V-MODEL

An explanation

Wyatt Kirschner, Jason Loosle, Melanie Lent

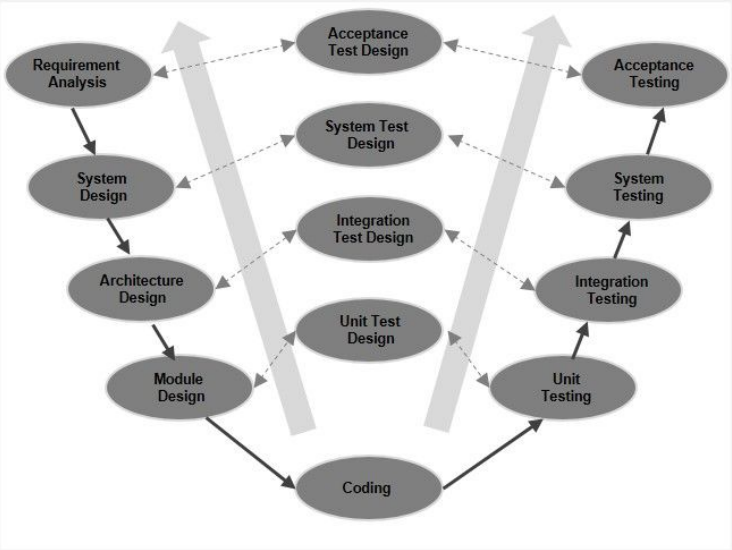
What is a V-Model?

The V-model consists of three sets of phases, which consist of *verification*, *coding*, and *validation*.

Verification comes first, and it involves planning the project from the customer's perspective, and designing and structuring the project itself.

Validation is essentially testing, making sure bugs are taken care of early on, that the system functions and communicates with itself properly, and that it will work well for the user.

As you can see to the left, you go back and forth a lot between the three.



Advantages of the V-Model! :)

There are many advantages to using this model of software process. Because of the highly disciplined approach to the software process, there is less room for error. This also allows for:

- ➔ Meticulous design
  - this creates an environment where we can be more specific with how the project should flow.
- ➔ Disciplined development and documentation
  - this helps maintain consistency throughout the process and lots of testing helps document problems early on.

The main advantage of the V-Model is the modularity of code, which is extremely important in Object Oriented Programming and the modularity of testing throughout the entire process which is just a productive way to write code. The early tester gets rid of the bug.

Disadvantages ... :(

- The focus on testing at the end of the cycle makes it easy to get pigeonholed into rushing tests at the end of the project in order to meet deadlines.
- Encourages a rigid and linear workflow making it hard to go back and make any changes once you've left the development stage which can cause setbacks, lost man-hours, and increased costs, making it poorly suited for long-term projects.

When should you use it?

Since there's such a strong emphasis on making a good plan and sticking to it, it's probably best suited for small- to medium-sized projects that can be more easily managed. Similarly, it's best for projects using resources and techniques that you understand well. Likewise, you should strongly consider the end user's experience, and technology involved should *not* be rapidly changing.

For these reasons, for example, the V-model is strongly favored by medical tech development, an environment where stability is necessary, and it is fairly clear throughout the entire process what is required.

Does anyone have questions?