1: Quy trình phần mềm gồm những giai đoạn nào ?

**6 giai đoạn phát triển phần mềm**

Quy trình phát triển phần mềm bao gồm 6 giai đoạn**:**

## ****Giai đoạn 1: Khảo sát dự án****

## ****Giai đoạn 2: Phân tích hệ thống****

## ****Giai đoạn 3: Thiết kế****

## ****Giai đoạn 4: Thực hiện****

## ****Giai đoạn 5: Kiểm thử****

## ****Giai đoạn 6: Triển khai và bảo trì****

2: So sánh Agile and Waterfall ? Ưu và nhược điểm của mỗi mô hình

Waterfall được coi là mô hình phát triển phần mềm đầu tiên được sử dụng. Đây là mô hình áp dụng theo tính tuần tự của các giai đoạn phát triển phần mềm; giai đoạn sau chỉ được thực hiện tiếp khi giai đoạn trước đã kết thúc.

phát

Agile là một trong những biện pháp triển phần mềm vô cùng linh hoạt. Nó hỗ trợ hướng tiếp cận cụ thể phần mềm cho đến quản lý dự án phần mềm. Từ đó, tạo ra một quá trình làm việc có tương tác lớn và được tích hợp để đưa sản phẩm đến với người tiêu dùng càng nhanh càng tốt.

## ****Ưu điểm và nhược điểm của mô hình Waterfall****

### Ưu điểm

* Là một trong những mô hình dễ nhất để quản lý. Về bản chất, mỗi giai đoạn có quá trình cụ thể, có điểm bắt đầu và điểm kết thúc. Điểm kết thúc của giai đoạn này là điểm đầu vào của giai đoạn kế tiếp và cứ thế “cuốn chiếu” cho đến hết dự án.
* Hoạt động tốt cho các dự án có kích thước nhỏ, các yêu cầu dễ hiểu.
* Quá trình và kết quả được ghi nhận một cách “hữu hình” hơn là các kết quả nhỏ nhưng thiếu tính hoàn thiện tổng thể. Đa phần khách hàng không muốn tham gia review một tính năng dở dang.
* Phương pháp dễ điều chỉnh cho các đội chuyển dịch
* Phương pháp quản lý dự án này có lợi cho việc quản lý các phụ thuộc.

### **Nhược điểm**

* Không phải là một mô hình lý tưởng cho một dự án kích thước lớn.
* Nếu yêu cầu không rõ ràng ngay từ đầu thì đó là phương pháp kém hiệu quả hơn.
* Rất khó di chuyển trở lại giai đoạn trước đó để thay đổi.
* Quá trình thử nghiệm bắt đầu khi quá trình phát triển kết thúc. Do đó, có nguy cơ rất lớn là các lỗi về sai nghiệp vụ, sai luồng thiết kế, sai cấu trúc dữ liệu… được phát hiện “muộn màng” sau khi giai đoạn phát triển đã hoàn thành, do đó rất tốn kém để sửa chữa sai lầm.

Alige

* **Ưu Điểm:**
* Cách tiếp cận này mang lại lợi ích cho các dự án có mục tiêu linh hoạt có thể trải qua thay đổi tầm trung.
* Là tập hợp các nguyên lý của các mô hình phát triển phần mềm linh hoạt và nhanh chóng.
* Giảm thời gian cần thiết để tận dụng một số tính năng của hệ thống Kết quả cuối cùng là phần mềm chất lượng cao trong thời gian ít nhất có thể và sự hài lòng của khách hàng.
* **Nhược Điểm:**
* Không phù hợp với những dự án lớn, và với những dự án đã có những nguyên tắc cố định
* Thời hạn và chi phí của dự án rất khó thông báo và tất cả các bên liên quan cũng phải dành thời gian để tham gia tích cực vào tiến độ hàng ngày.
* Tóm lại

## ****Kết luận****

* Mô hình Waterfall lý tưởng cho các dự án đã xác định được yêu cầu và không có thay đổi nào trong quá trình phát triển. Trong khi đó, Agile là phù hợp nhất với các dự án hoặc giai đoạn có yêu cầu thường xuyên thay đổi hơn.
* Thác nước dễ quản lý, mặc dù mọi công đoạn theo quy trình tuần tự và cứng nhắc.
* Agile rất linh hoạt và có thể thay đổi trong bất kỳ giai đoạn nào.
* Trong quá trình Agile, các yêu cầu có thể thay đổi thường xuyên. Tuy nhiên, trong mô hình Waterfall, các yêu cầu được xác định rõ ràng từ đầu và thống nhất bởi tất cả các bên. Có rất nhiều mô hình dự án “bắt buộc” phải triển khai Waterfall, thí dụ dự án xây dựng lại một hệ thống đã có bằng một giải pháp công nghệ mới, tiên tiến nhưng nghiệp vụ về cơ bản không không thay đổi nhiều.
* Trong mô tả Agile của dự án, các chi tiết có thể được thay đổi bất cứ lúc nào trong quá trình SDLC mà không thể thực hiện được trong phương thức Waterfall.

3: Biểu đồ Usecase dùng để làm gì

Biểu đồ chức năng, hay sơ đồ tình huống sử dụng (tiếng Anh *Use case diagram*) mô tả sự tương tác đặc trưng giữa người dùng và hệ thống. Mục đích là để nắm bắt các chức năng cốt lõi của hệ thống và hình dung các tương tác của tương tác của nhiều thứ khác nhau được gọi là tác nhân với các chức năng khác nhau

Nói chung ,sơ đồ này được sử dụng để :

-phân tích yêu cầu hệ thống

- thiết kế phần mềm trưc quan

-Nắm bắt các chức năng hệ thống

Mô hình hóa ý tưởng cơ bản đằng sau hệ thống

Chuyển tiếp và đảo ngược kĩ thuật của một hệ thống sử dụng các trường hợp thử nghiệm khác nhau.

-CHỉ định bối cảnh hệ thống

3.1: Làm thế nào để xác định được usecase

Use case là đối tượng người dùng muốn nhận được từ hệ thống. **Nó được đặt tên giống Động từ hoặc Động từ + cụm danh từ.** **Tên Use case thường ngắn gọn, rõ ràng, cụ thể và miêu tả đủ nghĩa của đối tượng người dùng**. Những động từ như “do”, “perform”, các danh từ như ”data”, “information” nên tránh nếu có thể.

3.2: Làm thế nào để xác định được actor

**Actor (người dùng):**Đây là thành phần chỉ người dùng hay một đối tượng bên ngoài bất kỳ có khả năng tương tác với hệ thống. Để xác định được chính xác người dùng hay đối tượng đó có phải là actor hay không, bạn nên xem xét thông qua một số câu hỏi như sau:

* Ai là người sử dụng (tác nhân chính) những chức năng chính của hệ thống?
* Ai là admin (tác nhân phụ) của hệ thống, ai là người thực hiện quản lý, cài đặt và bảo vệ hệ thống?
* Ai là người cần đến hệ thống hỗ trợ để các tác vụ thường ngày được thực hiện nhanh chóng?
* Những trang thiết bị phần cứng nào cần có sự xử lý và làm việc của hệ thống?
* Những hệ thống nào khác mà hệ thống này cần thiết phải thực hiện việc tương tác?
* Người input các dữ liệu vào trong hệ thống (đối với hệ thống thực hiện lưu trữ dữ liệu) là ai?
* Ai là người quan tâm tới những giá trị mà hệ thống có thể mang lại?

4: UML là gì ? UML ứng dụng trong phát triển phần mềm như thế nào

UML **là ngôn ngữ mô hình hóa hợp nhất dùng để biểu diễn hệ thống**. Nói một cách đơn giản là nó dùng để tạo ra các bản vẽ nhằm mô tả thiết kế hệ thống. Các bản vẽ này được sử dụng để các nhóm thiết kế trao đổi với nhau cũng như dùng để thi công hệ thống (phát triển), thuyết phục khách hàng, các nhà đầu tư v.v.

* UML cung cấp cho người dùng một ngôn ngữ mô hình hoá trực quan sẵn sàng để dùng và có ý nghĩa:
  + Cho phép phát triển và trao đổi những mô hình mang nhiều ý nghĩa.
  + Cung cấp khả năng mở rộng và chuyên môn hoá để mở rộng những khái niệm cốt lõi.
  + Độc lập với ngôn ngữ lập trình chuyên biệt và các tiến trình phát triển.
  + Cung cấp nền tảng về sự hiểu biết ngôn ngữ mô hình hoá.
  + Khuyến khích và hỗ trợ sự phát triển của các công cụ hướng đối tượng.
  + Hỗ trợ những khái niệm phát triển cấp độ cao như collaboration, framework, pattern and component.
  + Tích hợp một cách tốt nhất với thực tiễn.

## các loại biểu đồ UML

### 1. Biểu đồ ca sử dụng (Usecase)

* Usecase là một yêu cầu chức năng trong hệ thống
* Usecase miêu tả sự tương tác của người dùng cuối với hệ thống

### 2,Biểu đồ class

* Biểu đồ lớp cho phép chúng ta diễn tả các lớp trong UML
* Biểu đồ lớp mô tả cấu trúc tĩnh của hệ thống thông qua các lớp và các mối quan hệ giữa chúng

5. Yêu cầu phần mềm là gì ? Phân biệt yêu cầu chức năng và phi chức năng

Yêu cầu phần mềm là tất cả những nhu cầu tính năng sản phẩm mà người dùng muốn, bao gồm chức năng, hiệu năng, giao diện,… Các yêu cầu thường xoay quanh 4 nhóm sau: yêu cầu về phần cứng; yêu cầu về phần mềm, yêu cầu về data (dữ liệu) và cuối cùng là những yêu cầu về con người. Khi nhận các brief từ khách hàng, đội phát triển phần mềm phải tiến hành tìm hiểu, **phân tích yêu cầu phần mềm**để cuối cùng lập ra một bản đặc tả chuẩn chỉnh nhất.

## ****5 giai đoạn trong quá trình phân tích yêu cầu phần mềm****

### **Giai đoạn 1. Trước khi phân tích**

### **Giai đoạn 2. Phân tích yêu cầu phần mềm và thảo luận với khách hàng**

### **Giai đoạn 3. Mô hình hóa các yêu cầu phần mềm**

### **Giai đoạn 4. Đặc tả yêu cầu phần mềm**

### **Giai đoạn 5. Định dạng đặc tả yêu cầu**

Yêu cầu chức năng được định nghĩa là sự mô tả của chức năng hoặc dịch vụ của phần mềm hay hệ thống.

Thông thường, yêu cầu chức năng sẽ chỉ ra một hành vi hoặc một chức năng. Ví dụ phần mềm hay hệ thống phải có chức năng:

Hiển thị tên, kích thước, khoảng trống có sẵn và định dạng của một ổ đĩa flash được kết nối với cổng USB. Chức năng thêm khách hàng hay in hóa đơn.

Yêu cầu phi chức năng bao gồm tất cả những yêu cầu mà yêu cầu chức năng không có. Chúng chỉ ra những tiêu chí để đánh giá hoạt động của hệ thống thay vì hành vi. Ví dụ:

Thay đổi dữ liệu trong cơ sở dữ liệu nên được cập nhật cho tất cả người dùng sử dụng hệ thống trong 2 giây.

* Yêu cầu chức năng: mô tả chức năng hoặc dịch vụ của phần mềm hay hệ thống
* Yêu cầu phi chức năng: mô tả những ràng buộc và tính chất của phần mềm hay hệ thống

Vì vậy, trong thực tế yêu cầu phi chức năng sẽ được đánh giá là có phần quan trọng hơn. Nếu không thỏa mãn được các yêu cầu này thì phần mềm hoặc hệ thống sẽ không thể đưa vào sử dụng.

5.1: Các phương pháp để lấy yêu cầu phần mềm.

Quá trình thu nhập thu nhập thông tin về các hệ thống được yêu cầu và hiện có ,đồng thời chắt lọc các yêu cầu của người dùng và hệ thống từ thông tin này

Lấy yêu cầu bằng cách phỏng vấn :

Phỏng vấn kín dựa trên sanh sách câu hỏi xác định trước

Phỏng vấn mở các vấn đề khác nhau được tìm hiểu các bên liên quan

Làm th ế nào để Q & A có hi ệu qu ả ? ⎫

Xác định vấn đề: không rõ ràng, có được để biết thêm thông tin,…

Kiểm tra trong tất cả các tài li ệu mà khách hàng cung cấp để đảm b ảo câu h ỏi c ủa b ạn đã không được gi ải quyết.

Với câu hỏi kỹ thuật, kiểm tra đội / nhóm / công ty hoặc hỏi "Google" để giải quyết nó trước khi hỏi ra

⎫ Cung cấp cho các tham chi ếu chéo rõ ràng, hoàn toàn

⎫ Đính kèm màn hình ẫu, gi ới thi ệu, g ợiý của bạn n ếu có

⎫ Chuy ển đổi câu h ỏi Y / N ho ặc các lo ại tr ắc nghi ệm n ếu có th

ể ⎫ Trong Q & A, cho th ời h ạn mà b ạn mu ốn nh ận được câu tr ả l ời. N ếu không có câu tr ả l ời cho đến th ời h ạn, tác động là gì?

⎫ Hãy đọc l ại các câu h ỏi trước khi g ửi

6: Biểu đồ activity diagram dùng để làm gì ?

: Dùng để mô tả hoạt động trong một chức năng của hệ thống hay có thể hiểu là mô tả luồng xử lý của use case.Mô tả hoạt động chính và mối quan hệ giữa các hoạt động này trong quy trình

7: Các bước để lập kế hoạch của một dự án phần mềm?

### **Bước 1: Project Goal - Xác định mục tiêu dự án**

### **Bước 2: Project deliverable - Xác định thời gian chuyển giao**

### **Bước 3: Project Schedule - Xác định lộ trình dự án**

### **Bước 4: Supporting plans - Kế hoạch hỗ trợ**

## 

8: Liệt kê danh sách các loại kiểm thử phần mềm?

**Kiểm thử chức năng**: Chú trọng đến chức năng của chương trình, bảo đảm các chức năng của hệ thống thỏa mãn đúng yêu cầu.

**Kiểm thử cấu trúc**: Đảm bảo các thành phần bên trong của chương trình chạy đúng và chú trọng đến các thành phần cấu trúc nội tại của chương trình.

**Kiểm thử hiệu năng**: Kiểm thử việc vận hành của hệ thống, đảm bảo tối ưu việc phân bổ tài nguyên hệ thống nhằm đạt các chỉ tiêu như thời gian xử lý hay đáp ứng câu truy vấn.

**Kiểm thử khả năng chịu tải**: Kiểm thử các giới hạn của hệ thống, bảo đảm hệ thống vận hành đúng dưới áp lực cao.