Unit 33 1.1 I can investigate a range of development environments

When developing software there are a range of methodologies that can be applied to help develop your product.

The Waterfall model is a traditional, documentation driven form of software development methodology. The process is a sequential, relying on dependencies (and completion of milestones) within each stage to be completed before the next stage can progress. Costings, resources, timelines and requirements for the end product are determined at the start of the project and become the baseline for the entirety of the project. This process is simple and well-structured meaning it is easier to manage and the focus is on quality rather than speed. The biggest pitfall when using the Waterfall method is that the product is not delivered until the end of the project meaning that the product owner has not been able to give input during the production stage. This could mean the product does not meet the full requirements even though it fulfils the scope. The use of the Waterfall framework leads to less flexibility when major changes are required due to the impact on the baseline and scope of the project as well as the substantial documentation required for planning and making changes. Waterfall is best used when quality rather than speed is essential and when requirements for the end product are clearly defined and tangible. For example a banking app could not afford to have bugs due to the need for security and the product would therefore need a lot of time in the testing and development stages before it could be released.

Stages of Waterfall methodology:

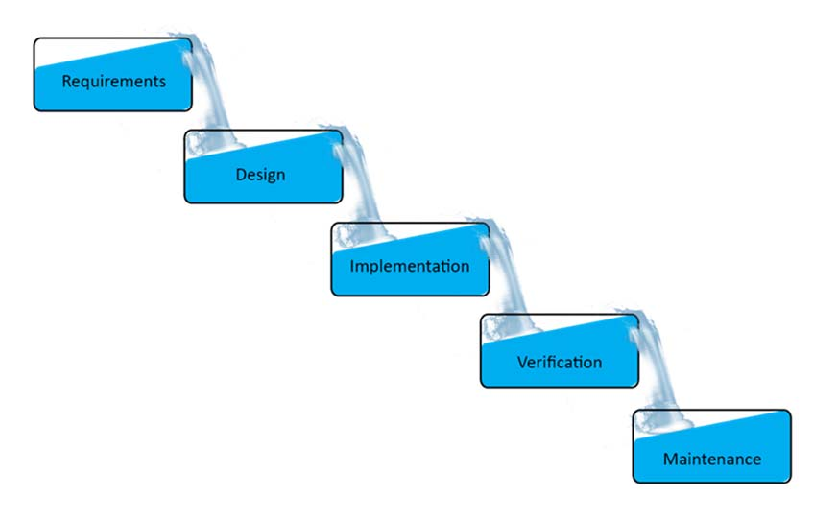


Image source: <https://www.researchgate.net/figure/Waterfall-methodology-Model_fig1_298402625>

In comparison Agile software development methodology is designed to be a fast, streamlined process, able to easily accommodate change. Requirements for the end product do not need to be as specific in comparison to using Waterfall. This is especially useful when it is hard to define the full requirements and features of the end product. Interaction with the product owner is frequent throughout the process allowing for continuous feedback. Software is developed and delivered incrementally via ‘sprints,’ a short time frame allocated to developing a useable piece of software e.g. two weeks. At the end of each sprint the product owner can give feedback and specify any changes required ahead of the next ‘sprint’. Agile focuses on delivering testable software that can continuously be improved based on the requirements of the project. The downside to Agile is that a project timeframe as well as costs and resources are not clearly defined which can present as a risk. The product owner requirements may also change frequently, especially when they are not clear on what they want from the end product or when there have been communication issues during the process.

Scrum is a structured agile framework which focuses on continuous improvement and has assigned roles responsible for delivering the project. Roles within the project team are clearly defined as follows;

* Product Owner – responsible for defining requirements of the project, ensuring they are delivering value to the customer, work effectively with stakeholders and to prioritise the product backlog (work tasks that make up the project)
* Scrum Master – also referred to as a ‘servant leader’, working closely with the Product Owner to help meet objectives as well as work with the Scrum team, organising, facilitating and leading sprint meetings and reviews
* Development Team – comprised of developers and designers who complete the work tasks that make up the product backlog

Scrum works on an incremental basis working to deliver measureable outcomes at the end of every sprint which typically take place over two weeks. There are four Scum ceremonies that make up the framework;

* Sprint planning – Sprint goals and planning for the upcoming sprint
* Daily scrum – 15 minute daily scrum meeting to update the team
* Sprint reviews – Demo any completed work and get approval for release, clarify work done/not done, review and revise the product back log
* Sprint retrospective – Evaluate how the last sprint went and identify what went well/ wrong, finalise next steps

Scrum is a flexible approach which helps to break down the overall project into smaller tasks and deliver results quickly whilst focusing on continual improvement. The use of sprints means that changes can be made quickly and applied to the next sprint. However the scrum methodology is reliant on the expertise, commitment and communication skills of the team to deliver measurable outcomes that can be improved on in the next sprint for this method to be successful. Additionally without a defined timescale projects can overrun bringing increased risks and costs.