

Vòng đời của phần mềm

#OOAD

1. Hệ thống

Một hệ thống sẽ bao gồm phần cứng và phần mềm

- phần cứng: hệ thống mạng, tài liệu
- phần mềm: phần mềm nền tảng (hệ điều hành), phần mềm trung gian và ứng dụng doanh nghiệp

2. Vòng đời hệ thống phần mềm

- Vòng đời của phần mềm thời kỳ được tính từ lúc nó được tạo ra đến khi chết đi (từ lúc hình thành để đáp ứng yêu cầu, vận hành, bảo dưỡng cho đến khi bị loại bỏ)
- Vòng đời của phần mềm được phân chia thành các pha chính: **phân tích, thiết kế, chế tạo, kiểm thử, bảo trì**
- Mọi sản phẩm phần mềm đều có vòng đời
- Vòng đời thường khá dài [#tìmvidu](#)
- Vòng đời thường rút ngắn do tiến bộ công nghệ

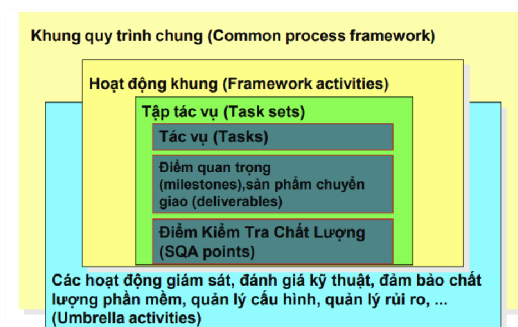
2.1 Các pha trong vòng đời phần mềm

- **Yêu cầu** - xác định nhu cầu của khách hàng và các ràng buộc của sản phẩm
- **Thiết kế** - xác định cấu trúc/tổ chức của hệ thống
- **Mã hóa** - code phần mềm
- **Kiểm thử** - vận hành thử để tìm và loại bỏ các khuyết điểm
- **Bảo trì** - sửa chữa và nâng cấp hệ thống sau khi khách hàng triển khai

2.2 Các mô hình vòng đời phần mềm

- Quá trình là một tập hợp các hoạt động liên kế với nhau, với đầu vào và đầu ra rõ ràng để hoàn thành một số nhiệm vụ đề ra.
- Mô hình vòng đời là một mô tả về quá trình thực hiện một sản phẩm trong một phần hoặc toàn bộ vòng đời của nó:
 - Các mô hình vòng đời tập trung vào các pha chính của chu kỳ và mối quan hệ giữa chúng với nhau
 - Các nghiên cứu gần đây về quy trình phần mềm xem xét chi tiết nhiều khía cạnh của việc phát triển và bảo trì
 - Mô hình vòng đời là một mô tả quy trình phần mềm

3. Quy trình phát triển phần mềm



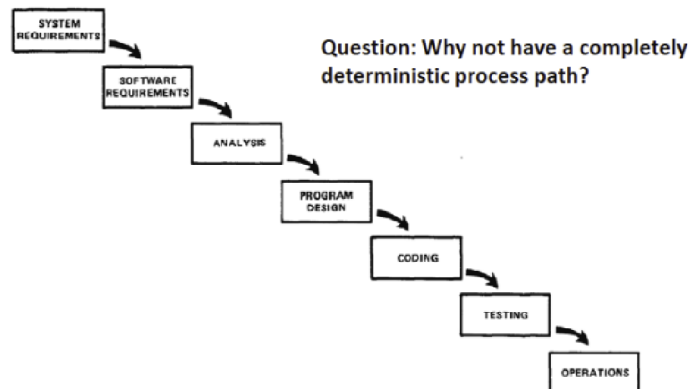
4. Các mô hình quy trình phần mềm

4.1 Mô hình thác nước

- Mô hình thác nước là mô hình vòng đời lâu đời nhất được đề xuất bởi **Winston Royce** vào năm 1970.
- Mô hình này được gọi là mô hình thác nước bởi vì mô tả chuỗi các pha - hoạt động qua các giai đoạn trong vòng đời "xuống dốc"
 - phân tích → yêu cầu → đặc tả → thiết kế → cài đặt → kiểm thử → bảo trì
- Có nhiều hơn phiên bản của mô hình thác nước
 - các giai đoạn / hoạt động có thể được cấu trúc theo các mức độ chi tiết khác nhau
 - phản hồi có thể linh hoạt hoặc ít hơn

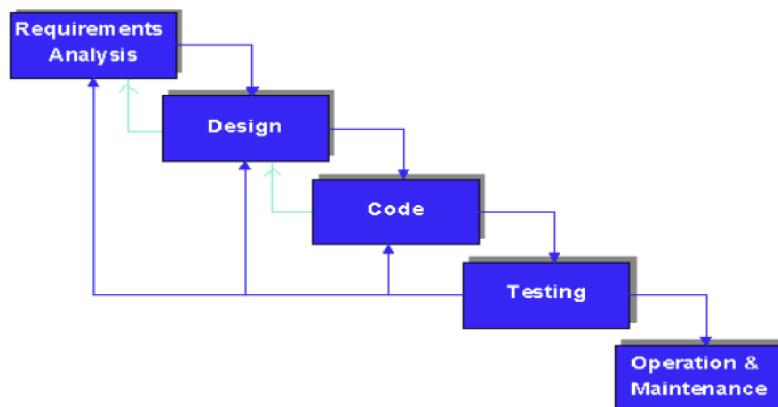
4.1.1 Mô hình thác nước nghiêm ngặt

Life Cycle Ideal - (Strict) Waterfall With No Feedback



4.1.2 Mô hình thác nước không nghiêm ngặt

Mặc dù mô hình thác nước nhấn mạnh một chuỗi sự kiện tuyến tính của các pha nhưng trong thực tế, luôn có một lượng lớn sự lặp lại của các pha trước đó.



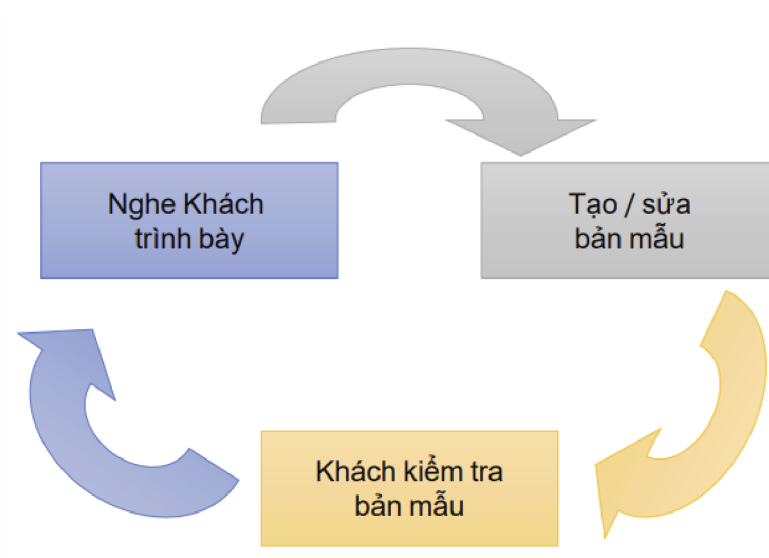
Điểm mạnh:

- Đề cao sự tuyến tính, hoàn thành một giai đoạn trước khi qua giai đoạn tiếp theo.
- Nhấn mạnh việc lập kế hoạch sớm, đầu vào của khách hàng và thiết kế hệ thống.
- Nhấn mạnh việc testing
- Cung cấp các chất lượng ở mỗi giai đoạn vòng đời

Điểm yếu

- phụ thuộc vào yêu cầu** được xác định sớm từ đầu
- phụ thuộc vào việc **tách các yêu cầu** khỏi thiết kế
- thiếu sự linh hoạt**, không khả thi trong một số trường hợp cần thay đổi
- nhấn mạnh vào **sản phẩm hơn là quy trình** #deohieu

4.2 Mô hình mẫu thử (Prototyping model)



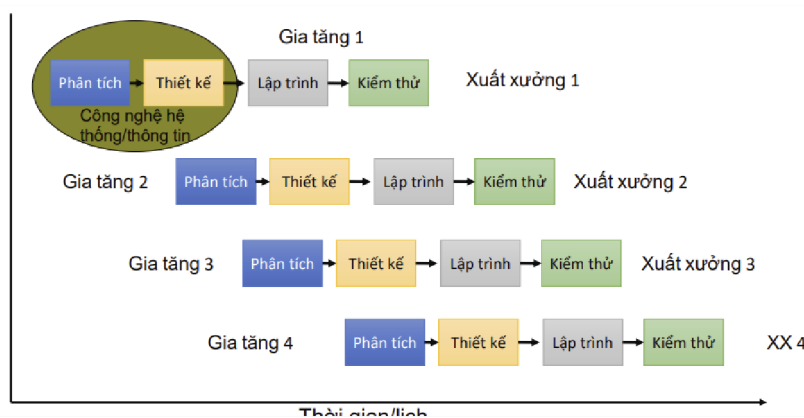
Mô hình được sử dụng khi **chỉ mới rõ mục đích chung chung** của phần mềm, không rõ yêu cầu đầu vào cũng như đầu ra.

Dùng để **thu thập yêu cầu** qua thiết kế nhanh.

Các kỹ thuật, công nghệ được sử dụng có thể chưa được tối ưu, mục đích chính là để thảo luận với khách hàng làm rõ mục tiêu, yêu cầu.

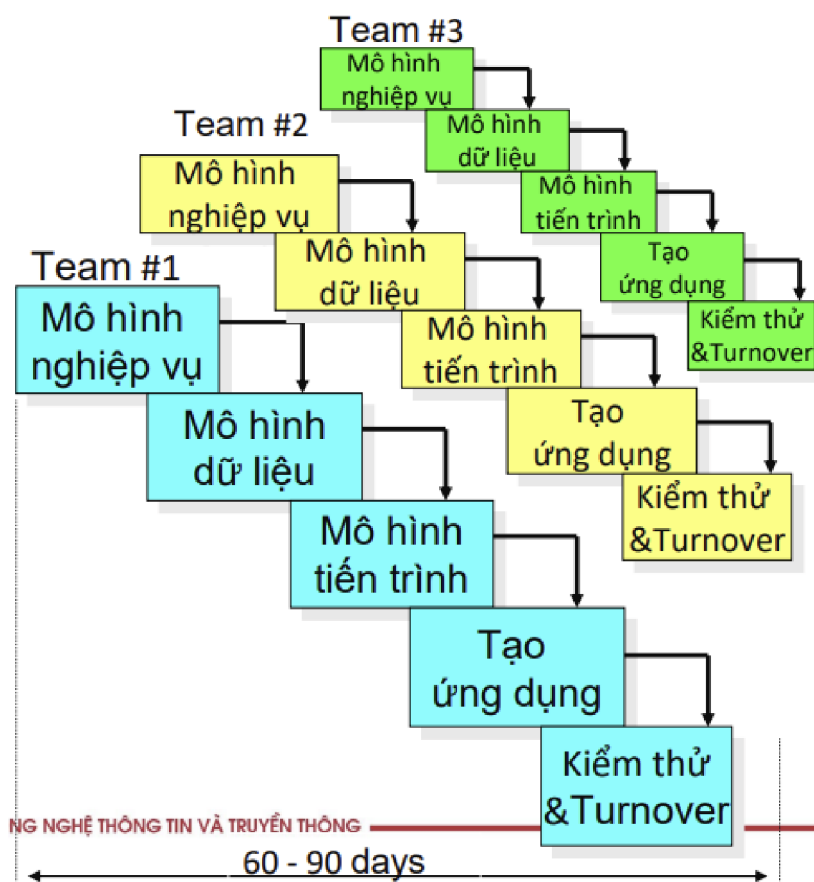
4.3 Mô hình gia tăng (the incremental model)

- Kết hợp giữa sự **tuyến tính** của thác nước với ý tưởng **lặp lại** của mô hình mẫu thử.
- Những yêu cầu cơ bản của hệ thống được ưu tiên phát triển
- Các chức năng, yêu cầu khác được phát triển theo thời gian
- Lặp lại quy trình để **tiến hóa** dần.

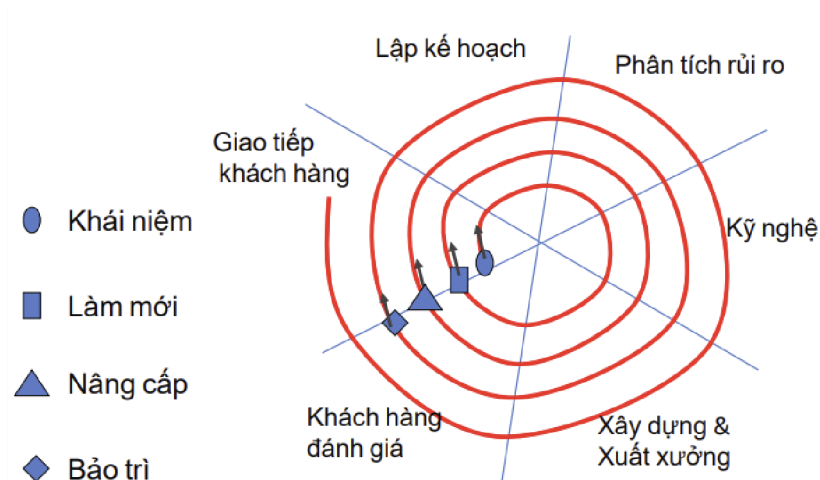


4.4 Mô hình phát triển ứng dụng nhanh (RAD)

- Là quy trình phát triển phần mềm gia tăng, tăng dần từng bước với mỗi chu trình rất ngắn (60-90 ngày).
- Xây dựng dựa trên hướng thành phần (Component-based construction) và khả năng tái sử dụng (reuse)
- Gồm một số nhóm, mỗi nhóm thực hiện RAD theo các pha
 - Mô hình nghiệp vụ
 - Mô hình dữ liệu
 - Mô hình xử lý
 - Tạo ứng dụng
 - Kiểm thử và đánh giá (Business, Data, Process, Appl. Generation, Test)



4.5 Mô hình xoắn ốc (spiral model)



- **Giao tiếp khách hàng:** đối thoại để tìm hiểu nhu cầu, ý kiến của khách hàng
- **Lập kế hoạch:** Xác lập tài nguyên, thời gian, công việc, etc
- **Phân tích rủi ro:** Xem xét mạo hiểm kỹ thuật và mạo hiểm quản lý
- **Kỹ nghệ:** Xây dựng biểu diễn ứng dụng
- **Xây dựng và xuất xưởng:** xây dựng, kiểm thử, cài đặt và cung cấp hỗ trợ người dùng (tư liệu, huấn luyện...)
- **Đánh giá của khách hàng:** Nhận các phản hồi của user về biểu diễn phần mềm trong giai đoạn kỹ nghệ và cài đặt.

Điểm mạnh:

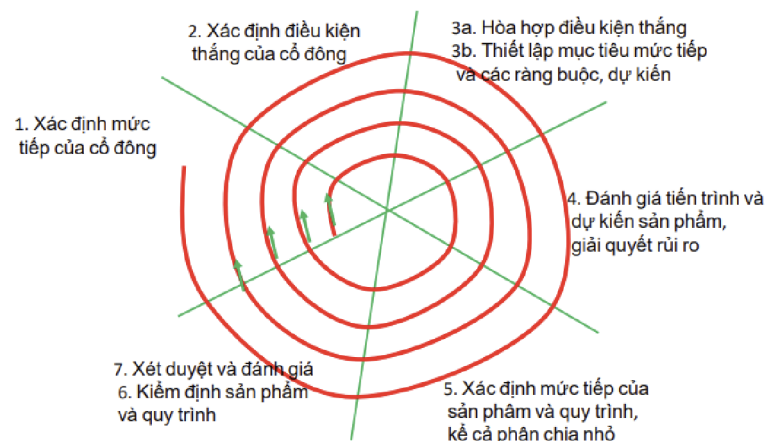
- Tốt cho các phần mềm **có quy mô lớn**
- Dễ **kiểm soát rủi ro**

Điểm yếu:

- Khó thuyết phục khách hàng về khả năng kiểm soát với mô hình
- Chưa được sử dụng rộng rãi

4.6 Mô hình xoắn ốc WINWIN

- Mục tiêu: **thỏa hiệp** giữa developer và customer theo mqh win-win
 - Customer có được phần mềm đáp ứng được nhu cầu
 - Developer có được thù lao phù hợp với công sức bản thân bỏ ra
- Các hoạt động chính trong model:
 - Xác định cổ đông (stakeholders)
 - Xác định điều kiện thắng cổ đông
 - Thỏa hiệp quyền lợi của các bên liên quan



4.7 Tổng kết các mô hình

- Thác nước:** mô hình tuyến tính
- Mẫu thử:** mô hình lặp đi lặp lại
- Gia tăng:** kết hợp giữa mô hình tuyến tính và lặp đi lặp lại
- Xoắn ốc:** kết hợp giữa mô hình tuyến tính và lặp đi lặp lại
- Phát triển nhanh:** mô hình lặp đi lặp lại

5. Tổng kết

- Vòng đời phần mềm tính từ khi sinh ra đến khi chết đi.
- Mô hình vòng đời là một mô tả về một quá trình thực hiện một sản phẩm phần mềm trong toàn bộ hoặc một phần vòng đời của nó
- Các mô hình: thác nước, mẫu thử, gia tăng, xoắn ốc, phát triển nhanh