



CÔNG NGHỆ PHẦN MỀM

BÀI 2: Khái niệm – Quy trình – Mô hình phát triển

ThS. Cao Minh Thành, 2023





2.1 Công nghệ phần mềm

2.1.1 Khái niệm

ĐN1: CNPM là sự nghiên cứu những nguyên tắc công nghệ đúng đắn nhằm tạo ra phần mềm một cách kinh tế, tin cậy và làm việc hiệu quả trên các máy thực.



2.1 Công nghệ phần mềm

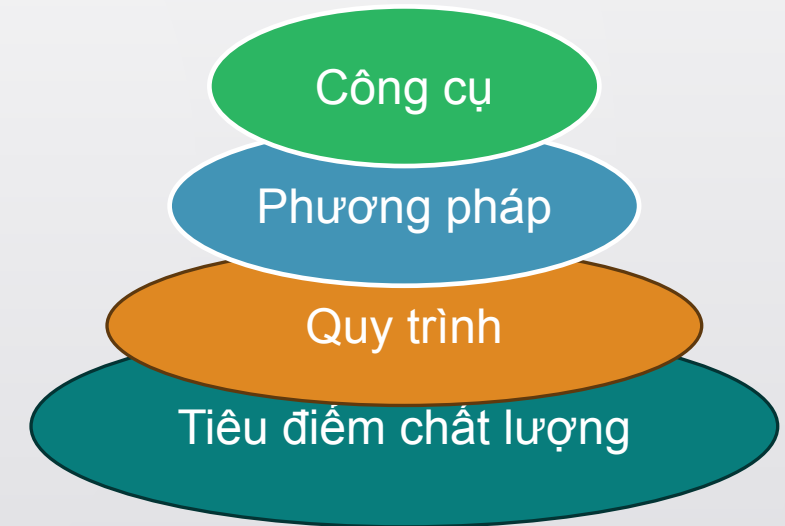
2.1.1 Khái niệm

ĐN2: CNPM là lĩnh vực khoa học về các phương pháp luận, kỹ thuật và công cụ tích hợp trong quy trình sản xuất và vận hành phần mềm nhằm tạo ra phần mềm với những chất lượng mong muốn.

2.1 Công nghệ phần mềm

2.1.1 Khái niệm

- Tầng quy trình (nền tảng): định nghĩa khung làm việc cho một tập quy trình chủ yếu (KPA – Key Process Area), tạo nên các cơ sở để quản lý các dự án phần mềm và thiết lập các ngưỡng để áp dụng các phương pháp kỹ thuật: mô hình, dữ liệu, báo biểu ...
- Phương pháp: kỹ thuật để chế tác phần mềm: phân tích yêu cầu người dùng, thiết kế hệ thống, lập trình, kiểm thử, bảo trì.
- Công cụ: cung cấp các hỗ trợ tự động, bán tự động cho quy trình và phương pháp phần mềm.





2.1 Công nghệ phần mềm

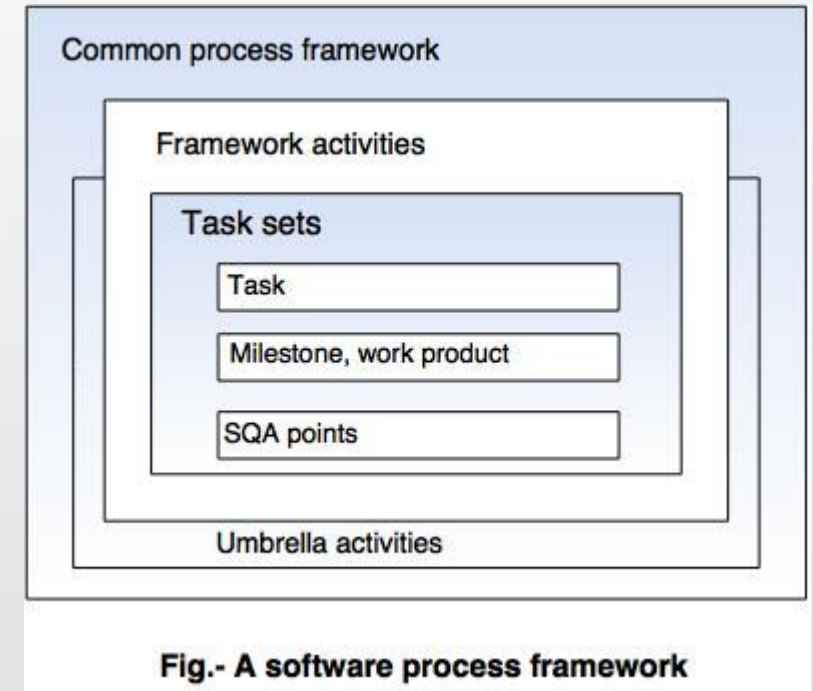
2.1.2 Vòng đời của phần mềm

Vòng đời phần mềm: là thời kì tính từ khi phần mềm được sinh ra cho đến khi chết đi (từ lúc hình thành yêu cầu, vận hành, bảo dưỡng cho đến khi loại bỏ).

Quy trình phần mềm gồm các pha chính: phân tích, thiết kế, chế thử, kiểm thử, bảo trì.

2.2 Qui trình phần mềm

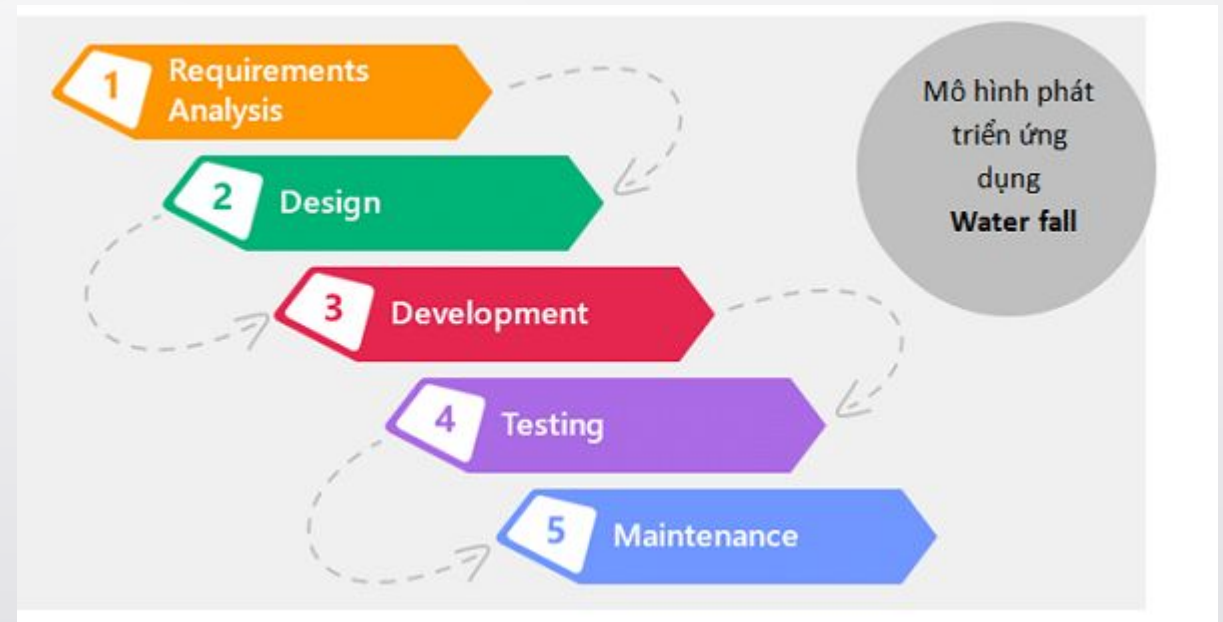
- Khung qui trình chung bao gồm một tập các hoạt động khung áp dụng trong mọi dự án phần mềm với kích cỡ và độ phức tạp khác nhau.
- Task sets: tập tác vụ gồm các nhiệm vụ, kết quả, điểm kiểm tra chất lượng.
- Các hoạt động hỗ trợ (umbrella) bao gồm các nhiệm vụ như đảm bảo chất lượng phần mềm, quản lý cấu hình và các biện pháp, bao trùm lên mô hình quá trình.



2.3 Các mô hình phát triển phần mềm

2.3.1 Mô hình tuyến tính (tuần tự)

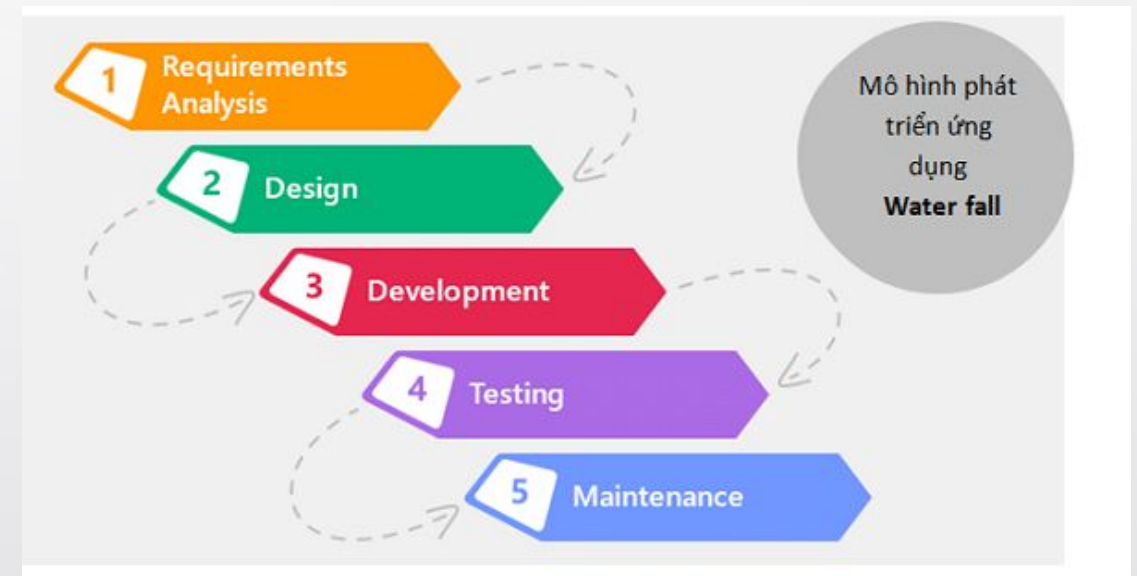
- Các pha được thực hiện một cách kế tiếp nhau, hết pha này đến pha khác (mô hình thác nước).



2.3 Các mô hình phát triển phần mềm

2.3.1 Mô hình tuyến tính (tuần tự)

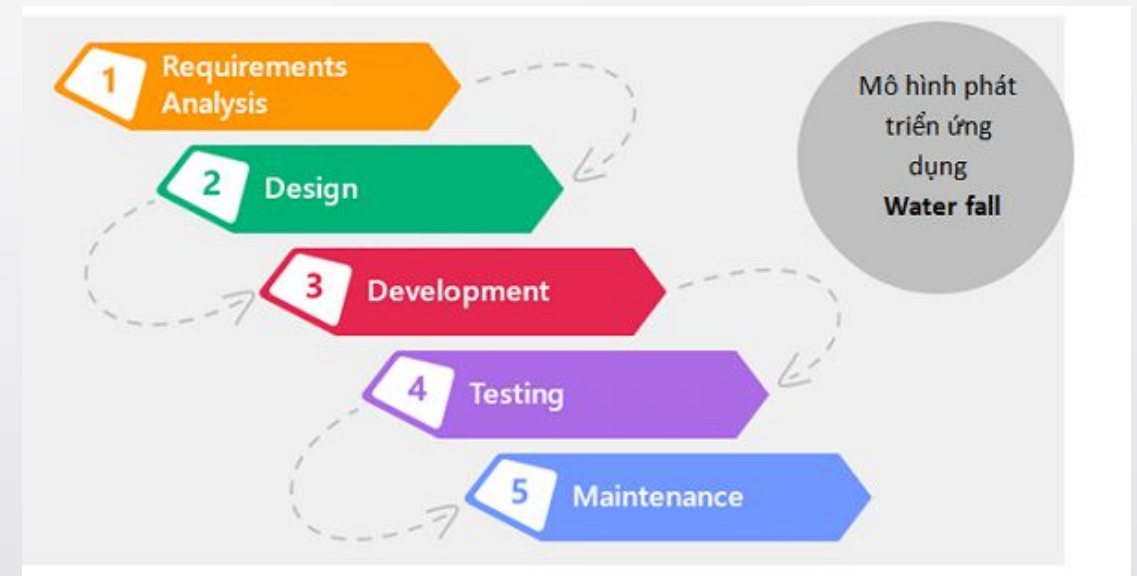
- Phân tích yêu cầu: tìm hiểu mục đích, phạm vi của phần mềm cần phát triển (chức năng, dữ liệu, hành vi, giao diện, ...)
- Thiết kế: thiết kế dữ liệu, kiến trúc phần mềm, giao diện, thuật toán.
- Tạo mã/ lập trình
- Kiểm thử: logic bên trong và bên ngoài, nhằm phát hiện ra lỗi
- Bảo trì: đáp ứng những thay đổi, nâng cấp phần mềm



2.3 Các mô hình phát triển phần mềm

2.3.1 Mô hình tuyến tính (tuần tự)

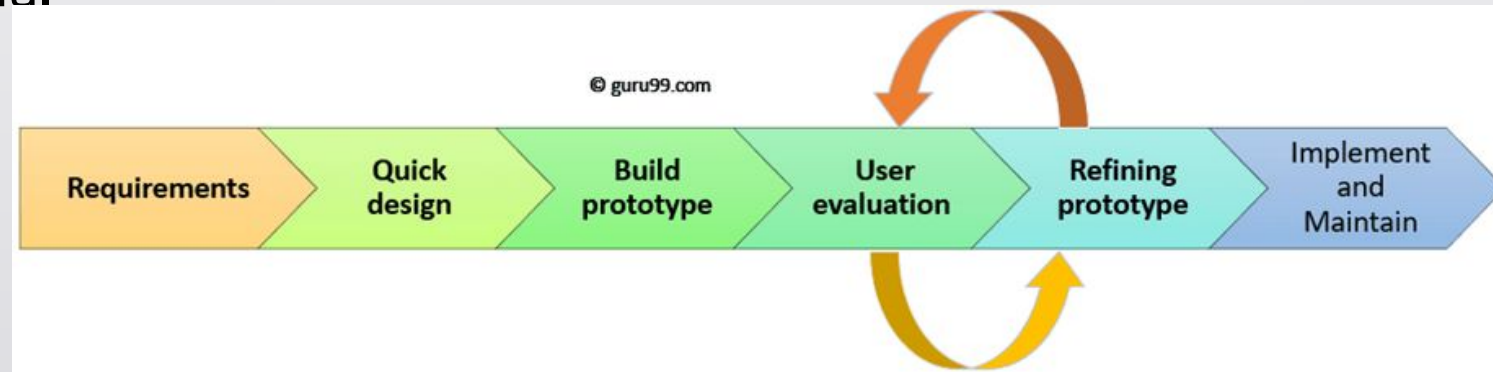
- Ưu điểm: dễ hiểu
- Khuyết điểm: các dự án phần mềm ít khi tuân theo dòng tuần tự, mà thường là lặp lại; khách hàng ít khi nào tuyên bố xong hết các yêu cầu; khách hàng phải chờ đợi thời gian nhất định mới có sản phẩm.



2.3 Các mô hình phát triển phần mềm

2.3.2 Mô hình chế thử (Prototyping Model)

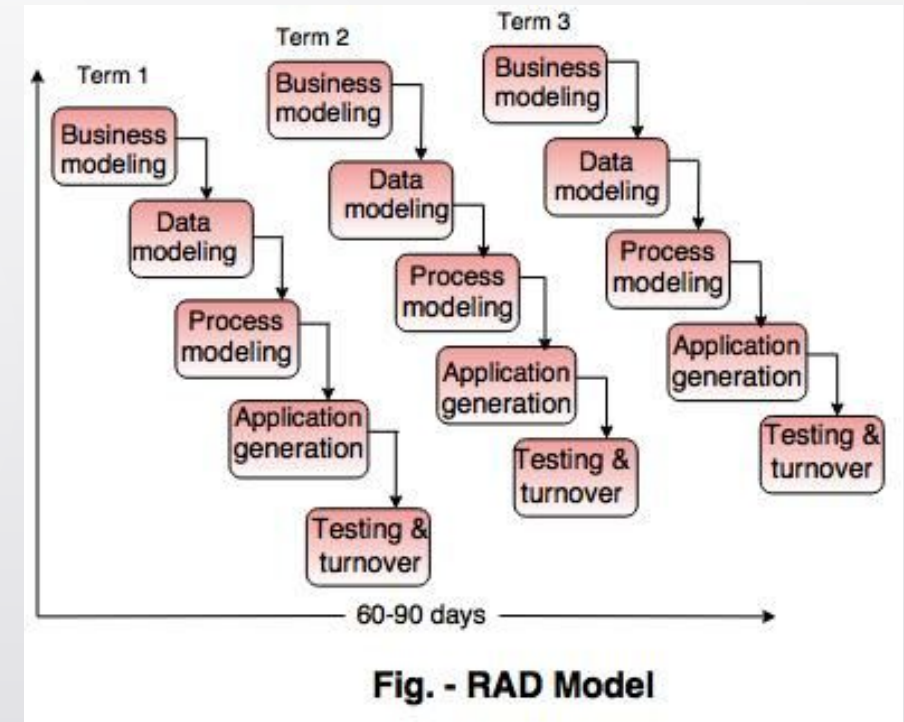
- DN chủ động tạo các kích thích, các nhu cầu cho khách hàng.
- Bản mẫu được thiết kế như một hệ sơ khai nhằm thu tóm yêu cầu người dùng.
- Các bản mẫu cần thiết kế nhanh, đơn giản, chưa cần tốt, miễn sao thu hút được ý kiến khách hàng.



2.3 Các mô hình phát triển phần mềm

2.3.3 Mô hình phát triển nhanh (Rapid Application Development - RAD)

- Cần sản phẩm trong thời gian ngắn (2-3 tháng).
- Quy trình phát triển phần mềm gia tăng, tang dần từng bước với mỗi chu trình phát triển ngắn 60-90 ngày.
- Gồm một số nhóm, mỗi nhóm làm 1 RAD gồm các pha: Mô hình nghiệp vụ, dữ liệu, xử lý, tạo ứng dụng, kiểm thử và đánh giá.

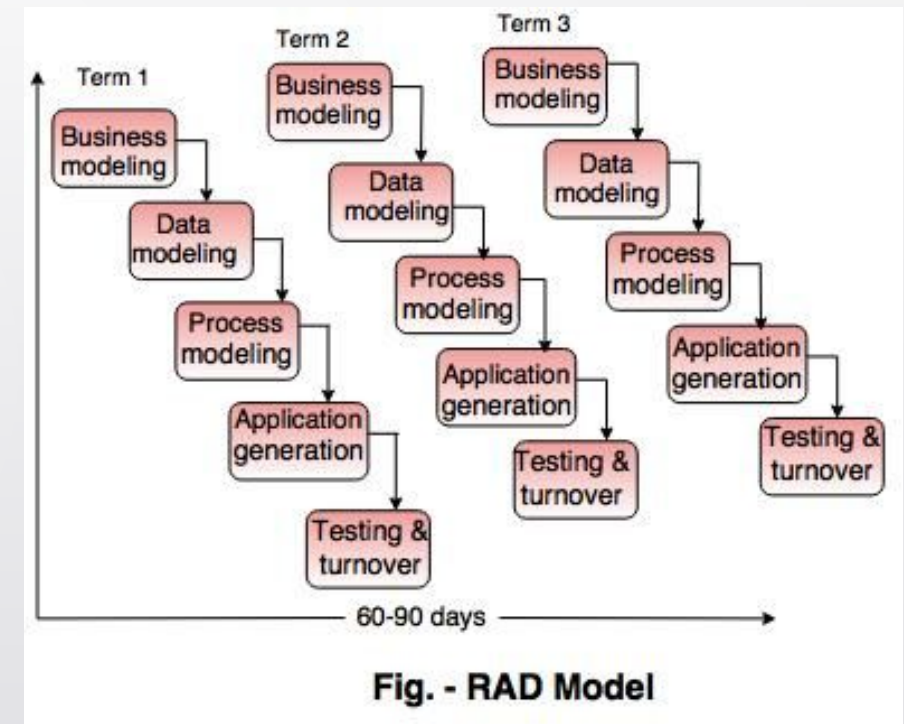



2.3 Các mô hình phát triển phần mềm

2.3.3 Mô hình phát triển nhanh (Rapid Application Development - RAD)

Ưu điểm: cung cấp sp nhanh.

Khuyết: cần đội ngũ đông đảo, giao kèo phải chặt chẽ, ứng dụng phải có tính module hóa.





2.3 Các mô hình phát triển phần mềm

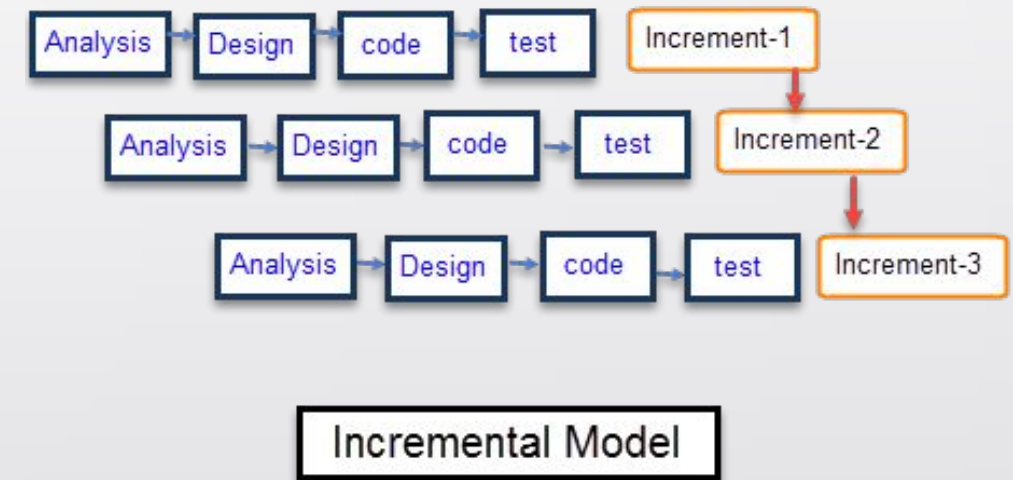
2.3.4 Mô hình tiến hóa (tiến hóa, xoắn ốc, phát triển đồng thời)

- Quy trình lặp
- Phát triển tăng dần, các phiên bản sau sẽ hoàn thiện hơn sau mỗi chu trình lặp.

2.3 Các mô hình phát triển phần mềm

2.3.4.1 Mô hình gia tăng (Incremental Model)

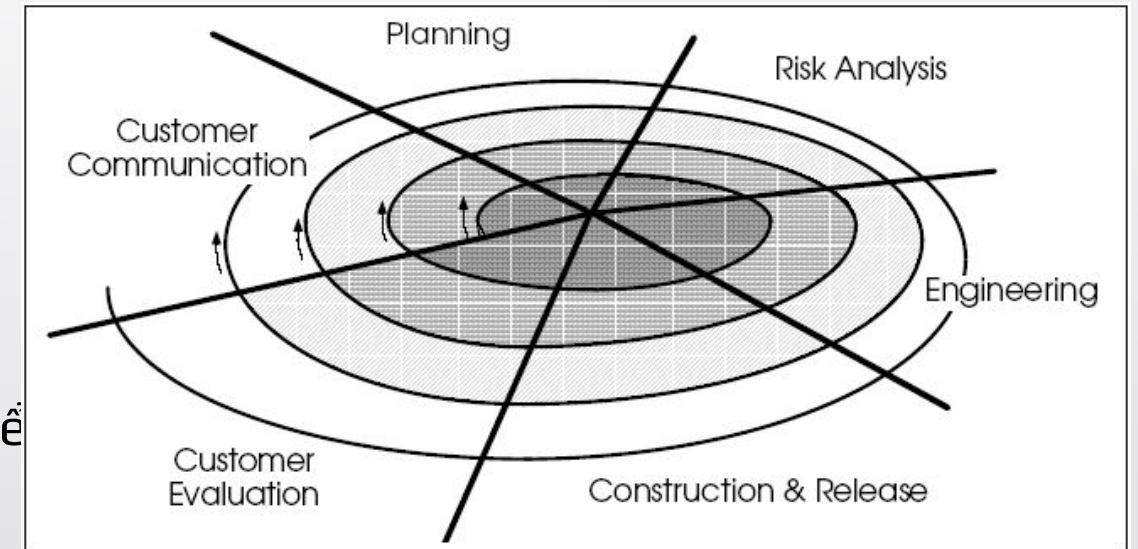
- Kết hợp mô hình tuyến tính và lặp của chế thử.
- Mỗi mạch tuyến tính của quy trình cung cấp 1 sản phẩm.
- Mạch đầu tiên cung cấp một số chức năng cơ bản nào đó, mạch cuối cùng cung cấp sp hoàn thiện.
- Khác với chế thử ở chỗ là sản phẩm dung được ở mỗi mạch, còn chế thử thì chưa quan tâm ngay đến sp.



2.3 Các mô hình phát triển phần mềm

2.3.4.2 Mô hình xoắn ốc (Spiral Model)

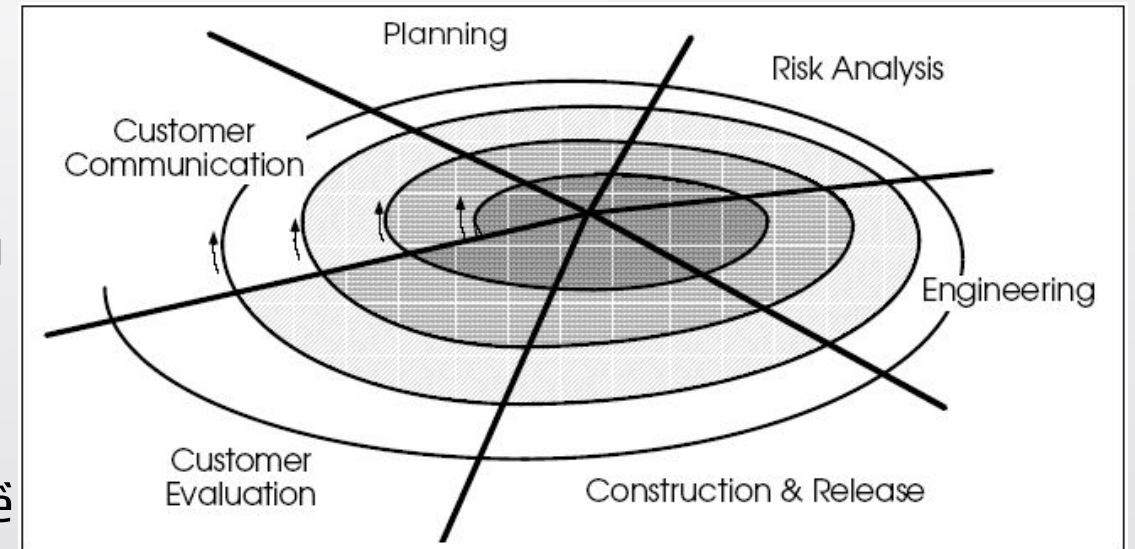
- Do Boehm đề xuất, kết hợp bản chất lặp của mô hình chế thử được giám sát với tính hệ thống của mô hình tuyến tính.
- PM được phát triển trong một loạt các phiên bản được gia tang, một số phiên bản đầu có thể là mô hình trên giấy hoặc mẫu thử.
- Các phiên bản sau trong các lần lặp với các công nghệ hoàn thiện hơn được tạo ra.



2.3 Các mô hình phát triển phần mềm

2.3.4.2 Mô hình xoắn ốc (Spiral Model)

- Ưu điểm: phần mềm được phát triển theo sự tiến hóa, khách hàng và nhà phát triển hiểu nhau tốt hơn, các rủi ro sẽ được giảm thiểu sau mỗi lần tiến hóa (nhờ áp dụng mô hình chế thử).
- Hạn chế: khả năng thuyết phục khách hàng về giảm thiểu tính rủi ro trong quá trình phát triển.





2.3 Các mô hình phát triển phần mềm

2.3.4.3 Mô hình phát triển đồng thời (Concurrent Development Model)



2.3 Các mô hình phát triển phần mềm

2.3.4.4 Mô hình lắp ráp hướng thành phần (Component assembly Model)



2.3 Các mô hình phát triển phần mềm

2.3.4.4 Mô hình lắp ráp hướng thành phần (Component assembly Model)