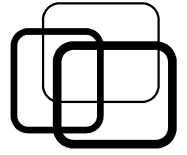
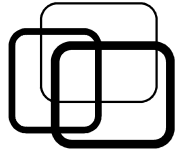


Quy trình phần mềm

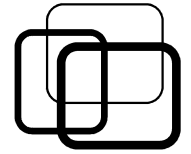
GV. Nguyễn Minh Huy



- Khái niệm cơ bản.
- Các loại quy trình phần mềm.



- **Khái niệm cơ bản.**
- **Các loại quy trình phần mềm.**

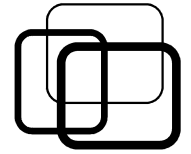


- Đặc điểm của làm việc không quy trình:
 - Không theo trình tự nhất định.
 - Không có bài bản định sẵn.
 - Cảm tính, lúc này lúc khác.
 - ➔ Khó dự đoán kết quả (predictable).
 - ➔ Khó kiểm soát chất lượng (controllable).
 - ➔ Khó lặp lại thành công (repeatable).
 - ➔ Đầy may rủi!!



© Can Stock Photo - csp3532811

Khái niệm cơ bản

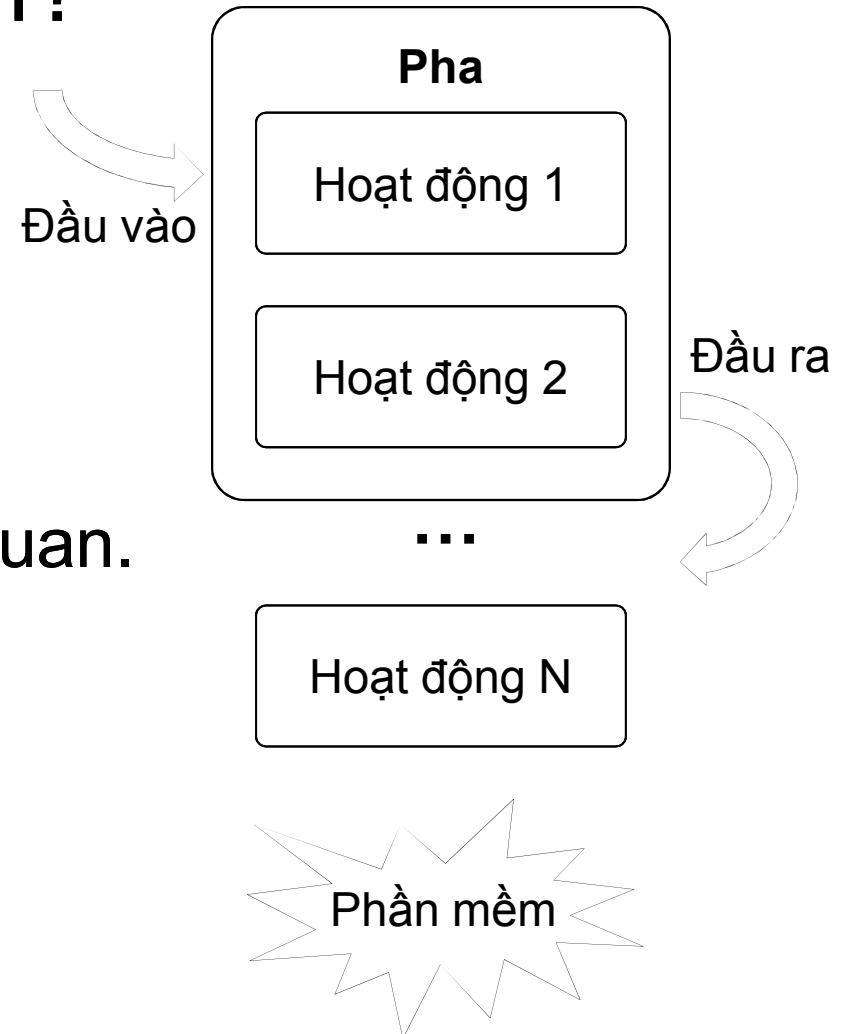


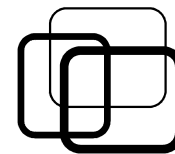
■ Quy trình phần mềm là gì?

- Chuỗi hoạt động.
- Theo trình tự nhất định.
- Sản xuất phần mềm.

■ Pha của quy trình:

- Chuỗi con hoạt động liên quan.
→ Công đoạn.
- Mỗi pha định nghĩa:
 - WHAT: làm gì.
 - WHO: ai tham gia.
 - INPUT: đầu vào.
 - OUTPUT: đầu ra.





■ Cách cách mô tả quy trình:

■ Workflow:

- Theo luồng công việc.
- Trình tự các bước quy trình.
- Cách thông dụng thể hiện quy trình.

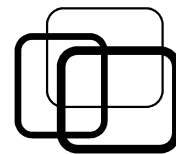
■ Dataflow:

- Theo luồng dữ liệu.
- Xoay quanh một đơn vị dữ liệu.

■ Role/Action:

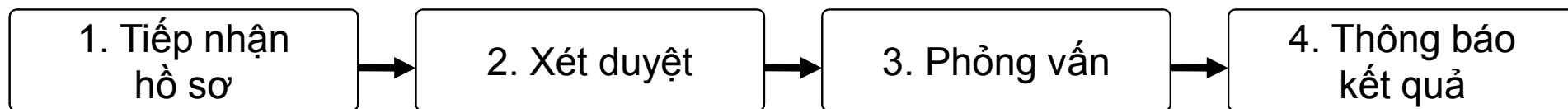
- Theo vai diễn.
- Xoay quanh một vai diễn.

Khái niệm cơ bản

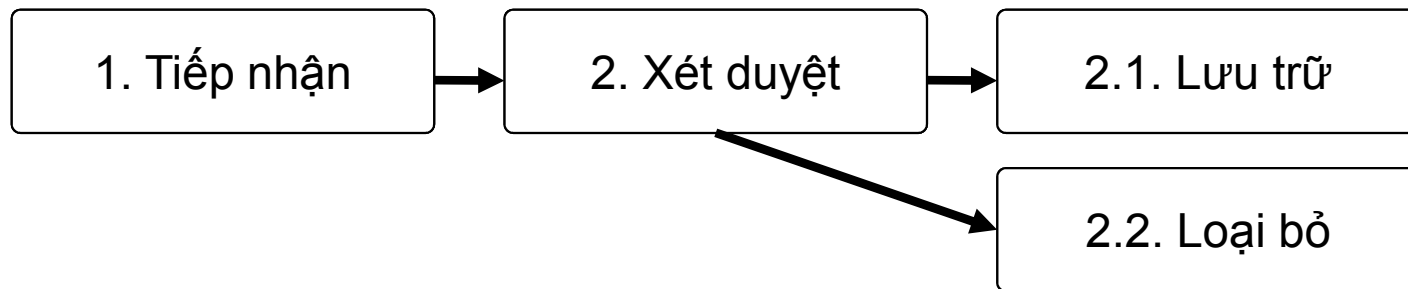


■ Quy trình phỏng vấn xin việc:

■ Workflow:



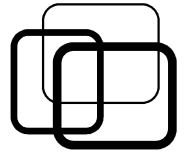
■ Dataflow: đơn vị dữ liệu “đơn xin việc”



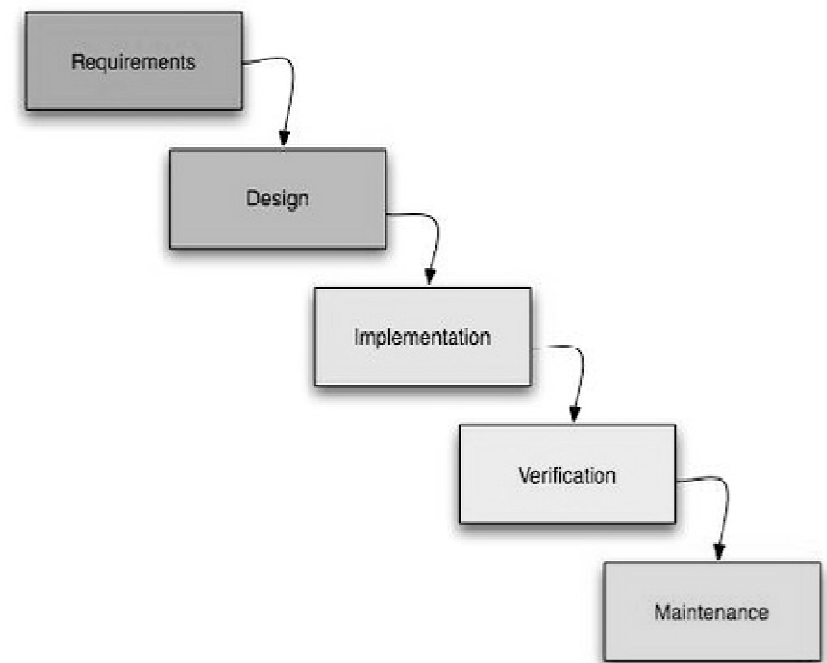
■ Role/Action: vai diễn “người phỏng vấn”,

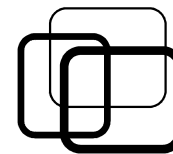


Khái niệm cơ bản

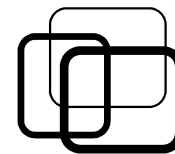


- Các pha chính yếu trong quy trình:
 - Phân tích yêu cầu.
 - Thiết kế phần mềm.
 - Cài đặt phần mềm.
 - Kiểm nghiệm phần mềm.
 - Bảo trì phần mềm.

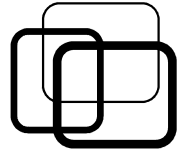




- Phân tích yêu cầu:
 - Trả lời câu hỏi **WHAT**.
 - Xác định những gì cần làm.
 - Vai trò tham gia: BA (**B**usiness **A**nalyst).
 - Các hoạt động:
 - Thu thập yêu cầu khách hàng.
 - Đặc tả yêu cầu.
 - Kiểm nghiệm yêu cầu.
 - Mô hình hóa phần mềm.

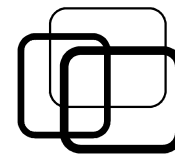


- Thiết kế phần mềm:
 - Trả lời câu hỏi **HOW**.
 - Xác định làm như thế nào.
 - Vai trò tham gia: kiến trúc sư hệ thống (architect).
 - Các hoạt động:
 - Thiết kế tổng thể (high-level design).
 - Thiết kế kiến trúc.
 - Thiết kế chi tiết (low-level design).
 - Thiết kế dữ liệu.
 - Thiết kế giao diện.
 - Thiết kế xử lý.



■ Cài đặt phần mềm:

- Hiện thực hóa bản thiết kế.
- Vai trò tham gia: lập trình viên (developer).
- Các hoạt động:
 - Cài đặt mã nguồn.
 - Cài đặt cơ sở dữ liệu.
 - Cấu hình hệ thống.



■ Kiểm nghiệm phần mềm:

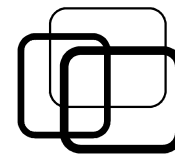
■ Phát hiện lỗi:

- So với phân tích, thiết kế.
- So với cách hiểu khách hàng.

■ Vai trò tham gia: kiểm tra viên (tester), khách hàng, lập trình viên.

■ Các hoạt động:

- Kiểm thử đơn vị (lập trình viên).
- Thanh tra mã nguồn.
- Kiểm thử hệ thống (tester).
- Nghiệm thu (khách hàng).



■ Bảo trì:

- Pha “hậu mãi”.

- Vận hành, chỉnh sửa, nâng cấp.

- Vai trò tham gia:

 - Kỹ thuật viên (technical support).

 - Lập trình viên.

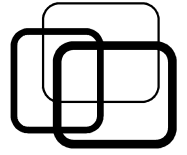
- Các hoạt động:

 - Thiết lập nền tảng hoạt động: máy móc, mạng, ứng dụng, ...

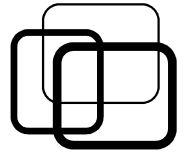
 - Lắp đặt và cấu hình phần mềm.

 - Thông báo và sửa lỗi phần mềm.

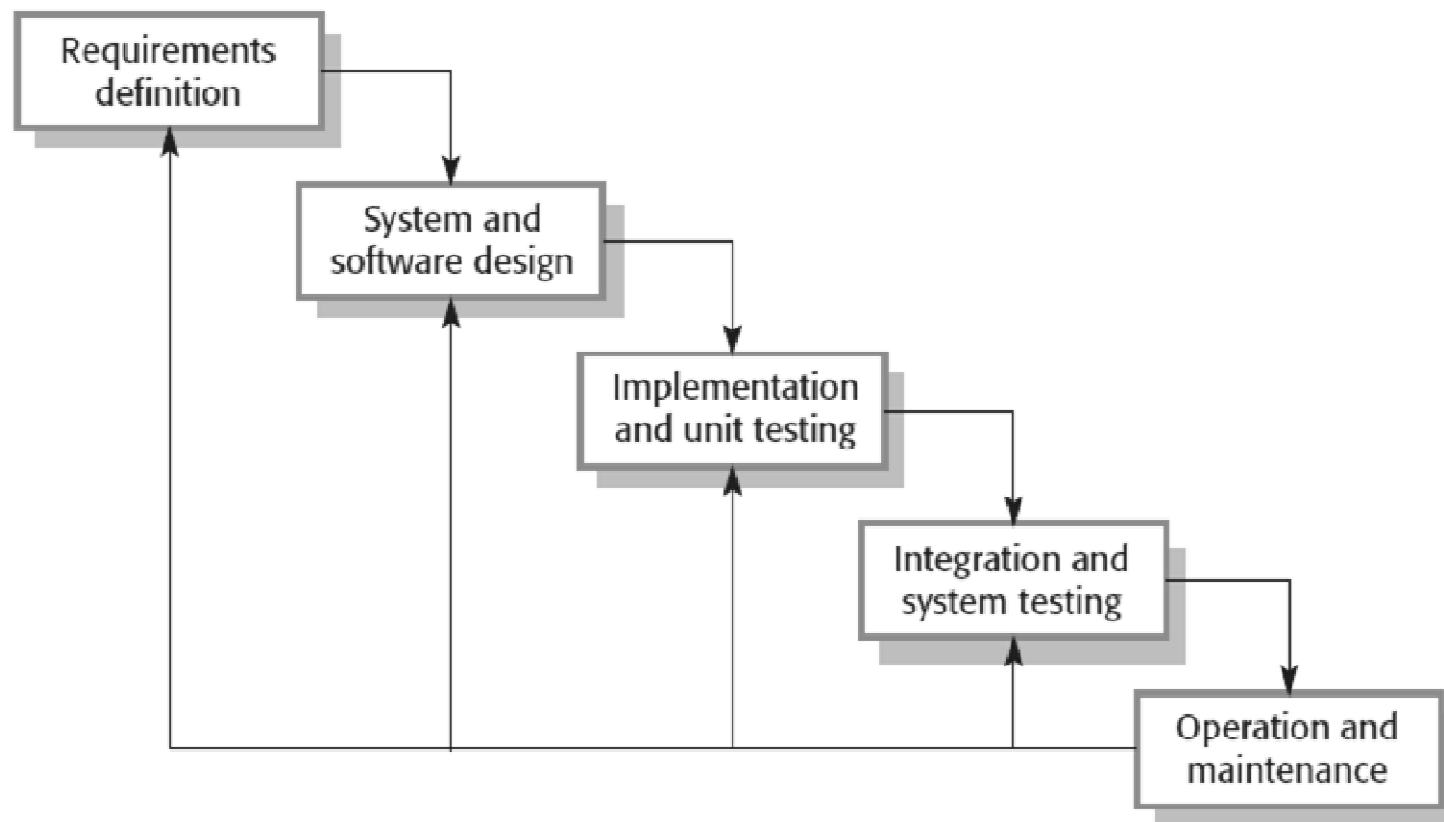
 - Nâng cấp phần mềm.

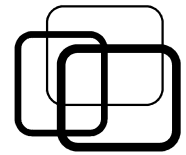


- Khái niệm cơ bản.
- **Các loại quy trình.**



- Mô hình thác nước – Waterfall Model:
 - Winston Royce đề cập, 1970.





■ Mô hình thác nước – Waterfall Model:

■ Đặc trưng:

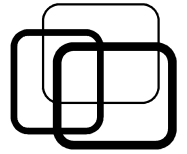
- Các pha diễn ra TUẦN TỰ và ĐỘC LẬP.
- Tách rời giữa phác thảo và hiện thực.
- Chú trọng kiểm nghiệm sau khi làm.

■ Ưu điểm:

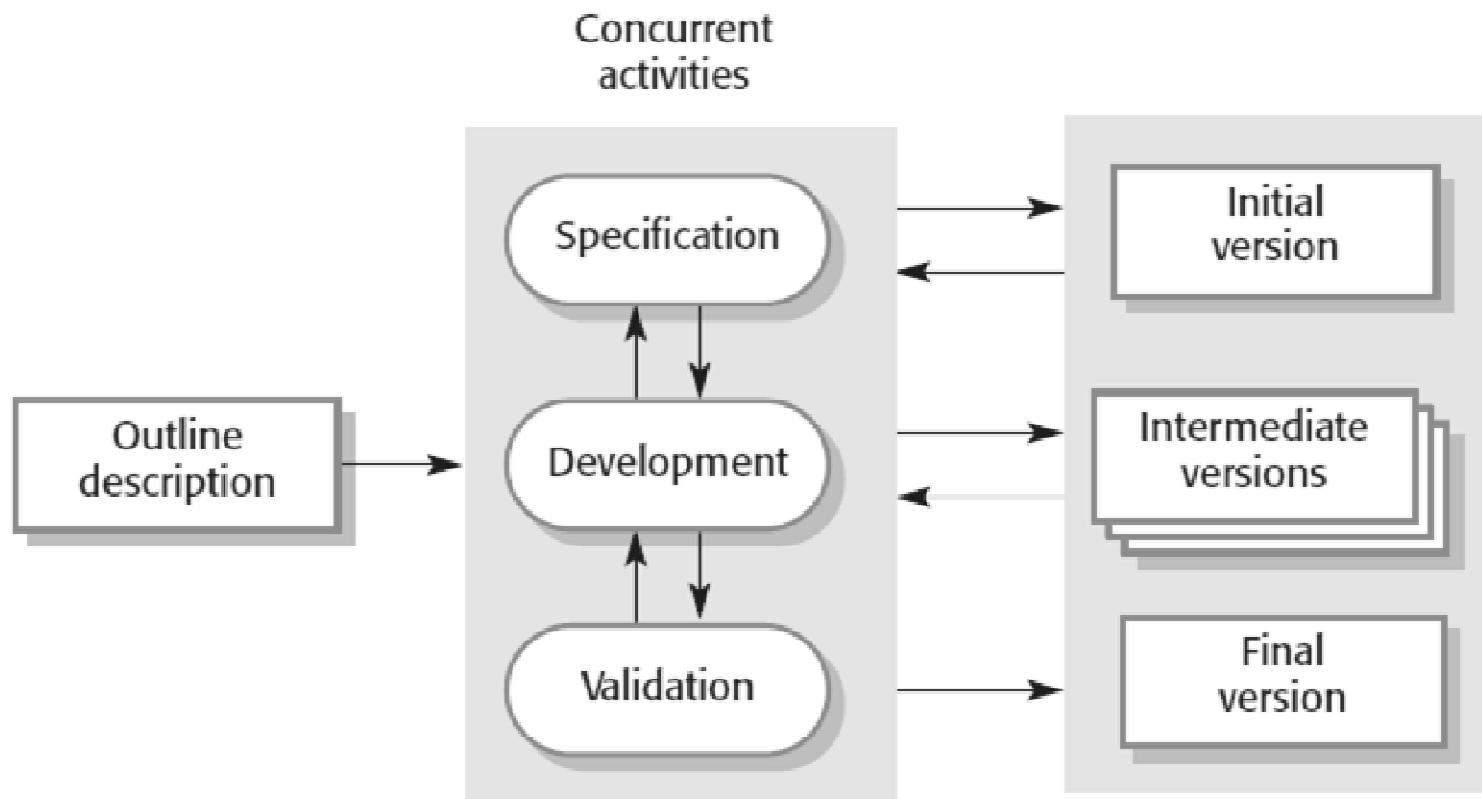
- Thực hiện có hệ thống và bài bản.
- Tiên liệu chặt chẽ trước khi làm.
- Có thể dự đoán kết quả (predictable-process).

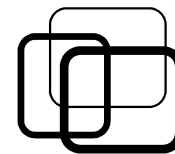
■ Khuyết điểm:

- Khó khăn khi có thay đổi xảy ra.
 - Chỉ thích hợp với dự án có yêu cầu rõ ràng và ổn định.
- ➔ Cải tiến cho phép quay lui.



- Mô hình tiến hóa – Evolutionary Model:
 - Cải tiến phương pháp Code & Fix.





■ Mô hình tiến hóa – Evolutionary Model:

■ Đặc trưng:

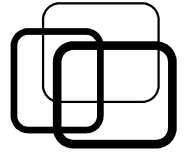
- Không đặt nặng tiên liệu trước.
- Cải tiến dần qua các phiên bản.
- Nhận phản hồi thường xuyên.
- Phân tích, thiết kế, và kiểm tra sơ lược.

■ Ưu điểm:

- Yêu cầu không cần rõ ràng và ổn định.
- Thích ứng tốt với thay đổi (adaptive-process).

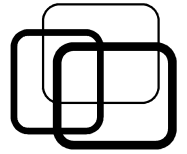
■ Khuyết điểm:

- Thiếu cách tiếp cận toàn diện.
- Dễ dẫn đến làm việc chấp vá.
- Khó dự đoán kết quả.

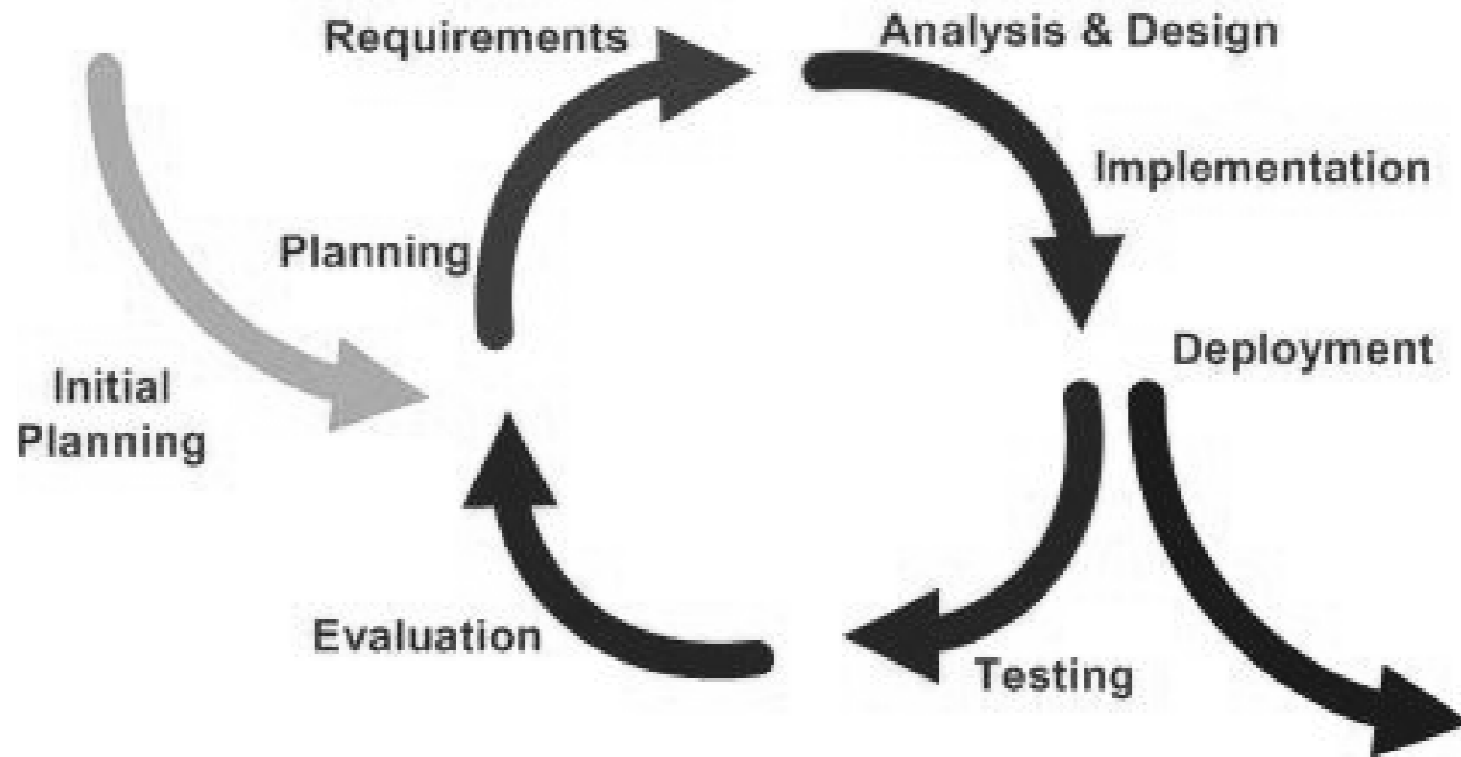


- Hiện thực hóa mô hình tiến hóa:
 - Phát triển qua nhiều vòng lặp ngắn.
 - Mỗi vòng lặp là một quy trình con.
 - ➔ Có sản phẩm THẬT sau mỗi vòng lặp.
 - Các quy trình được sử dụng:
 - Quy trình Waterfall lặp.
 - Quy trình xoắn ốc.
 - Quy trình RUP.
 - Phương pháp Agile.

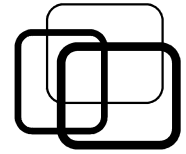
Các loại quy trình



■ Waterfall lặp:

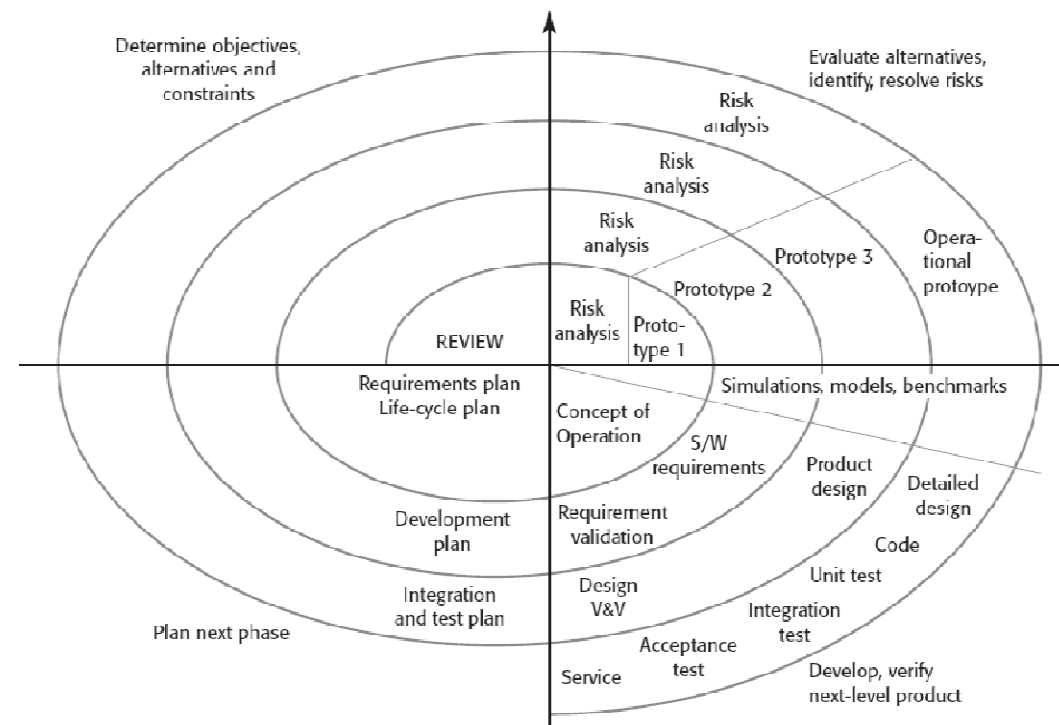


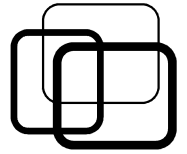
Các loại quy trình



■ Quy trình xoắn ốc (Spiral Process):

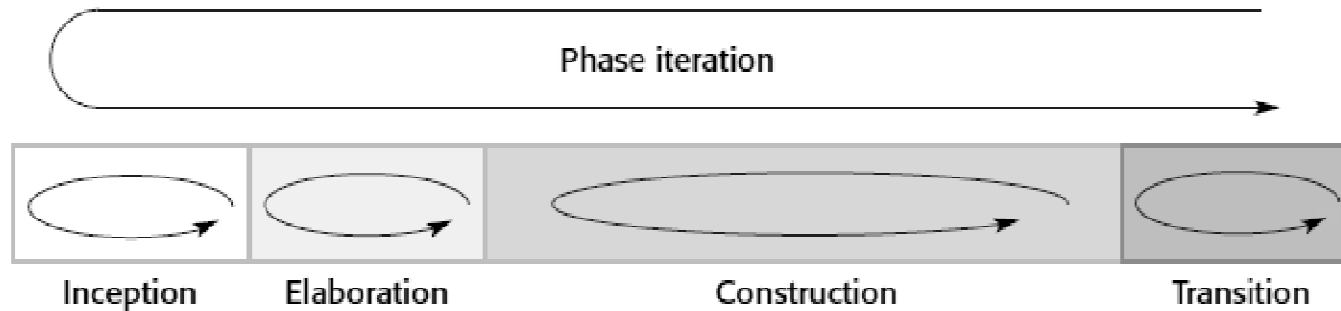
- Barry Boehm đề xuất, 1986.
- Mỗi xoắn ốc là một pha.
- 4 hoạt động trong một pha:
 - Xác định mục tiêu.
 - Định lượng rủi ro.
 - Thực hiện.
 - Kế hoạch pha sau.





■ Quy trình RUP (**R**ational **U**nified **P**rocess):

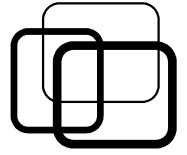
- Do IBM-Rational đề xuất (2003).



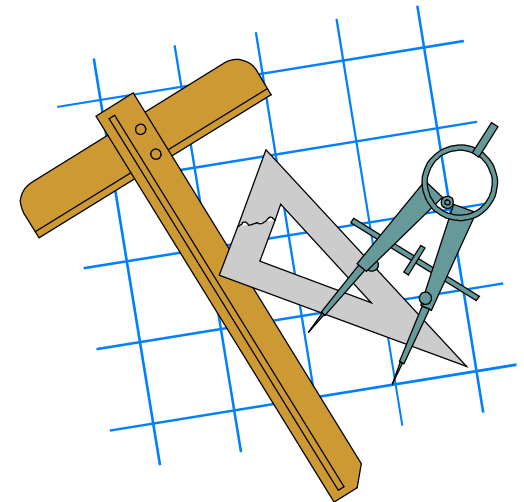
■ Gồm 4 pha:

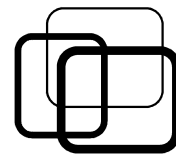
- Inception: khảo sát.
- Elaboration: phân tích & thiết kế.
- Construction: cài đặt & kiểm nghiệm.
- Transition: bảo trì & nâng cấp.

■ Cách lặp: lặp mỗi pha, lặp cả quy trình.



- Bài tập mô phỏng quy trình:
 - Nhóm của bạn được giao thực hiện một “DỰ ÁN”.
 - 1 – 2 thành viên trong nhóm đóng một VAI DIỄN.
 - Mỗi vai diễn thực công việc ĐỘC LẬP.





■ Các vai diễn:

■ Customer:

- Cung cấp yêu cầu qua trả lời phỏng vấn.

■ Analyst:

- Thu thập yêu cầu: thông qua phỏng vấn (3 lần).
- Phân tích yêu cầu: phân nhóm, liệt kê, giải thích.

■ Architect:

- Phác thảo sản phẩm: lập khung sườn, dàn ý.
- Gợi ý thực hiện: diễn giải cách thực hiện.

■ Developer:

- Hiện thực hóa sản phẩm: phát triển theo dàn ý.

■ QC Person:

- Kiểm tra sản phẩm: lập bảng đánh dấu đạt/chưa đạt.

