**BÁO CÁO SEMINAR**

# I/ GIỚI THIỆU CHUNG VỀ QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

## 1/Khái niệm về quy trình phát triển phần mềm

Quy trình phát triển phần mềm là một tập hợp các hoạt động tổ chức mà mục đích của chúng là xây dựng và phát triển phần mềm.

Một quy trình tốt và hợp lí luôn tạo ra những sản phẩm đạt tiêu chuẩn. Nó giúp tương tác hóa các hoạt động và yếu tố với nhau một các nhịp nhàng, đem lại hiệu quả.

Có thể cho rằng quy trình phần mềm đem lại chất lượng, năng suất, giá thành phần phềm, từ đó tăng tính cạnh tranh và đem lại lợi nhuận cao cho doanh nghiệp.

Những câu hỏi được đặt ra ở đây là:

* Nhân sự: Ai sẽ làm? Ai làm gì?
* Thời gian: Khi nào làm? Làm mất bao nhiêu thời gian?
* Phương pháp: Làm như thế nào?
* Công cụ: Dùng công cụ gì để làm công việc này?
* Chi phí: Chi phí bỏ ra bao nhiêu? Thu về bao nhiêu? (ước tính)
* Mục tiêu: Mục tiêu hướng đến là gì?

Mỗi loại hệ thống khác nhau thì cần những quy trình phát triển khác nhau.

## 2/Các hoạt động cơ bản của quy trình phát triển phần mềm

Có 4 thao tác là nền tảng của hầu hết các quy trình phát triển phần mềm:

* Đặc tả phần mềm: Định nghĩa được các chức năng, điều kiện hoạt động của phần mềm.
* Phát triển phần mềm: Là quá trình xây dựng các đặc tả.
* Đánh giá phần mềm: Phầm mềm phải được đánh giá để chắc chắn rằng ít nhất có thể thực hiện những gì mà tài liệu đặc tả yêu cầu.
* Tiến hóa phần mềm: Đây là quá trình hoàn thiện các chức năng cũng như giao diện để ngày càng hoàn thiện phần mềm cũng như các yêu cầu đưa ra từ phía khách hàng.

## 3/ Các mô hình phát triển phần mềm thường gặp

* Mô hình Waterfall (Waterfall model)
* Mô hình chữ V (V-model)
* Mô hình tiến hóa (Evolutionary)
* Mô hình phát triển ứng dụng nhanh (RAD)
* Mô hình xoắn (Spiral)
* Mô hình lập trình Agile

# II/ KHÁI NIỆM MÔ HÌNH WATERFALL

## 1/Khái niệm mô hình Waterfall

* Mô hình thác nước (*waterfall model*) là một mô hình của quy trình phát triển phần mềm, trong đó quy trình phát triển trông giống như một dòng chảy, với các pha được thực hiện theo trật tự nghiêm ngặt và không có sự quay lui hay nhảy vượt pha là: phân tích yêu cầu, thiết kế, triển khai thực hiện, kiểm thử, liên kết và bảo trì

## 2/Đặc điểm mô hình Waterfall

* Việc phát triển phần mềm dựa trên một tập hợp các giai đoạn (phases) có thứ tự liên tiếp, trật tự của các giai đoạn là xác định, và các kết quả của một giai đoạn trước sẽ được sử dụng làm đầu vào (input) cho các giai đoạn sau.
* Một khi tiến trình phát triển phần mềm kết thúc và hệ thống phần mềm được bàn giao cho khách hàng, thì hệ thống phần mềm sẽ không thể được thay đổi, điều chỉnh

# III/ CÁC PHA THỰC HIỆN

## 1/Phân tích:

* Các yêu cầu được tập hợp và phân tích để đánh giá tính khả thi.
* Các thông tin được ghi lại chi tiết trong tài liệu đặc tả yêu cầu
* Đối tác tham gia: khách hàng, người dùng, người phân tích
* Tài liệu yêu cầu: Danh sách yêu cầu, Mô tả chi tiết yêu cầu
* Xác định và phân tích các yêu cầu cho hệ thống
* Chức năng
* Ràng buộc

## 2/Thiết kế :

* Xác định các yêu cầu kỹ thuật về phần cứng và hệ thống giúp xác định kiến trúc hệ thống tổng thể của phần mềm
* Đầu vào: tài liệu yêu cầu, môi trường triển khai
* Đầu ra: tài liệu thiết kế
* Tài liệu thiết kế:
* Mô hình kiến trúc hệ thống
* Mô hình thiết kế chi tiết, Dữ liệu, Thuật toán, Giao diện

## 3/Mã hóa:

* Dựa trên các tài liệu ở phần thiết kế, thực hiện viết code tạo ra phần mềm.
* Kiểm tra, giám sát mã lệnh.
* Gỡ lỗi (*Debugging*).
* Đầu vào: tài liệu thiết kế
* Đầu ra: chương trình thực hiện được tài liệu chương trình.

## 4/Kiểm thử

* 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
* Đảm bảo phần mềm thỏa mãn yêu cầu khách hàng.
* Đầu vào: tài liệu yêu cầu, tài liệu thiết kế, chương trình, tài liệu chương trình
* Đầu ra: tài liệu kết quả kiểm thử

## 5/Bảo trì:

* Phần mềm được cài đặt và đưa vào sử dụng thực tế.
* Sửa các lỗi không phát hiện ở giai đoạn trước đó.
* Cải tiến các chức năng và đáp ứng yêu cầu mới

# IV/ ƯU VÀ NHƯỢC ĐIỂM CỦA WATERFALL

## 1/ Ưu điểm:

* Đơn giản, dễ áp dụng, quy trình tuần tự theo từng bước một.
* Dễ quản lý vì có tính cố định theo từng bước.
* Các giai đoạn xử lý rõ ràng và hoàn thành cùng một thời điểm.
* Dễ dàng theo dõi và quản lý: Vì Waterfall rất đơn giản, có cùng cấu trúc và cách tiếp cận ở mọi giai đoạn, nên dù dự án bạn đang thực hiện là gì, bạn vẫn thấy dễ hiểu, dễ theo dõi và dễ quản lý. Bạn không cần phải có kiến thức hay hiểu biết trước nào về một trong các phương pháp quản lý dự án hàng đầu trước khi bắt đầu làm việc với Waterfall. Nó cũng rất cứng nhắc trong vấn đề về các thành phẩm và bản sửa đổi (mỗi giai đoạn đều có một danh sách cụ thể các công việc / hoạt động / cột mốc cần được thực hiện trước khi chuyển sang giai đoạn tiếp theo), vì vậy dự án rất dễ để kiểm soát.
* Ít rủi ro hơn – Mỗi giai đoạn của Waterfall đều đòi hỏi một lượng công việc nhất định cần hoàn thành và kiểm tra lại, nhờ đó bạn sẽ có thêm cơ hội để tìm kiếm, sửa chữa các sai sót / vấn đề và khắc phục chúng tại chỗ, trước khi sang đến giai đoạn kế tiếp.
* Tài liệu hóa mọi thứ - Waterfall chú trọng vào tài liệu trong từng giai đoạn, giúp bạn dễ dàng truyền đạt cách tiếp cận của mình đến khách hàng và các bên liên quan. Hơn nữa, khách hàng có thể tham khảo tài liệu mỗi khi họ cần thêm thông tin chi tiết (như các thông tin về chi phí, qui mô, thời gian, v.v...).

## 2/Nhược điểm:

* Độ linh hoạt không cao, khi một công đoạn trong quá trình thực hiện mà phát hiện ra sai sót thì phải dừng lại toàn bộ và trở về giai đoạn đầu tiên để update dữ liệu và cập nhật lại công đoạn đó.
* Không có sản phẩm nguyên mẫu từ ban đầu, không đáp ứng được yêu cầu dịch vụ.
* Sự rủi ro và độ chắc chắn kém.
* Không thích hợp cho những dự án có độ phức tạp cao, những dự án dài hơi, dự án hướng đối tượng
* Không thể thực hiện thay đổi dễ dàng – Một khi giai đoạn của dự án đã hoàn thành, rất khó khăn, tốn kém để quay lại và thay đổi giai đoạn đó khi cần thiết. Ví dụ, khi một nhóm phát triển phần mềm thiết kế ra một sản phẩm, họ nhận thấy rằng một tính năng bị thiếu, do đó họ bị buộc phải quay lại, bắt đầu lại từ đầu để thêm vào tính năng này.
* Thời hạn dài - Bạn không thể tự lựa chọn sẽ làm gì vào thời điểm nào, bạn cần theo đúng quy trình của dự án và phải hoàn thành một giai đoạn trước khi bắt đầu giai đoạn tiếp theo, nên bạn sẽ mất rất nhiều thời gian để đạt được kết quả cuối cùng, hoặc thậm chí chỉ để giới thiệu một bản demo cho khách hàng.
* Xác định dữ liệu ngay từ đầu - Giai đoạn đầu của phương pháp Waterfall có lẽ là thời điểm quan trọng và khó khăn nhất. Nó đòi hỏi phải có sự giao tiếp với khách hàng và các bên liên quan, phải hiểu chính xác kết quả cuối cùng mà họ muốn đạt được là gì. Mặc dù ở hầu hết các trường hợp, khách hàng đều rất khó để xác định chính xác những gì họ muốn ngay từ đầu, cũng như họ không thể vẽ ra trong đầu tiến trình phát triển của dự án. Cách tiếp cận này thực sự không hiệu quả lắm nếu sử dụng phương pháp Waterfall .

**Khi nào thì dùng mô hình Waterfall ?**

* Mô hình này chỉ thích hợp khi các yêu cầu đã được tìm hiểu rõ ràng
* Những thay đổi sẽ được giới hạn một cách rõ ràng trong suốt quá trình thiết kế
* Những dự án nên là dự án ngắn hạn
* Hiểu rõ tài liệu và công nghệ sử dụng

# V/SO SÁNH WATERFALL VỚI MÔ HÌNH RAD

|  |  |
| --- | --- |
| **WATERFALL** | **RAD** |
| **Dễ dàng theo dõi và quản lí**: Vì Waterfall rất đơn giản, có cùng cấu trúc và cách tiếp cận ở mọi giai đoạn, nên dù dự án bạn đang thực hiện là gì, bạn vẫn thấy dễ hiểu, dễ theo dõi và dễ quản lý. | **Khó lên kế hoạch**: Khá là khó để xác định rõ ràng thời gian bàn giao sản phẩm cuối cùng, vì trong mô hình RAD các thành phần hoặc chức năng được phát triển song song như thể chúng là các dự án nhỏ |
| **Ít rủi ro hơn** | **Rủi ro cao**: cần có một team giỏi để xác định yêu cầu phần mềm  Yêu cầu về dev/ design phải có nhiều kinh nghiệm |
| **Cần thời gian** để đưa ra một kết quả cụ thể thậm chí là bản demo | **Nhanh chóng** đưa ra một cái gì đó cho khách hàng để xem và sử dụng và cung cấp thông tin phản hồi liên quan đến việc cung cấp và yêu cầu của họ |
| **Thời hạn dài**: Bạn cần theo đúng quy trình của dự án và phải hoàn thành một giai đoạn trước khi bắt đầu giai đoạn tiếp theo, nên bạn sẽ mất rất nhiều thời gian để đạt được kết quả cuối cùng | **Giảm thời gian phát triển:** vì trong mô hình RAD các thành phần hoặc chức năng được phát triển song song như thể chúng là các dự án nhỏ nên thời gian hoàn thành sẽ nhanh hơn nhưng đồng thời rủi ro cũng cao hơn |
| **Xác định dữ liệu ngay từ đầu**: Giai đoạn đầu của phương pháp Waterfall có lẽ là thời điểm quan trọng và khó khăn nhất. Nó đòi hỏi phải có sự giao tiếp với khách hàng và các bên liên quan, phải hiểu chính xác kết quả cuối cùng mà họ muốn đạt được là gì | Khuyến khích khách hàng đưa ra phản hồi, **không cần nắm bắt dữ liệu ngay từ đầu** |