



Các mô hình trong phát triển phần mềm

Bởi:

Tiêu đề: Các mô hình trong phát triển phần mềm

Bài viết dưới đây sẽ cho bạn kiến thức tổng quan về các mô hình phát triển phần mềm
Các mô hình SEP

Mô hình SEP còn được gọi là chu trình hay vòng đời phần mềm (SLC - Software Life Cycle). SLC là tập hợp các công việc và quan hệ giữa chúng với nhau diễn ra trong quá trình phát triển phần mềm. Có khá nhiều mô hình SLC khác nhau, trong đó một số được ứng dụng khá phổ biến trên thế giới:

Các mô hình một phiên bản (Single-version models)

- Mô hình Waterfall (Waterfall model)
- Mô hình chữ V (V-model)
- Các mô hình nhiều phiên bản (Multi-version models)
- Mô hình mẫu (Prototype)
- Mô hình tiến hóa (Evolutionary)
- Mô hình lặp và tăng dần (Iterative and Incremental)
- Mô hình phát triển ứng dụng nhanh (RAD)
- Mô hình xoắn (Spiral)

Mô hình Waterfall

Mô hình này bao gồm các giai đoạn xử lý nối tiếp nhau.

Phân tích yêu cầu và tài liệu đặc tả (Requirements and Specifications): là giai đoạn xác định những “đòi hỏi” (“What”) liên quan đến chức năng và phi chức năng mà hệ thống phần mềm cần có. Giai đoạn này cần sự tham gia tích cực của khách hàng và kết thúc bằng một tài liệu được gọi là “Bản đặc tả yêu cầu phần mềm” hay SRS (software requirement specification), trong đó bao gồm tập hợp các yêu cầu đã được duyệt (reviewed) và nghiệm thu (approved) bởi những người có trách nhiệm đối với dự án (từ phía khách hàng). SRS chính là nền tảng cho các hoạt động tiếp theo cho đến cuối dự án.

Phân tích hệ thống và thiết kế (System Analysis and Design): là giai đoạn định ra “làm thế nào” (“How”) để hệ thống phần mềm đáp ứng những “đòi hỏi” (“What”) mà khách hàng yêu cầu trong SRS. Đây là chính là cầu nối giữa “đòi hỏi” (“What”) và mã (Code) được hiện thực để đáp ứng yêu cầu đó.

Hiện thực và kiểm thử từng thành phần (Coding and Unit Test): là giai đoạn hiện thực

“làm thế nào” (“How”) được chỉ ra trong giai đoạn “Phân tích hệ thống và thiết kế”.

Kiểm thử (Test): giai đoạn này sẽ tiến hành kiểm thử mã (code) đã được hiện thực, bao gồm kiểm thử tích hợp cho nhóm các thành phần và kiểm thử toàn hệ thống (system test). Một khâu kiểm thử cuối cùng thường được thực hiện là nghiệm thu (acceptance test), với sự tham gia của khách hàng trong vai trò chính để xác định hệ thống phần mềm có đáp ứng yêu cầu của họ hay không.

Cài đặt và bảo trì (Deployment and Maintenance): đây là giai đoạn cài đặt, cấu hình và huấn luyện khách hàng. Giai đoạn này sửa chữa những lỗi của phần mềm (nếu có) và phát triển những thay đổi mới được khách hàng yêu cầu (như sửa đổi, thêm hay bớt chức năng/đặc điểm của hệ thống).

Thực tế cho thấy đến những giai đoạn sau mới có khả năng nhận ra sai sót trong những giai đoạn trước và phải quay lại để sửa chữa. Đây chính là kiểu waterfall dạng lặp (Iterative Waterfall).

Mô hình chữ V

Trong mô hình Waterfall, kiểm thử được thực hiện trong một giai đoạn riêng biệt. Còn với mô hình chữ V, toàn bộ qui trình được chia thành hai nhóm giai đoạn tương ứng nhau: phát triển và kiểm thử. Mỗi giai đoạn phát triển sẽ kết hợp với một giai đoạn kiểm thử tương ứng.

Tinh thần chủ đạo của V-model là các hoạt động kiểm thử phải được tiến hành song song (theo khả năng có thể) ngay từ đầu chu trình cùng với các hoạt động phát triển. Ví dụ, các hoạt động cho việc lập kế hoạch kiểm thử toàn hệ thống có thể được thực hiện song song với các hoạt động phân tích và thiết kế hệ thống.

Mô hình mẫu

Mô hình mẫu (prototype). Trong đó, qui trình được bắt đầu bằng việc thu thập yêu cầu với sự có mặt của đại diện của cả phía phát triển lẫn khách hàng nhằm định ra mục tiêu tổng thể của hệ thống phần mềm sau này, đồng thời ghi nhận tất cả những yêu cầu có thể biết được và sơ lược những nhóm yêu cầu nào cần phải được làm rõ.

Sau đó, thực hiện thiết kế nhanh tập trung chuyển tải những khía cạnh thông qua prototype để khách hàng có thể hình dung, đánh giá giúp hoàn chỉnh yêu cầu cho toàn hệ thống phần mềm. Việc này không những giúp tinh chỉnh yêu cầu, mà đồng thời giúp cho đội ngũ phát triển thông hiểu hơn những gì cần được phát triển. Tiếp theo sau giai đoạn làm prototype này có thể là một chu trình theo mô hình waterfall hay cũng có thể là mô hình khác.

Chú ý, prototype thường được làm thật nhanh trong thời gian ngắn nên không được xây dựng trên cùng môi trường và công cụ phát triển của giai đoạn xây dựng phần mềm thực sự sau này. Prototype không đặt ra mục tiêu tái sử dụng cho giai đoạn phát triển thực sự sau đó.

Mô hình tiến hóa

Mô hình này thực sự cũng là một dạng dựa trên mô hình mẫu, tuy nhiên có sự khác biệt:

- mô hình tiến hóa xây dựng nhiều phiên bản prototype liên tiếp nhau.
- những phiên bản prototype trước sẽ được xây dựng với mục tiêu có thể tái sử dụng trong những phiên bản sau.

Mô hình lặp và tăng dần

Mô hình lặp và tăng dần có lúc được hiểu là một. Tuy nhiên, trong bài viết này, ta có thể phân biệt ít nhiều sự khác biệt.

Trước tiên, hai mô hình này đều có điểm giống nhau là đều dựa trên tinh thần của mô hình tiến hóa, và có thêm đặc điểm nhắm đến việc cung cấp một phần hệ thống để khách hàng có thể đưa vào sử dụng trong môi trường hoạt động sản xuất thực sự mà không cần chờ cho đến khi toàn bộ hệ thống được hoàn thành (trong mô hình mẫu hay tiến hóa, các phiên bản mẫu hay trung gian đều không nhắm đến đưa vào vận hành thực sự cho khách hàng, trừ phiên bản cuối cùng). Để khách hàng có thể sử dụng, mỗi phiên bản đều phải được thực hiện như một quy trình đầy đủ các công việc từ phân tích yêu cầu với khả năng bổ sung hay thay đổi, thiết kế, hiện thực cho đến kiểm nghiệm và có thể xem như một quy trình (chu trình) con. Các chu trình con có thể sử dụng các mô hình khác nhau (thông thường là waterfall). Hình 5 minh họa hai mô hình này, trong đó mỗi chu trình con là một waterfall nhỏ.

Mục tiêu của phiên bản đầu tiên là phát triển phần lõi và nhóm các chức năng quan trọng. Sau mỗi phiên bản được đưa vào sử dụng, các kết quả đánh giá sẽ được phản hồi và lập kế hoạch cho chu trình con của phiên bản tiếp theo để thực hiện:

- Những thay đổi cho phiên bản trước đó nhằm đáp ứng nhu cầu khách hàng tốt hơn
- Có thể thêm những chức năng hoặc đặc điểm bổ sung
- Sự khác nhau giữa hai mô hình tăng dần và lặp có thể được hiểu đơn giản như sau (so với sản phẩm được hoàn thành trong chu trình con trước):
- Mô hình tăng dần (Incremental): thêm chức năng vào sản phẩm (xem minh họa Hình 6).
- Mô hình lặp (Iterative): thay đổi sản phẩm (xem minh họa Hình 6)
- Một SEP có thể kết hợp cả hai mô hình lặp lẫn tăng dần, chẳng hạn RUP (Rational Unified Process).

Mô hình phát triển nhanh

Mô hình phát triển nhanh (RAD - Rapid Application Development) chính là mô hình tăng dần với chu kỳ phát triển cực ngắn. Để đạt được mục tiêu này, RAD dựa trên phương pháp phát triển trên cơ sở thành phần hóa hệ thống cùng với việc tái sử dụng các thành phần thích hợp. RAD thích hợp cho những hệ thống quản lý thông tin.

Mô hình xoắn

Mô hình này được xây dựng bởi Barry Boehm, đặt trọng tâm phân tích rủi ro và xem xét kế hoạch để giải quyết chúng, thông qua nhiều chu kỳ con nối tiếp được lặp liên tiếp dựa trên bản chất của mô hình lặp.

Trong mô hình này, việc phân tích và giải quyết những vấn đề có rủi ro cao tập trung vào thiết kế từng khía cạnh cụ thể chứ không dựa vào việc xử lý các vấn đề một cách chung chung.

Trgon đó mỗi chu kỳ bao gồm 4 giai đoạn con như sau:

1. Xác định mục tiêu chất lượng cho sản phẩm được thực hiện, đồng thời xác định sự lựa chọn mua, tái sử dụng hay tự thiết kế và hiện thực các thành phần của hệ thống.
2. Phân tích sự lựa chọn và các rủi ro có thể xảy ra. Việc này được thực hiện bởi nhiều hoạt động khác nhau thông qua làm mẫu hay mô phỏng.

Các mô hình trong phát triển phần mềm

3. Phát triển và kiểm định sản phẩm ở mức tiếp theo dựa trên kết quả định hướng được chỉ ra trong giai đoạn con số 2 (phân tích rủi ro)
4. Kiểm duyệt tất cả các kết quả của các giai đoạn con xảy ra trước đó và lập kế hoạch cho chu kỳ lặp tiếp theo.

Chúng tôi luôn cập nhật kiến thức công nghệ hay tại: <https://www.facebook.com/hauisoftware>