

Research Paper: Selection of Waterfall and Agile Methodologies in Software Testing

Author: Lok Shan CHAN

Abstract

Until software is completely developed and put into operation, it goes through a series of steps. It includes the development of applications, frameworks, and other software, as well as the conception, specification, design, programming, documentation, testing, and bug fixing. Software testing is done by software engineers or test engineers to see whether the software meets its criteria, requirements, and serves its intended purpose. Software testing is an essential part of the software development process. It assists software engineers in evaluating software or applications with the goal of determining whether the developed software satisfies the stated specifications in order to deliver a high-quality end product (Balaji & Murugaiyan, 2012). Waterfall and agile software development testing processes are the two most common approaches used by software engineers and test engineers. This research compares and contrasts software testing in a waterfall and agile development process.

Main Content

The selection and preparation of which method to use before a software engineer starts his or her project is important. The approach selected has a large impact on whether a software development project succeeds or fails. The way a development team works is affected by the software testing they use. The waterfall software development testing methodology, also known as the linear sequential life cycle model, is a sequential process in which project developers only move on to the next phase of software development after the previous phase has been completed successfully. If deliverables at a specific phase fails to meet requirements, system developers re-do the phase until the desired software outcomes are achieved (Brandall, 2018). Agile methodology, on the other hand, is an approach for assisting software engineers and test engineers in the continuous iteration of development and testing of software during the development process. Unlike waterfall methodology, agile methodology allow for

concurrent development and testing of software deliverables. More collaborations among the client, software engineers, managers, and test engineers are possible with agile methodology (Van Casteren, 2017).

Since it is one of the simplest models to handle, software engineers favor waterfall methodology over agile methodology. It can be used in projects with unique deliverables at each phase. This means that software engineers move on to the next phase of the project after ensuring that the previous one complies with all project requirements. Smaller projects with clearly understood requirements benefit from the waterfall approach (Brandall, 2018). Waterfall processes and outcomes, unlike agile methodology, are well known. Processes and results are well known at every phase of the project due to the simplicity of its method. Software engineers understand what they are expected to achieve in a certain amount of time. In contrast to the agile methodology, the approach is cost effective, and software engineers are able to prevent mistakes at the early stages of project development and quickly find and correct bugs (Brandall,2018). In contrast to agile software testing, waterfall software testing makes it difficult for software engineers to make improvements to previous project deliverables. It is a form of organized software development technique that can be very rigid at times. Since the Agile methodology is so adaptable, improvements to the project can be made even after the initial preparation has been completed. Furthermore, the technique is not appropriate for large-scale projects.

Testing occurs after the build phase in a waterfall methodology, while testing occurs simultaneously with software development in an agile methodology. One of the major drawbacks of the waterfall approach is that collecting and documenting project specifications at each stage of the development process can be time-consuming and challenging (Fair, 2012). It is difficult to expect project deliverables too early in the process while using the waterfall approach (Van Casteren, 2017).As a consequence, project assumptions can be inaccurate or inconsistent with client standards and expectations. Agile approach, in contrast to the waterfall approach, is a client-centered process that ensures the client is actively involved in all project phases. This means that all of the client's expectations are met. The waterfall approach starts with a list of client specifications, and software engineers work to fulfill those requirements. Furthermore, by using the waterfall software testing process, the risk is higher since the

potential for errors is greater, and if a single error occurs, it would necessitate going back a few steps to correct the bugs (Van Casteren,2017).

Conclusion

As a conclusion, the waterfall approach tends to be a more conventional software development testing strategy that does not place a high value on client participation in the software development process. The agile testing approach tends to be the most effective and appropriate software testing methodology. The development team is able to get client input and make improvements as needed because of the high degree of client participation. Since it is a continuous process, the methodology enables early detection of defects, and software engineer are able to deliver a high-quality end product that meets the client's needs. Despite the fact that the waterfall approach has specific goals and instructions and is easy to execute, it can be costly if the client is dissatisfied. This is because making improvements to the software in the event of client disappointment can be costly.

Reference

Balaji, S. and Murugaiyan, M.S., 2012. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management, 2(1).

Brandall, B., 2018, Waterfall vs Agile: Which methodology is right for you? Accessed March18,2021 <https://www.process.st/waterfall-vs-agile/>

Van Casteren, W., 2017, The Waterfall Model and Agile Methodologies: A comparison by project characteristics.

Fair, J., 2012, Agile versus Waterfall: approach is right for my ERP project? Paper presented atPMI® Global Congress 2012—EMEA, Marsailles, France. Newtown Square, PA: Project Management Institute.

Marti, R. C., 1999, Agile Software Development – Principles, Patterns and Practices.

Copyright protected @ ENGPAPER.COM and AUTHORS



<https://www.engpaper.com>