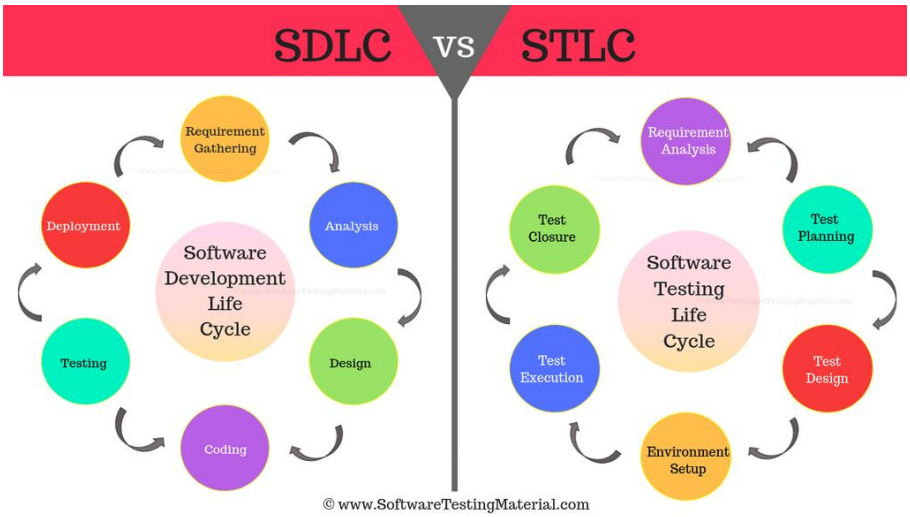
Tìm hiểu về V Model

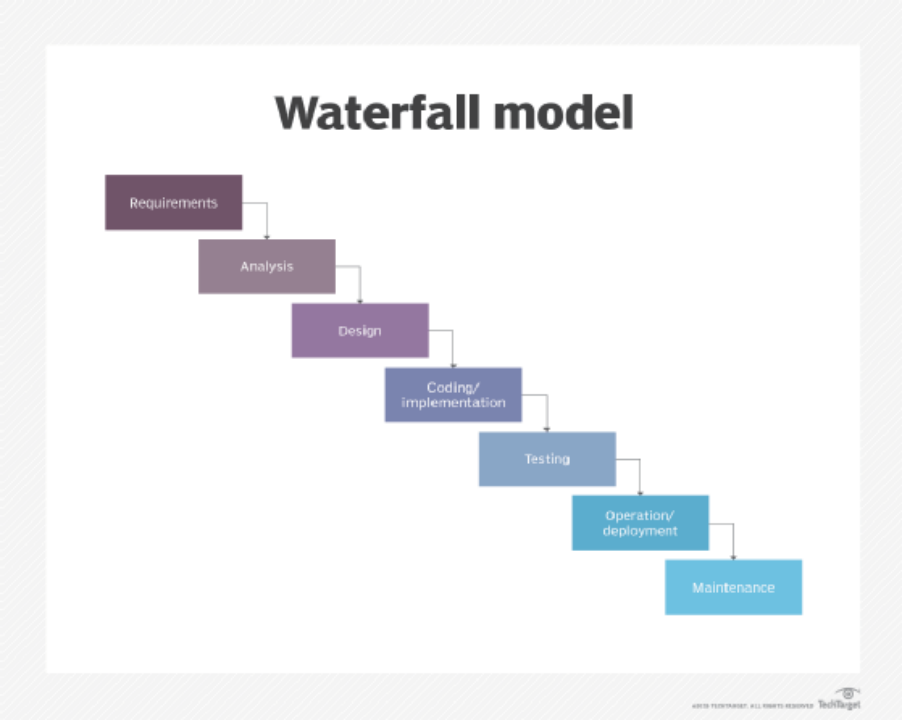
Trước khi tìm hiểu về V model, cần phải biết:

Software Development Life Cycle: chu kì phát triển phần mềm, nó là các chuỗi hoạt động do nhà phát triển thực hiện để thiết kế và phát triển phần mềm. Trong SDLC còn kết hợp các công việc được đóng gói bởi các tester và các bên liên quan

Software Testing Life Cycle: vòng đời kiểm thử phần mềm. Nó bao gồm một loại các hoạt động được thực hiển bởi tester để kiểm tra các sản phẩm phần mềm. Trong STLC đôi khi còn liên quan đến developer



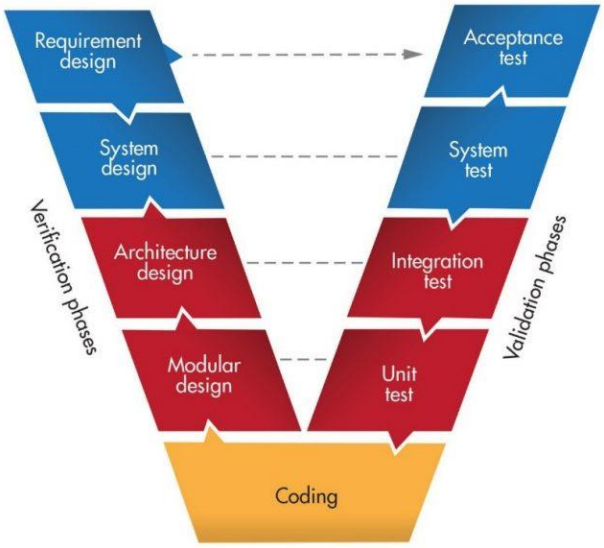
Mô hình thác nước: là mô hình tuần tự các giai đoạn khác nhau của hoạt động phát triển phần mềm. Mỗi giai đoạn được thiết kế đẻ thực hiện các hoạt động trong chu kì phát triển phần mềm. Giai đoạn test trong mô hình này chỉ được bắt đầu khi đã triển khai xong hệ thống.



Có thể thấy răng việc test chỉ được tiến hành sau khi đã lập trình xong. Nhưng khi làm việc trong dự án lớn, các hệ thống phức tạp, bạn có thể sẽ bỏ lỡ các chi tiết chính trong giai đoạn lấy yêu cầu. Trong trường hợp như vậy, một sản phẩm không chính xác sẽ được bàn giao cho khách hàng thì có thể phải làm lại từ đầu. Hoặc cho dù bạn có quản lý các yêu cầu một cách chính xác nhưng gặp lỗi nghiêm trọng trong thiết kế và kiến trúc phần mềm bạn sẽ phải thiết kế lại toàn bộ phần mềm để sửa lỗi.

V- model

V- model chỉ là mở rộng của mô hình thác nước, giúp giải quyết được vấn đề chính của mô hình này. Mỗi giai đoạn trong quy trình phát triển thì đều đi kèm một giai đoạn test. Đây là mô hình cực kì nghiêm ngặt và giai đoạn tiếp theo chỉ bắt đầu sau khi giai đoạn trước đã hoàn thành



Các giai đoạn testing và phát triển tương ứng được sắp xếp song song. Nên giai đoạn xác minh nằm 1 bên và giai đoạn thẩm định nằm bên còn lại. Giai đoạn lập trình kết nối 2 bên

Giai đoạn xác minh:

- Requirement Design

Đây là giai đoạn đầu trong tiến trình phát triển khi mà yêu cầu sản phẩm được làm rõ dưới góc nhìn khách hàng. Giai đoạn này cần giao tiếp một cách kĩ lưỡng với khách hàng để hiểu nguyện vọng và yêu cầu của họ. Giai đoạn này rất quan trọng vì đôi khi khách hàng không hiểu họ cần gì

- System Design

Khi mà đã có yêu cầu sản phẩm cụ thể, tiếp theo là phải thiết kế hệ thống. Giai đoạn này sẽ là hiểu và chi tiết hoá toàn bộ phần cứng và thiết lập giao tiếp cho sản phẩm khi phát triển. Kế hoạch test hệ thống sẽ dựa trên thiết kế hệ thống. Thực hiện điều này sớm sẽ dành ra đc nhiều thời gian để thực hiện test về sau.

- Architecture Design

Các thông số kỹ thuật kiến trúc được hiểu và thiết kế trong giai đoạn này. Thông thường thì sẽ có nhiều hơn một cách tiếp cận được đề xuất và dựa trên tính khả thi để đưa ra quyết định cuối cùng. Thiết kế hệ thống được chia nhỏ hơn nữa thành các mô-đun đảm nhận các chức năng khác nhau. Đây cũng được gọi là High Level Design (HLD).

Việc truyền dữ liệu và giao tiếp giữa các mô-đun bên trong và với ngoại vi được xác định rõ ràng và các bài test tích hợp có thể được thiết kế trong giai đoạn này.

- Modular Design

Xác định thiết kế chi tiết bên trong cho tất cả các mô-đun hệ thống, được gọi là Thiết kế mức thấp (LLD). Điều quan trọng là thiết kế phải tương thích với các mô-đun khác trong kiến trúc hệ thống và các hệ thống bên ngoài khác. Các bài unit test là một phần thiết yếu của bất kỳ quá trình phát triển nào và giúp loại bỏ tối đa các lỗi từ rất sớm. Các bài unit test này có thể được thiết kế ở giai đoạn này dựa trên các thiết kế mô-đun bên trong.

- Coding

Lập trình các mô-đun hệ thống được thiết kế trong giai đoạn thiết kế được thực hiện trong giai đoạn Lập trình. Ngôn ngữ lập trình phù hợp nhất được quyết định dựa trên yêu cầu về hệ thống và kiến trúc.

Việc Lập trình được thực hiện dựa trên các hướng dẫn và tiêu chuẩn. Code phải trải qua nhiều lần đánh giá mã và được tối ưu hóa để đạt hiệu suất tốt nhất trước khi bản build cuối cùng được đưa vào kho lưu trữ.

Giai đoạn thẩm định

- Unit Testing

Các bài Unit Test được thiết kế trong giai đoạn thiết kế mô-đun được thực thi trên code. Unit Test là bài kiểm tra code và giúp loại bỏ lỗi ở giai đoạn đầu, mặc dù không thể phát hiện tất cả các lỗi bằng phương pháp này.

- Integration Testing

Integration Testing liên kết với giai đoạn thiết kế kiến trúc. Integration Testing được thực hiện để kiểm tra sự kết hợp và giao tiếp của các mô-đun nội bộ trong hệ thống.

- System Testing

System Testing được liên kết trực tiếp với giai đoạn thiết kế hệ thống. Giai đoạn này kiểm tra toàn bộ chức năng và giao tiếp của hệ thống đang được phát triển với các hệ thống bên ngoài. Hầu hết các vấn đề về tương thích phần mềm và phần cứng có thể được phát hiện trong quá trình này.

- Acceptance Testing

Acceptance Testing được liên kết với giai đoạn phân tích yêu cầu kinh doanh và liên quan đến việc thử nghiệm sản phẩm trong môi trường người dùng. Acceptance Testing phát hiện ra các vấn đề tương thích với các hệ thống khác có sẵn trong môi trường người dùng. Nó cũng phát hiện ra các vấn đề không hoạt động như lỗi hiệu suất trong môi trường người dùng thực tế.

Ứng dụng của V-model

Dưới đây là một số các hoàn cảnh phù hợp cho việc sử dụng V-model

- Yêu cầu đặt ra rõ ràng và cố định
- Công nghệ không biến động và phải được hiểu rõ bởi nhóm phát triển
- Dự án ngắn

V-Model - Ưu và nhược

Ưu điểm:

- Mô hình nghiêm ngặt và các giai đoạn được hoàn thành lần lượt
- Hoạt động tốt với các dự án nhỏ mà có yêu cầu rõ ràng
- Đơn giản trong cách hiểu và sử dụng
- Dễ dàng quản lí bởi tính nghiêm ngặt của mô hình. Mỗi giai đoạn có các phân phối cụ thể và một quy trình xem xét.

Nhược điểm:

- Rủi ro cao và không chắc chắn.
- Không phải là một mô hình tốt cho các dự án hướng đối tượng và phức tạp.
- Hoạt động kém với các dự án dài và đang diễn ra.
- Không phù hợp với các dự án mà các yêu cầu có nguy cơ thay đổi.
- Khi một ứng dụng đang trong giai đoạn thử nghiệm, rất khó để quay lại và thay đổi một chức năng.
- Không có phần mềm nào hoạt động được sản xuất cho đến cuối vòng đời