* (Addison-Wesley Professional) chapter 2 pp 15-45