3.2 I can reflect on the efficiency of project management techniques

**Waterfall:** Critical Path Analysis

**5. Waterfall Methodology**

**5.1 What is Waterfall**

Waterfall was first described in a presentation at the Symposium on Advanced Programming Methods for Digital Computers on 29 June 1956 by Herbert D. Benington ([Wikipedia](https://en.wikipedia.org/wiki/Waterfall_model)).

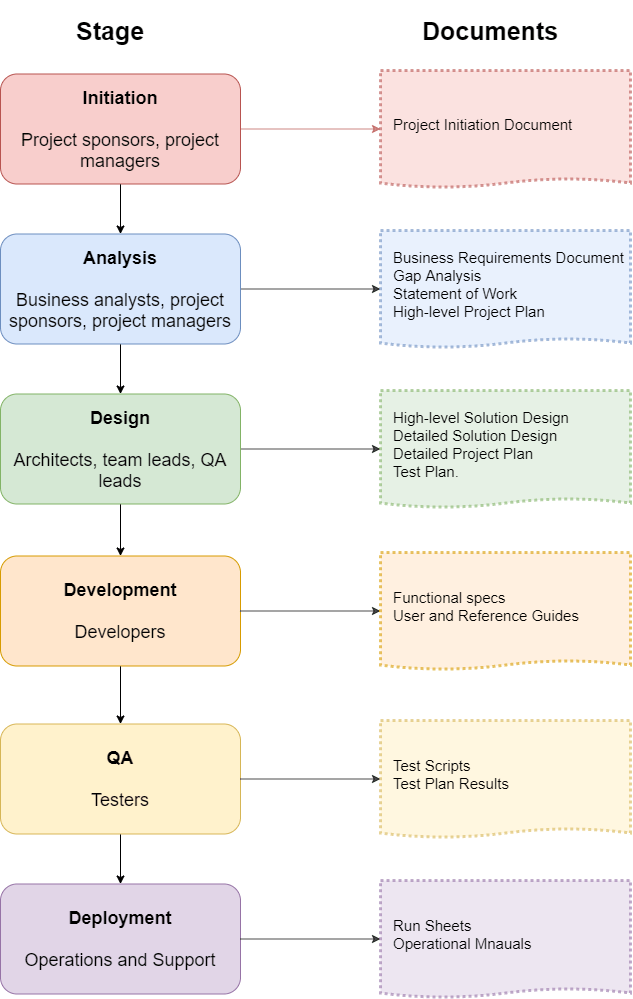
*Waterfall consists of breaking down all project tasks and activities into a pipeline of sequential stages where each stage depends on the output from its predecessor.*

Later on, in 1970, a paper published by Winston Royce on [Managing the Development of Large Software Systems](http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf) gave Waterfall its current fame and stature. The term Waterfall was never used until it appeared in a 1976 paper by Bell and Thayer.

**5.2 How Does Waterfall Work**

The Waterfall model is a series of sequential stages where each stage of the pipeline specializes in one area of the [software development lifecycle](https://softwaredominos.com/home/software-design-development-articles/).

The stages of Waterfall typically are Initiation, Analysis, [Design](https://softwaredominos.com/home/software-design-development-articles/solution-design-and-its-role-in-successful-projects/), Development, [Testing](https://softwaredominos.com/home/software-design-development-articles/software-testing-and-quality-assurance-a-modern-analysis-of-its-internal-dynamics-and-impact-on-delivery/), and Deployment.

Waterfall Stages and Documents Produced

In a [properly setup process chain](https://softwaredominos.com/home/business-management-articles/engineering-superior-production-processes-a-no-nonsense-guide-for-everyone/), each stage in the SDLC has an owner responsible for executing a set of transformations on inputs it has received to produce artifacts, which will serve as inputs to the next stage.

The table below describes the five stages and the deliverables produced in each of them.

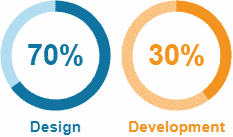
| **Waterfall Project Phase** | **Deliverables** |
| --- | --- |
| *Project Initiation* | **Project Initiation Document**: defines the project sponsors, pre-requisites, overall scope.  [**Business Requirements Document**](https://softwaredominos.com/home/software-design-development-articles/business-requirements-and-stakeholder-management-an-essential-guide-to-definition-and-application-in-it-projects/#7-Business-Requirements-Document-BRD): details the business requirements to be met.  **SLA** or Service Level Agreement. |
| *Analysis* | [**Software Orders of Magnitude**](https://softwaredominos.com/home/software-design-development-articles/software-estimation-how-to-get-it-right-the-first-time/): a list of high-level features (or Epics) and a rough order of magnitude of their efforts.  [**High-Level Design**](https://softwaredominos.com/home/software-design-development-articles/high-level-solution-design-documents-what-is-it-and-when-do-you-need-one/): represents a bird’s eye view of what the solution looks like.  [**Solution Design**](http://softwaredominos.com/home/software-design-development-articles/write-solution-design-document/): This is a blueprint of the system to be implemented.  **Detailed Project Plan**: includes all the detailed tasks required for successful delivery and a detailed effort estimation  **Gap Analysis**: usually conducted to understand the differences between what the system offers off-the-shelf and what the end result looks like.  **Statement of Work**: describes the scope, deliverables, timeframes, prerequisites and milestones.  [**Test Strategy**](https://softwaredominos.com/home/software-design-development-articles/software-testing-and-quality-assurance-a-modern-analysis-of-its-internal-dynamics-and-impact-on-delivery/)**:** a high-level plan of how the new releases will be tested. More of a guideline to testing than actual test cases at this stage. Discusses test approaches, test environments, and resourcing. |
| *Development* | **Source Code**: new version of the source code  **Test Scripts**: the set of test cases to be used for testing the changes. These also include any [Test Automation](https://softwaredominos.com/home/software-design-development-articles/test-automation-strategies-a-deepdive-into-an-essential-solution-for-your-daily-agile-practices/(opens%20in%20a%20new%20tab)) scripts or [unit test](https://softwaredominos.com/home/software-design-development-articles/unit-testing-is-it-worth-the-effort/) scripts in case [Test-Driven Development](https://softwaredominos.com/home/software-design-development-articles/test-driven-development-and-the-power-of-self-validating-code/) is used.  **User Guides and Reference Manuals**: can also be produced at this stage. |
| [*Testing and QA*](https://softwaredominos.com/home/software-design-development-articles/software-testing-and-quality-assurance-a-modern-analysis-of-its-internal-dynamics-and-impact-on-delivery/) | **Test Cases**: an actual test case with summaries, acceptance criteria, steps to produce, and test parameters.  **Test Report:** a report of the test runs with details on failed cases. |
| *Deployment* and Support | **Rollout Plan**: a step-by-step plan on how production deployment will happen. Typically with a duration for each step. This is especially important if downtime is involved.  **Installation, Admin, and Operations Guides**: these are the manuals required to run the system in production mode.  [**DevOps**](https://softwaredominos.com/home/software-design-development-articles/devops-finding-your-path-to-successful-and-continuous-improvement/)**Script:** if DevOps practices are used for building and deploying applications. |

Key Deliverables of Waterfall Stages

**5.3 Key Principles Behind Waterfall**

Waterfall project management, when compared to lightweight methodologies like Agile, presents two very distinct features.

The first feature is massive planning in the form of design and documentation, and the other is using a single sequential pipeline for delivery.

Design vs Development Efforts in Waterfall

The most significant chunks of work in the delivery pipeline are usually design and development.

Waterfall emphasizes design activities to limit the scope and impact of any changes and therefore reduce the size of testing to the bare minimum. Most experts recommend a ratio of 70/30 for design and development.

The downside of this approach is that planning and design happen only at the beginning of the project, and revisiting the plans at any later stage is very costly.

To [mitigate the risk](https://softwaredominos.com/home/business-management-articles/technical-risk-management-and-decision-analysis-introduction-and-fundamental-principles/) of any rework, Waterfall tries to get it right the first time.

**5.4 Advantages of Using Waterfall**

Despite its core weaknesses, Waterfall has many advantages:

1. *Waterfall runs on an intuitive model*that is easy to explain and understand. The sequential nature also makes it simple to manage.
2. *A fixed structure with clear milestones, checkpoints, and deliverables*requires a straightforward implementation process with basic management techniques and little hand-shaking and coordination between different teams.
3. *A fixed project scope*reduces the overhead associated with rescoping, redrafting documents and project plans, and acquiring approvals from senior management.
4. *Generous documentation.*No one generally complains about having proper and quality documentation.

These advantages, however, are not enough to justify using Waterfall. A Hybrid model can have all the benefits and very few of the disadvantages of Waterfall. It also combines the good features of Agile to make it even more appealing. We will have a few words to say on Hybrid models later on.

**5.5 When Is Waterfall Inappropriate**

Waterfall project management works great if the following criteria are met:

1. *The requirements are well known at project inception.*The chances of conditions changing later on in the project are minimal.
2. *The customer can see the end result and sign it off*before development starts so that the chances of developers building unwanted or unusable features are tiny.
3. *The technology used must be mature and well-understood.*Any issues of a technical nature are quickly resolved. Using only mature technology is one of the principles of Operational Excellence.

Points 1 and 2 are usually challenging to satisfy. Here are some reasons why that may be true:

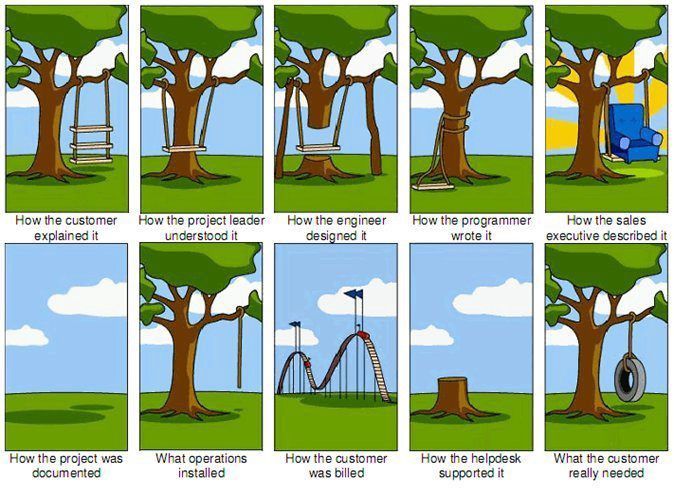
1. In some cases, such as building a website, the customer may not know what they want until they see it.
2. Using websites again as an example, you may not be able to gauge the user experience until you deploy on production. This weakness is ubiquitous for issues related to the look-and-feel of websites.
3. For sufficiently big projects, customer priorities might change. Although not as likely as points 1 and 2, this is still possible.

The below table lists a few real-world examples of scenarios where Waterfall cannot cope.

| **Scenario** | **Example** |
| --- | --- |
| **Unknown or Unclear Requirements**  *In these cases, the requirements can never be fully understood at the beginning of a project*. | 1) Migrating legacy, poorly-documented systems are an example where requirements can never be fully known before testing.  2) [Performance requirements](https://softwaredominos.com/home/software-design-development-articles/stress-and-performance-tests-for-online-systems-how-to-prepare-execute-and-analyse-their-results/) for applications that outperform expectations.  3) Websites where users’ behaviour patterns cannot be predicted.  4) Innovative products yet to be understood. |
| **Lengthy Projects** **/ Changing Requirements**  *If the duration of the project is extended, chances are that some requirements might change*. | 1) Regulatory updates in financial products are one example of requirements changing during the project.  2) Change in client priorities or business strategy. |
| **New, Poorly-Understood Technologies**  *The team is not fluent in the technology used*. | 1) Supporting new OS platforms or web framework  2) Deployment mode changes like offering the product as a service |

Waterfall Shortcomings

The below caricature is the best explanation of what usually happens.



#### Information source **Georges Lteif**