

First Generation: Linear Models

Railroad construction vs Software development

[vedio] Railroad construction

Sledgehammer vs Spike driving machine

Software development (depends on the project)

- text editor + keyboard + rough idea of what to do
- in-depth knowledge of the latest software engineering processes and practices
- know about all the options available -> decide to use the best tool for the job
- understand the variety of processes available -> make the best choice possible for a project

Process models

[recall] Last courses: Processes, Practices and Activities

What & Why useful?

In the following courses

- Some simple and useful processes [in the past]
 - deficiencies
- -> how they've evolved to create some of the more common processes we see today.
- just because one process is more evolved and advanced, does not mean that the other is now useless or obsolete.
- -> know the options available and their pros and cons.
- -> use a process appropriate for the task at hand

Linear process models



Linear process models

-> follow a pattern of phases completed one after another without repeating prior phases.

The product is designed, developed, and released **without** revisiting earlier phases.

Quiz

Please choose the linear process model from the list.

- A. each phase happens sequentially and then loops back to the beginning when all the phases are complete.
- B. each phase happens in parallel with other phases, until the product is done with no repetition between or within phases.
- C. each phase happens sequentially and never loops or repeats.
- D. each phase can be repeated, until the product is complete.

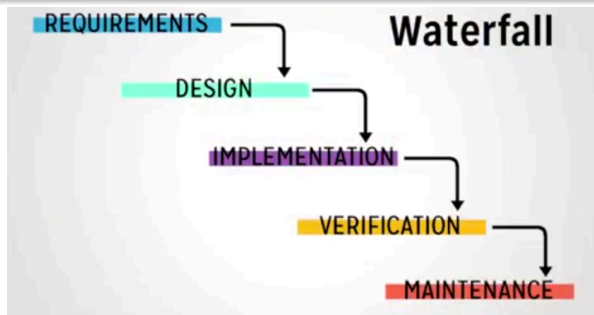
Please choose the linear process model from the list.

- A. each phase happens sequentially and then loops back to the beginning when all the phases are complete.
- B. each phase happens in parallel with other phases, until the product is done with no repetition between or within phases.
- ✓ C. each phase happens sequentially and never loops or repeats.
- D. each phase can be repeated, until the product is complete.

The Waterfall

The Waterfall

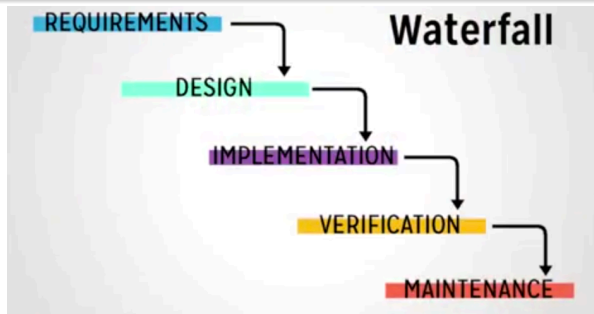
- A basic linear process.
- One thing happens after another.
- Each phase of the process is fed into by an **approved** work product from the previous phase.



The Waterfall

Example

- » requirements phase: a product requirements document (approved)
- » design: a completed a set of models, schema, and business rules (signed off).
- »



The Waterfall

Strengths: allow developers

- to get started on building a product quickly.
- to avoid the issue of changing requirements by determining their scope early and sticking to it.
 - emphasis on documenting like requirements and architecture,
 - -> capture a common written understanding of the software by the development team

Weaknesses: it is inefficient and restrictive

- not very adaptable to changes, not very agile
- -> it does not allow for the development team to review and improve upon their product.
 - The waterfall model is not designed to address midstream changes, which may require revisiting earlier phases.

The Waterfall

Problems

- the client doesn't get to see the product until the very end.
- - what if your client needs a change since the approved requirements document?
- -> the slow response frequently leads to disappointed clients
- -> unable to ensure that the work being done is appropriately verified

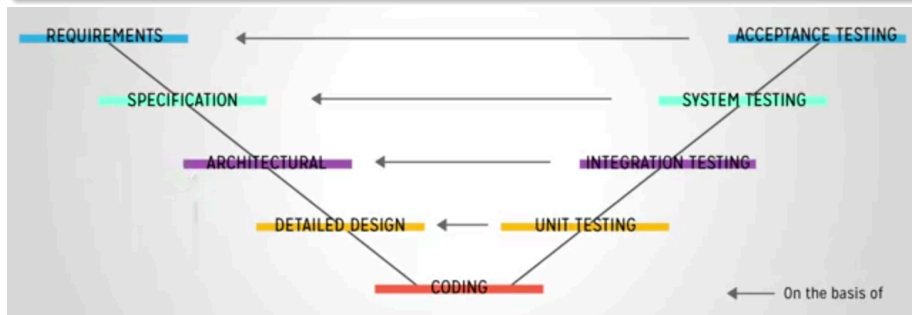
Variations of the waterfall model

-> to allow feedback opportunities to earlier phases and their activities [to support certain changes].

V-model

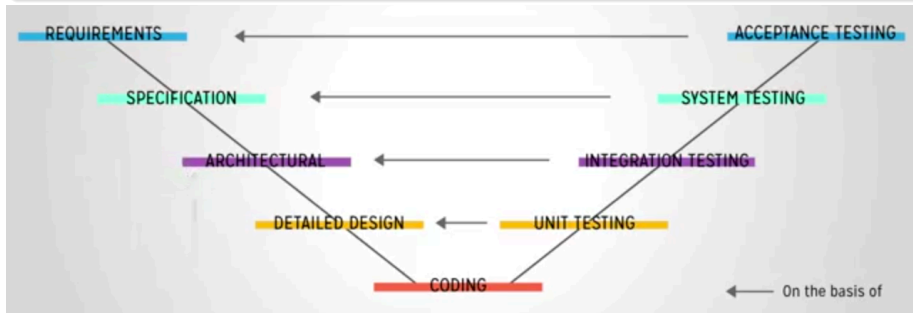
Idea: organize each level of verification to appropriate phases, rather than all at once

- similar to the waterfall model: one thing happens after another in sequential order
- difference: divides the process into two branches



V-model

- emphasizes verification activities to ensure the implementation matches the design behavior.
- ensures that the implemented design matches requirements.
 - » requirements -> system architecture and design
 - » implementation
 - » verification activities [for the corresponding phase on the left]



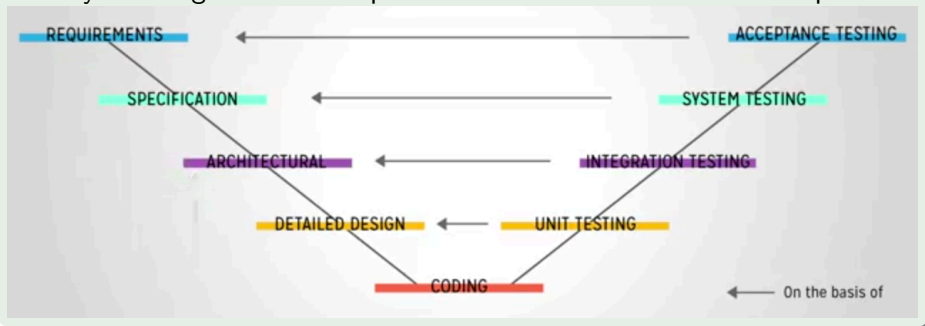
V-model

Advantages and disadvantages of the V model, same as the waterfall model

- straightforward to apply,
- doesn't account for the inevitability of change of software development.
- client doesn't get to see the finished product until the very end when everything is complete.

The V-model does allow for the development team to verify the work of constructive phases of the process.

Study our diagram which depicts the V-model of software development.



If you are in the integration testing phase, which phase are you verifying when you run your test?

- A. unit testing.
- B. coding.
- C. high level design.
- D. operational testing.

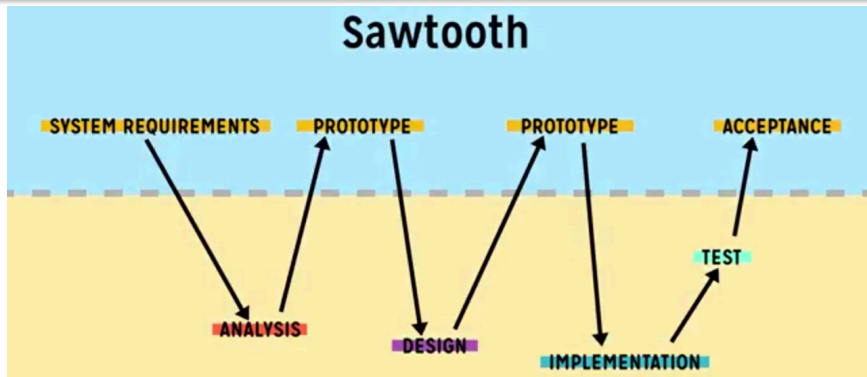
If you are in the integration testing phase, which phase are you verifying when you run your test?

- A. unit testing.
- B. coding.
- ✓ C. high level design.
- D. operational testing.

Sawtooth Model

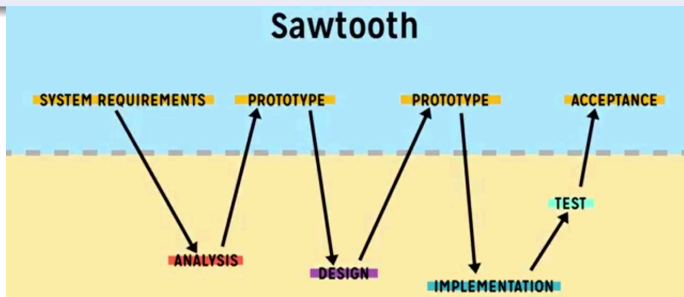
Client interaction throughout the process

- a process that allow us to involve the client along the way?
- instead of only at the end when the product is complete



Sawtooth Model

- **linear model** of software development & similar to the Waterfall and the V-model
 - > but client tasks are interspersed throughout the process, so that feedback can be gathered at meaningful times.
- **distinct part:** it distinguishes between **the client** and **the dev team**:
[tasks requiring the clients presence] **vs** [tasks **only** requiring the dev team]
- » Sawtooth Model is easy to apply & but still doesn't address change very well.



Linear process models

Three important pre-agile manifesto process models

- Waterfall model
- V-model
- Sawtooth model

Commonalities: phases are in sequence

- main reason for their shared advantages and disadvantages
 - allow development to happen in a straightforward way,
 - but greatly restrict the project to fit the process.

Linear process models

software development vs machine manufacturing

the product is machined and assembled according to certain requirements. And once produced, the product only requires minor maintenance upkeep.

» The emphasis, then, is on getting the requirements right, upfront, and not changing them afterwards.

Developing a software development is a creative endeavor -> experimentation & constant rework

Why linear models?

Why linear models?

- In the past, computer time was expensive compared to human labor.
- -> For software development, the cycle time between writing a line of code and seeing its result could be hours.
- The focus: make tasks like programming efficient for computers [though not necessarily for people].
- » Early Thinking: try to get things right the first time and avoid re-work

Avoid linear models totally?

- Documenting the internals of a software product for a new developer.
- You might still describe the project in a linear way, through the phases and associated documents.