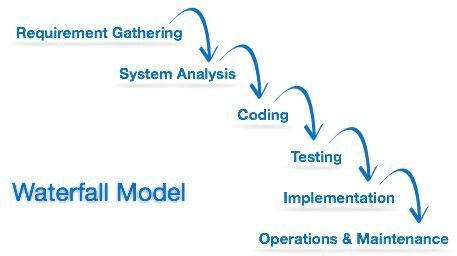
1. *Nêu khái niệm, các pha, ưu điểm và nhược điểm của mô hình Thác nước (water full)*
2. *Nêu khái niệm, các pha, ưu điểm và nhược điểm của mô hình Bản mẫu*
3. *Nêu khái niệm, các pha, ưu điểm và nhược điểm của mô hình Tiếp cận lặp*
4. *So sánh mô hình Thác nước và mô hình Bản mẫu*

**Mô hình thác nước ( Waterfall model)**



**Mô tả**

* Đây được coi như là mô hình phát triển phần mềm đầu tiên được sử dụng.
* Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm.
* Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.

**Phân tích mô hình**

* **Requirement gathering**: Thu thập và phân tích yêu cầu được ghi lại vào tài liệu đặc tả yêu cầu trong giai đoạn này.
* **System Analysis**: Phân tích thiết kế hệ thống phần mềm, xác định kiến trúc hệ thống tổng thể của phần mềm.
* **Coding**: Hệ thống được phát triển theo từng unit và được tích hợp trong giai đoạn tiếp theo. Mỗi Unit được phát triển và kiểm thử bởi dev được gọi là Unit Test.
* **Testing**: Cài đặt và kiểm thử phần mềm. Công việc chính của giai đoạn này là kiểm tra và sửa tất cả những lỗi tìm được sao cho phần mềm hoạt động chính xác và đúng theo tài liệu đặc tả yêu cầu.
* **Implementation**: Triển khai hệ thống trong môi trường khách hàng và đưa ra thị trường.
* **Operations and Maintenance**: Bảo trì hệ thống khi có bất kỳ thay đổi nào từ phía khách hàng, người sử dụng.

**Ứng dụng**

Mô hình thường được áp dụng cho các dự án phần mềm như sau:

* Các dự án nhỏ , ngắn hạn.
* Các dự án có ít thay đổi về yêu cầu và không có những yêu cầu không rõ ràng.

**Ưu điểm**

* Dễ sử dụng, dễ tiếp cận, dễ quản lý.
* Sản phẩm phát triển theo các giai đoạn được xác định rõ ràng.
* Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm các lỗi.

**Nhược điểm**

* Ít linh hoạt, phạm vi điều chỉnh hạn chế.
* Rất khó để đo lường sự phát triển trong từng giai đoạn.
* Mô hình không thích hợp với những dự án dài, đang diễn ra, hay những dự án phức tạp, có nhiều thay đổi về yêu cầu trong vòng đời phát triển.
* Khó quay lại khi giai đoạn nào đó đã kết thúc.

**Mô hình bản mẫu**

Mô hình được bắt đầu bằng việc thu nhập yêu cầu với sự có mặt của các đại diện của các phía phát triển lẫn khách hàng nhằm định ra các mục tiêu tổng thể của hệ thống phần mền sau này.

Sau đó thực hiện thiết kế nhanh, tập trùn chuyển tải những khía cạnh thông qua prototype để khách hàng có thể hình dung, đánh giá giúp hoàn chỉnh yêu cầu cho toàn bộ hệ thống phần mềm.

Tiếp theo sau là giai đoạn làm bản mẫu này có thể là một chu trình theo mô hình waterfall hay cũng có thể là mô hình khác.

Ưu điểm:

Người sử dụng sớm hình dung ra chức năng và đặc điểm của hệ thống

Cải thiện được sự liên lạc giữa nhà phát triển và người sử dụng

Nhược điểm:

Khi bản mẫu không thể chuyển tải hết các chức năng, đặc điểm của hệ thống phần mềm thì người dùng có thể thất vọng và mất đi sự quan tâm đến hệ thống được phát triển.

Bản mẫu thường được làm nhanh, thậm chí vội vàng theo kiểu”hiện thực-sửa” và có thể thiếu sự phân tích đánh giá một cách cản thận tất cả các khía cạnh của hệ thống cuối cùng.

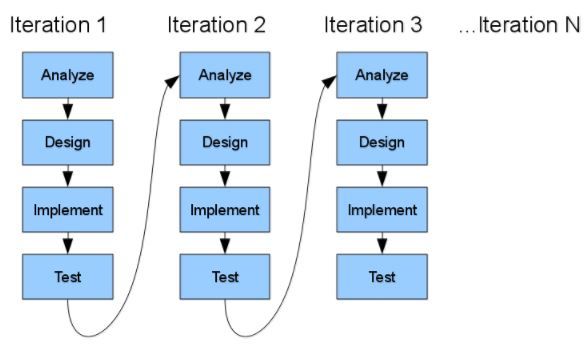
Mô hình này vẫn chưa thể cải thiện được việc loại trừ khoảng cách giữa các yêu cầu và ứng dụng cuối cùng.

Ứng dụng:

Hệ thống dựa trên giao diện người dùng

Khách hàng, nhất là người dùng cuối cùng không thể xác định rõ ràng yêu cầu.

**Mô hình tiếp cận lặp**



**Mô tả**

* Một mô hình được lặp đi lặp lại từ khi start cho đến khi làm đầy đủ spec.Quá trình này sau đó được lặp lại, tạo ra một phiên bản mới của phần mềm vào cuối mỗi lần lặp của mô hình.
* Thay vì phát triển phần mềm từ spec đặc tả rồi mới bắt đầu thực thi thì mô hình này có thể review dần dần để đi đến yêu cầu cuối cùng.

**Ứng dụng**

* Yêu cầu chính phải được xác định; tuy nhiên, một số chức năng hoặc yêu cầu cải tiến có thể phát triển theo thời gian.
* Một công nghệ mới đang được sử dụng và đang được học tập bởi nhóm phát triển trong khi làm việc trong dự án.
* Phù hợp cho các dự án lớn và nhiệm vụ quan trọng.

**Ưu điểm**

* Xây dựng và hoàn thiện các bước sản phẩm theo từng bước.
* Thời gian làm tài liệu sẽ ít hơn so với thời gian thiết kế.
* Một số chức năng làm việc có thể được phát triển nhanh chóng và sớm trong vòng đời.
* Ít tốn kém hơn khi thay đổ phạm vi, yêu cầu.
* Dễ quản lý rủi ro.
* Trong suốt vòng đời, phần mềm được sản xuất sớm để tạo điều kiện cho khách hàng đánh giá và phản hồi.

**Nhược điểm**

* Yếu cầu tài nguyên nhiều.
* Các vấn đề về thiết kế hoặc kiến trúc hệ thống có thể phát sinh bất cứ lúc nào.
* Yêu cầu quản lý phức tạp hơn.
* Tiến độ của dự án phụ thuộc nhiều vào giai đoạn phân tích rủi ro.

SO SÁNH MÔ HÌNH THÁC NƯỚC VÀ MÔ HÌNH BẢN MẪU:

|  |  |
| --- | --- |
| Mô hình Waterfall | Mô hình mẫu |
| Ưu điểm: Các giai đoạn được định nghĩa, với đầu vào và đầu ra rõ ràng. Mô hình này cơ bản dựa trên tàiliệu nhất là trong các giai đoạn đầu, đầu vào và đầu ra đều là tài liệu. Sản phẩm phần mềm được hình thành thông qua chuỗi các hoạt động xây dựng phần mềm theotrình tự rõ ràng | Ưu điểm: Người sử dụng sớm hình dung ra chức năng và đặc điểm của hệ thống. Cải thiện sự liên lạc giữa nhà phát triển và người sử dụng |
| Nhược điểm: Đòi hỏi tất cả yêu cầu phần mềm phải được xác định rõ ràng ngay từ đầu dự án. Nhưng đa số dựán thực tế cho thấy yêu cầu phần mềm thường ẩn chứa không nhiều thì ít những điểm khôngchắc chắn.  Một thực tế là các dự án hiếm khi được thực hiện đầy đủ các bước trong suốt chu kỳ dự án. Đặc biệt là giai đoạn kiểm thử khi gần đến ngày giao hàng chẳng hạn, nếu có trục trặc xảy ra do yêucầu phần mềm không rõ ràng hay thiết kế có lỗi, xu hướng là mã nguồn được sửa đổi trực tiếpmà không qua các bước bổ sung theo đúng mô hình, nên dẫn đến bản đặc tả phần mềm cũng nhưmột số sản phẩm trung gian khác như bản thiết kế, cho dù có được cập nhật sau này cũng có thểkhông phản ánh đầy đủ những gì đã được sửa đổi trong mã nguồn. Người sử dụng không có cơ hội tham gia trong suốt thời gian của các giai đoạn trung gian từ thiết kế cho đến kiểm thử. Đặc biệt với những dự án lớn, người sử dụng chỉ có thể nhận ra rằnghệ thống phần mềm không phù hợp cho nhu cầu của họ vào thời điểm cuối dự án. Nói chung, mô hình này thường ẩn chứa nhiều rủi ro mà chỉ có thể phát hiện ở giai đoạn cuối cùng  và chi phí để sửa chữa có thể rất cao. | Nhược điểm: Khi mẫu (prototype) không chuyển tải hết các chức năng, đặc điểm của hệ thống phần mềm thì người sử dụng có thể thất vọng và mất đi sự quan tâm đến hệ thống sẽ được phát triển.Prototype thường được làm nhanh, thậm chí vội vàng, theo kiểu "hiện thực - sửa" và có thể thiếusự phân tích đánh giá một cách cẩn thận tất cả khía cạnh liên quan đến hệ thống cuối cùng. Nói chung mô hình này vẫn chưa thể cải thiện được việc loại trừ khoảng cách giữa yêu cầu vàứng dụng cuối cùng. |
| Ứng dụng: Yêu cầu được định nghĩa rất rõ ràng, chi tiết và hầu như không thay đổi, thường xuất phát từ sản phẩm đã đạt mức ổn định. Yêu cầu mới bổ sung (nếu có) cũng sớm được xác định rõ ràng, đầy đủ từ đầu dự án. Đội ngũ thực hiện quen thuộc và hiểu rõ tất cả yêu cầu của dự án, và có nhiều kinh nghiệm vớicác công nghệ được dùng để phát triển sản phẩm. Dự án được xác định hầu như không có rủi ro | Ứng dụng: Hệ thống chủ yếu dựa trên giao diện người dùng (GUI) Khách hàng, nhất là người sử dụng cuối, không thể xác định rõ ràng yêu cầu. |