Waterfall Methodology

By Matthew Dear

About

Waterfall is a programming methodology that was introduced by Dr Winston W Royce in a paper published in 1970. The paper describes a model of 6 logical stages that need to be followed in order to make it to project completion. However, these stages are also required to be followed in order backwards if you find something that needs changing in the design stage for instance you will have to work your way to and from that stage to make the change while updating all the other stages to get there.

The Steps:

- **Requirements**: In this stage potential requirements for the system are discussed and are written in a requirements document that is used in all remaining stages to inform the decision making. It explains what the application should be able to do but, does not explain the exact way this should be done.
- **Analysis**: In the analysis stage the requirements are analysed to generate the diagrams and business logic for the application.
- **Design**: During this stage technical requirements of the project will be decided on including, programming language, data layers, and services. More documentation will be created at this stage including UX/UI designs and Model diagram.
- **Coding**: The source code for the application is written in this stage, implementing all things discussed in previous stages.
- **Testing**: At this stage the code will go though QA and Testing to find bugs and report them to be resolved. It is normal for this stage to call for a repeat of the coding stage if bugs our found.
- **Operations**: The final stage is where the application is deployed and support and maintenance plans are put in place.

Pro's

- **Adapts to Shifting Teams**: Due to the large amount of documentation laid out in the first stages of the waterfall methodology.
- **Forces Structured Organization**: The developer is forced to be disciplined with the project documentation that is created. Making sure they have documentation to run for the complete life cycle, once again making the waterfall method more robust to team changes.
- **Allows for Early Design Changes**: Due to the large amount of documentation created at the beginning of the project more changes can be made by the client easily in the early stages of development. However, it is much harder to make changes near the end and customers will not see a prototype for an extremely long time.
- **Suited for Milestone-Focused Development**: Due to the linier nature of waterfall this is a great model for use with large organisations that require dates and a complete timeline of the project as it is much easier to develop both of those things using this model.

Con's

- **Nonadaptive Design Constraints**: The waterfall model is unable to adapt to changes quickly as a major design flaw discovered in the application at the testing phase means costly delays or potentially scrapping the project completely.
- **Ignores Mid-Process User/Client Feedback**: The waterfall model is unable to adapt to client feedback at a late stage without taking steps backwards thought the model that will be costly and time-consuming.
- **Delayed Testing Period**: Testing is not started until the penultimate stage of the model potentially finding design flaws at an extremely late stage. It also breeds lackadaisical programming as testing is seen as an afterthought rather than an integral part of the development process.