

Risk Management in Agile and Waterfall Models: A Review

Sara Alshathry¹, Betool Alnamlah², Norah Alkassim³, Nor Shahida Mohd Jamail⁴

^{1,2,3,4}College of Computer and Information Sciences, Prince Sultan University,
Saudi Arabia

¹ssara-yousef@hotmail.com, ²batool_alnamlah@hotmail.com,

³219420554@psu.edu.sa, ⁴njamail@psu.edu.sa

Abstract

One of the essential factors to consider when choosing a suitable development method is by assessing the risks associated with the particular methodology. Most organizations prefer risk-free approaches as the best way of maximizing their profit. Risks are viewed in terms of the factors likely to cause them and how they will impact the project. This paper focuses on two popular system methodologies, the waterfall and agile models, and the assessment of the risks involved in these two methods. The survey made use of related literature analyzing risk management in these two development models. The waterfall approach is a sequential method that involves specific predetermined stages handled by different professionals. Involvement of dedicated customers in each sprint of the process eliminates the uncertainties of releasing a product that would not address the needs of its users. The adaptive nature of agile methodologies ensures that the newly-developed system addresses all the users' needs. The waterfall method is suitable for well-planned long-term projects that run within a strict budget that is specified before the development begins. Agile methodologies are ideal for projects that do not require too much formal planning and documentation.

Keywords: Agile, Waterfall Methodologies, Software Applications.

1. Introduction

Most prosperous business entities have integrated their service delivery channels to include web-based applications that provide customers with routine services. The effectiveness of an application software determines how fast the company can get a return on its investments [1]. The objective of this research paper is to present a review of various risks that are associated with agile and waterfall software development approaches. The author utilized peer-reviewed journals and other publications from different authors to determine the risks in each method, as well as suggesting possible mitigation techniques. After comparing the resources from different scholars, the researcher observed that agile methodologies are suitable for short-term projects. The waterfall model, on the other hand, requires prolonged projects as some phases may take a longer time-frame to implement [2].

The adaptive nature of agile methodologies enables the latter to monitor and mitigate risks associated with the development stages. The attribute is evident in the preliminary sprints that allow the stakeholders and customers to suggest features of the next release based on the current system [3]. Instead of relying on over-analyzed predictions, the agile model accounts for the actual organizational and customer needs when developing a new application [3]. The real-time data ensures that the newly-developed system addresses all end-user requirements at the time of release.

The agile system development method emphasizes customer satisfaction and the adaptability of the process. It integrates incremental and iterative models through the rapid delivery of a working product [4]. The system is broken into small incremental builds in the form of iterations involving cross-functional teams that work simultaneously.

Every iteration takes a maximum of three weeks that include planning, requirements specification, design, coding, testing, and release of the final version.

The use of risk boards to manage risks is an improved technique that accounts for the latter's severity. The primary purpose of breaking the task into small builds is to deliver specific features before releasing the final product [5]. Every iteration provides a working software, and each incremental build takes care of a particular need to ensure the final product addresses all the customer needs. This survey points out that no substantial research has been conducted to recommend the best development approach since their success largely depends on how well they are implemented, according to the organization's needs [6].

Since the model does not have proper planning, developers are likely to be distracted by a slight additional requirement. Such attributes will overrun the project because it will enlarge and become more complex. The increments made in each sprint are likely to make the system more complicated that the developers will need to adopt proper mechanisms of tracking their progress. The increments also increase the duration for project completion as well as additional costs. Agile approaches entail informal one-on-one communication between the developers, and having a large team can compromise the effective exchange of ideas [7]. The agile model largely relies on individual expertise, and it is likely to fail if communications between customers and the developers are ineffective [8]. Updates are released based on end-user preferences. If the customer suggests requirements that do not represent the needs of the majority, the new software is likely to be rejected by users.

The waterfall approach segregates the development lifecycle into various distinct phases that involve different processes. It is initiated by identifying the underlying problem in the first step, then drafting a long-term development plan to establish a substantial solution. To make this approach effective and risk-free, Dr. Winston Royce, the man behind the plan-oriented methodologies, developed a roadmap for a successful development that would also enhance cost-effectiveness [9].

Organizations can manage the uncertainties of their projects by clearly specifying their requirements and setting aside sufficient funds to facilitate the latter. To mitigate possible operational risks, Royce defined the objectives to be completed in every phase. This survey analyzes two commonly used software development models and points the possible causes of uncertainty in these processes.

The requirements in this methodology are usually frozen at the planning stage before the development begins. Therefore, the project cannot account for radical changes in end-user preferences, and this may hinder the customer satisfaction [10]. The sequential nature of waterfall processes means that progressing to the next phase is determined by the successful completion of the previous stage [11]. It can create uncertainties in subsequent project stages and alter the projected time-frame since one team may take longer to complete their tasks. The absence of monitors to evaluate each phase means that the final product is only tested in the last stage. Bugs detected at this step are likely to take the project back to its starting point, leading to a waste of time and resources [12]. The waterfall model is usually used for long-term projects. In a dynamic environment, the preference of users is likely to go with emerging trends. Therefore, the long-awaited system is likely to be finalized at the time when its objectives are no longer relevant to the consumers.

The objective of this paper is to establish risks in the agile and waterfall system development approaches. Since it's a comparative analysis, it is significant to acknowledge the existing literature on the same methodologies to establish the distinctive risks in each of them. The survey will focus on three papers from different authors that attempted to highlight the major risks associated with waterfall and agile development models. Such studies are meant to be used as future references and improve research in the related study fields.

The use of literature review will, therefore, seek to give a substantial meta-analysis of plan-oriented and scrum methodologies to give researchers as well as stakeholders an

insight of what they should anticipate when choosing a specific development approach. This survey cited relevant publications from various software development experts as well as researchers to certify the authenticity and relevance of the paper. Risk management is a factor that surfaces in most aspects of the entrepreneurial industry, and the techniques used in mitigating the uncertainties in software development can as well be viable in other production industries [13].

2. Research Method

There are various methodologies used in system development. Picking the most suitable methods depends on several factors, one of them being the risk associated with the project. It is advisable to consider the elements of a project that are likely to make it risky. It is also essential to evaluate the impact of the risk likely to be imposed by the proposed system. No substantial effort has been made to compare the risk factors of various system development approaches. This research paper was used to investigate the risk management statuses of the most popular development approaches.

The surveyor noted that the waterfall model is associated with unavoidable risks due to its nature. Firstly, the constant change of requirements during the development period compromises the efficiency of this model. The rigidity of this approach leaves no room for adjusting any provision, which may be necessary to cope up with the market changes [14]. Since different experts handle every stage, some team members may have to wait for long durations before they execute their duties. This is because one phase must be completed before the development proceeds to the next one. The absence of quality assurance teams in each step means that all the aspects are tested for bugs in one testing stage. If any major defects are detected here, it may take too long to be corrected, leading to a waste of time and resources [15].

In the agile model, the researchers observed that there are no proper mechanisms for assessing and managing risks. However, software systems in these models are usually extensive. The incremental processes may increase the development time frame, leading to unnecessary expenses. Large development teams can undermine the coordination of activities. The method over relies on individual brilliance, meaning that their absence directly affects the project's efficiency [16]. Since developers rely on customer feedback, they are likely to use unbiased views to make modifications to an existing application.

The traditional waterfall approaches had risk management teams to assess the risk factors of a project before commencing its development. The short iterations in agile development approaches enable them to respond to business market changes [17]. Risk management should begin at the early development phases and continue throughout the lifecycle. This research was used to analyze risk management in agile development techniques.

Agile development approaches tend to minimize risks in the system by striving to provide customers with beneficial applications. Effective risk management in this model takes place according to specific guidelines. The risk is first identified and analyzed to establish its severity. The severity of the risk is used to prioritize the identified risk [18]. An action plan is then created to handle the risks according to their priority. A follow-up is made through continuous monitoring to ensure that the action plans are effective.

2.1. The Waterfall Model and Agile Methodologies: A Comparison by Project Characteristics

The researcher of this literature attempted to compare the agile and waterfall development techniques in terms of issues arising from each process. Dr. Winston Royce asserted that large and complex software development processes are best executed using this model [19]. The specification and acquisition of the project's requirements should be made in the initial stages to clearly define its budget and eliminate the doubts of miscellaneous expenses during the development lifecycle. Proper planning and documentation at every stage ease the transfer of technology from the developers to the stakeholders and end-users [20].

However, the surveyor noted that the waterfall method had several setbacks that need to be addressed in terms of risk management. To start with, this model is likely to implement unused software features since it restricts the modification of requirements in early preliminary stages. The documentation that takes place at every phase requires more effort and funds. The model is, therefore, a baseline for contemporary system development methodologies [21].

In the agile approach, the researcher noted the active participation of customers in every development sprint. The face-to-face communication between developers enables them to act on collective decisions swiftly rather than waiting for correspondences [22]. It also reduces resource-intensive intermediate artifacts. Automated tests in every sprint minimize the risk and cost of detecting bugs in the late development stages. The stakeholders' input at the beginning any project in the agile model gives a roadmap and the objectives of the development.

However, this survey also observed a significant number of risk factors associated with agile methodologies. The informal evaluation mechanisms do not give sufficient analysis for establishing the system's quality. Furthermore, the frequent release of working products can render the model costly, and organizations may wait for long before realizing the return on their investments [23].

The research concludes that every development model serves specific purposes, and each of them has its advantages as well as drawbacks. Agile approaches are suitable for small projects or pilot development programs. Long-term projects that require voluminous documentation and structured planning are more compatible with the waterfall model [24].

3. Results And Discussion

Waterfall methodology is the most used software development method since the inception of system development because of its organization and logical procedures. However, agile remains to be the suitable model for mitigating risks due to its capability to cope with the requirements changes promptly. Most researchers have also pointed out the rigidity of requirements specification in the waterfall method, noting that it is the primary source of uncertainty. In terms of cost-effectiveness, the waterfall model requires more resources since it involves numerous documentations in every phase. The traditional approach equally tends to take more time since the stages cannot be overlapped [25]. The agile model is suitable for attaining customer satisfaction since the requirements can be altered at any time during the development process, making it the best risk mitigation methodology. The teamwork of various engineers ensures that urgent information is quickly conveyed and implemented by the right personnel before releasing the final software version to end-users. Table 1 below shows how [1], [2], and [3] analyzed risks in the two approaches:

Table 1. Comparison Between Agile and Waterfall Methodologies

<i>Author</i>	<i>Risks in Agile</i>	<i>Risks in Waterfall</i>
---------------	-----------------------	---------------------------

[1]	<ul style="list-style-type: none"> • Its incremental attribute in each sprint may expand the project size and lead to unnecessary expenses. • Overreliance on individual brilliance subjects the project to specific experts, meaning that their absence can cripple the entire development process. 	<ul style="list-style-type: none"> • Risks are inevitable due to its rigidness in the requirements specification, which does not accommodate emerging needs. • The phases cannot be overlapped, and some teams may wait for longer before executing their duties, creating uncertainties in the projected completion date.
[2]	<ul style="list-style-type: none"> • A risk board is formed to analyze potential project drawbacks and evaluate their severity. After establishing an action plan, the board meets daily to evaluate the mitigation process. • Improper planning of activities to be completed in each sprint creates uncertainties that may sophisticate the development process. 	<ul style="list-style-type: none"> • Plan-driven approaches used risk registers to monitor and manage the latter. However, too many fields in the register complicate the risk management process. • A considerable amount of time is spent on establishing the project roadmap without anticipating changes in customer preference.
[3]	<ul style="list-style-type: none"> • The frequent release of updates does not give users adequate time to evaluate previous versions. It is, therefore, difficult to identify the key performance indicators. • Informal communication between developers in a complicated environment means that developers lack proper techniques for tracking their progress. 	<ul style="list-style-type: none"> • Unnecessary features are likely to be implemented since it does not account for radical changes. • Voluminous documentation in every stage may lead to miscellaneous expenses, which would otherwise be done at once in the final development phase.

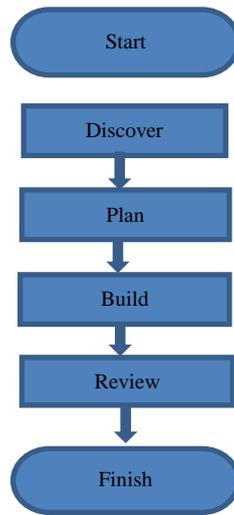


Figure 1. Waterfall development model

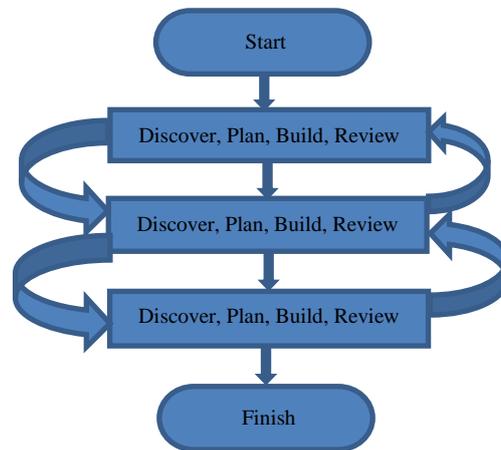


Figure 2. Agile development model

Figure 1 provides a graphical representation of the logical flow of different development phases in the waterfall model. There have 4 main phases which are Discover, Plan, Built and Review. One stage must be completely implemented before developers can proceed to the next process. In Discover phase, all the data and specification of requirements are collected. Including functional and non-functional requirement. In this phase the project team need to deal and negotiate with client of the system. The second phase is Plan which is all the analysis are done in this phase. The analysis of the data collection, the milestone, and the budget will be plan and finalize in this phase. In this phase al include the physical ang logical analysis of the system. The third phase is Build where start the implementation are started. All the functional requirement will be function by invoke the coding and script. In this phase, the programmer will take the whole responsible to make the system function. The last phase in waterfall model is Review. The verification and the validation will do in this phase. In order to do the verification and validation, there have 4 main tradition testing will be applied which are Unit testing, Integration Testing, Validation testing and System testing. This phase is the root and important for the maintenance phase. Figure 2 shows how development in agile methodologies goes through various iterations according to the

alterations made in the requirements specification. The 4 main phases in Waterfall model are running parallel in iterations. The engineers work simultaneously to deliver different parts of the system, and each iteration involves reviews to ensure that errors are rectified in time before implementation [26]. The iterations are usually repeated until the software meets the update objectives. This ensures that the updates cater for all the customer feedback on the previous release.

4. Conclusion

This study has analyzed different risks that usually arise during the software development process. The uncertainties largely depend on the nature of the preferred method, with waterfall approaches being characterized by rigidness throughout the development phases. The agile model is rapidly gaining popularity amongst developers due to its flexibility, making it the most suitable risk mitigation technique. Every methodology has its risks, and it is recommended to consider other underlying factors before settling for the most suitable approach. However, this research concludes that agile methods are rapidly being adopted by many enterprises to cope with the fast-growing technological advancements. Future scholars should also determine the opportunity cost of picking either of these approaches in terms of cost-effectiveness. This will help investors make the right decisions whenever they intend to develop new systems for their business applications. The waterfall model is a predictive development approach that outlines the project's long-term goals and approximates how long it should take to achieve them. Each phase has clear objectives to be accomplished, and this eliminates any operational uncertainties in subsequent stages. However, the requirements specification should be flexible to accommodate radical changes in the development environment. The agile methodology, on the other hand, is an adaptive approach that emphasizes customer satisfaction with minimum planning and documentation. Active customer participation ensures that developers do not release an irrelevant product to the end-users. However, much still needs to be done in terms of managing uncertainties since only a few customers give suggestions on the software features. The agile practices of continuous development and deployment do not account for feasibility surveys to establish the project's cost-effectiveness.

Acknowledgments

This work was partially supported by the Artificial Intelligence Data Analytics Lab (AIDA), Prince Sultan University, Riyadh, Saudi Arabia.

References

- [1] J. Cleland-Huang and M. Vierhauser, "Discovering, Analyzing, and Managing Safety Stories in Agile Projects", 2018 IEEE 26th International Requirements Engineering Conference (RE), vol. 7, no. 10, pp. 262-273, 2018.
- [2] A. Alsanad, A. Chikh and A. Mirza, "A Domain Ontology for Software Requirements Change Management in Global Software Development Environment", IEEE Access, vol. 7, no. 3, pp. 49352-49361, 2019.
- [3] N. Anjum and A. Kabir, "Introducing Refined Agile Model (RAM) in the Context of Bangladesh'S Software Development Environment Concentrating on the Improvement of Requirement Engineering Process", International Journal of Software Engineering & Applications, vol. 10, no. 4, pp. 9-28, 2019.
- [4] O. Thom-Manuel, C. Ugwu and L. Onyejebu, "A New Mathematical Risk Management Model for Agile Software Development Methodologies", International Journal of Software Engineering & Applications, vol. 9, no. 2, pp. 67-86, 2018.

- [5] H. Kaur, A. Sharma. "Specification of Non-Functional Requirements (NFRs) in Request for Proposal (RFP)." *International Journal of Software Engineering and Technology (IJSET)*, vol. 1, No 3, pp. 197-202, 2016.
- [6] J. Dooley, *Software development and professional practice*, 1st ed. [Berkeley, CA]: Apress, 2011.
- [7] J. Highsmith and K. Orr, *Adaptive software development*. New York, New York: Dorset House Publishing, 2013.
- [8] J.Rashid, W.Mehmood, M.W.Nisar. "A Survey of Model Comparison Strategies and Techniques in Model Driven Engineering." *International Journal of Software Engineering and Technology (IJSET)*, Vol 1, No 3, pp.165-176, 2016.
- [9] A. Elzamy and B. Hussin, "Quantitative and Intelligent Risk Models in Risk Management for Constructing Software Development Projects: A Review", *International Journal of Software Engineering and Its Applications*, vol. 10, no. 2, pp. 9-20, 2016.
- [10] J. Verner, O. Brereton, B. Kitchenham, M. Turner and M. Niazi, "Risks and risk mitigation in global software development: A tertiary study", *Information and Software Technology*, vol. 56, no. 1, pp. 54-78, 2014.
- [11] V.Eswar, S.Suresh N.Dessai. "Embedded Software Engineering Approach to Implement BCM5354 Processor Performance.", *International Journal of Software Engineering and Technology (IJSET)*, vol. 1, no. 1, pp. 41-58, 2016.
- [12] R. Bhujang and S. V., "Risk Impact Analysis across the Phases of Software Development", *Lecture Notes on Software Engineering*, vol. 2, no. 3, pp. 282-287, 2014.
- [13] L. Fink and B. Pinchovski, "It is about time: Bias and its mitigation in time-saving decisions in software development projects", *International Journal of Project Management*, vol. 38, no. 2, pp. 99-111, 2020.
- [14] C. Lindholm, "Involving user perspective in a software risk management process", *Journal of Software: Evolution and Process*, vol. 27, no. 12, pp. 953-975, 2015. Available: 10.1002/smr.1753.
- [15] G. Borrego, A. Morán, R. Palacio Cinco, O. Rodríguez-Elias and E. García-Canseco, "Review of approaches to manage architectural knowledge in Agile Global Software Development", *IET Software*, vol. 11, no. 3, pp. 77-88, 2017.
- [16] A. Elzamy, B. Hussin and N. Salleh, "Top Fifty Software Risk Factors and the Best Thirty Risk Management Techniques in Software Development Lifecycle for Successful Software Projects", *International Journal of Hybrid Information Technology*, vol. 9, no. 6, pp. 11-32, 2016.
- [17] P. Jain, "Towards the Adoption of Modern Software Development Approach: Component Based Software Engineering", *Indian Journal of Science and Technology*, vol. 9, no. 32, 2016.
- [18] R. Kumar, P. Maheshwary and T. Malche, "Inside Agile Family Software Development Methodologies", *International Journal of Computer Sciences and Engineering*, vol. 7, no. 6, pp. 650-660, 2019.
- [19] A. Dima and M. Maassen, "From Waterfall to Agile software: Development models in the IT sector, 2006 to 2018. Impacts on company management", *Journal of International Studies*, vol. 11, no. 2, pp. 315-326, 2018.
- [20] S. Ghobadi and L. Mathiassen, "Risks to Effective Knowledge Sharing in Agile Software Teams: A Model for Assessing and Mitigating Risks", *Information Systems Journal*, vol. 27, no. 6, pp. 699-731, 2016.
- [21] E. Tegegne, P. Seppänen and M. Ahmad, "Software development methodologies and practices in start-ups", *IET Software*, vol. 13, no. 6, pp. 497-509, 2019.
- [22] A. Hodijah, "Analysis Service Engineering Development Lifecycle: An Object-Oriented Approach", *Journal of Physics: Conference Series*, vol. 1090, no.1, pp. 012020, 2018.

- [23] S. Yamada, T. Aoki and T. Toyota, "Quantitative Software Quality/Reliability Prediction Based on Project Management Data for Waterfall and Agile Development Paradigms", *OPSEARCH*, vol. 45, no. 4, pp. 391-404, 2008.
- [24] A. Khalid, S. Zahra and M. Fahad Khan, "Suitability and Contribution of Agile Methods in Mobile Software Development", *International Journal of Modern Education and Computer Science*, vol. 6, no. 2, pp. 56-62, 2014.
- [25] V. Rajasekaran, "Issues in Scrum Agile Development Principles and Practices in software development", *Indian Journal of Science and Technology*, vol. 8, no. 35, 2015.
- [26] B.R. Reddy A and S. Bindu C, "A Systematic Survey on Waterfall Vs. Agile Vs. Lean Process Paradigms", *i-manager's Journal on Software Engineering*, vol. 9, no. 3, pp. 34-59, 2015.