



- 4



Mục tiêu của bài học

- Hiểu được thế nào là vòng đời phần mềm
- Biết được quy trình phát triển phần mềm
- Các mô hình phát triển phần mềm phổ biến

- Một hệ thống, ví dụ hệ thống kinh doanh, bao gồm:
 - phần cứng, hệ thống mạng, phần mềm*, tài liệu
 - phần mềm* (software): bao gồm phần mềm nền tảng (như hệ điều hành), phần mềm trung gian (middle software), và ứng dụng doanh nghiệp (Business Application Software)



2. Vòng đời phần mềm

- Vòng đời phần mềm là thời kỳ tính từ khi phần mềm được **sinh (tạo) ra** cho đến khi **chết đi** (từ lúc hình thành đáp ứng yêu cầu, vận hành, bảo dưỡng cho đến khi loại bỏ không đâu dùng)
- Quy trình phần mềm (vòng đời phần mềm) được phân chia thành **các pha chính: phân tích, thiết kế, chế tạo, kiểm thử, bảo trì**. Biểu diễn các pha có thể khác nhau theo từng mô hình



Vòng đời phần mềm

- Mọi sản phẩm phần mềm đều có vòng đời.
- Vòng đời thường khá dài — một số sản phẩm phần mềm đã “tồn tại” được 30 năm.
- Vòng đời có thể được rút ngắn do tiến bộ công nghệ



Các pha trong vòng đời PM

- Một cách rõ ràng hoặc rõ ràng, tất cả các sản phẩm phần mềm đều trải qua ít nhất các giai đoạn sau:
 - **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 phần mềm
 - **Mã hóa** — viết phần mềm
 - **Kiểm thử** — vận hành hệ thống để tìm và loại bỏ các khiếm khuyết
 - **Bảo trì** — sửa chữa và nâng cao sản phẩm sau khi khách hàng triển khai



Các mô hình vòng đời phần mềm

- Quá trình là **một tập hợp các hoạt động**, với các đầu vào và đầu ra được xác định rõ ràng, để hoàn thành một số nhiệm vụ.
- 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 vòng đời có xu hướng **tập trung vào các pha chính** của chu kỳ và mối quan hệ của chúng với các pha khác.
 - Các nghiên cứu gần đây về quy trình phần mềm đã 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

Khung quy trình chung (Common process framework)

Hoạt động khung (Framework activities)

Tập tác vụ (Task sets)

Tác vụ (Tasks)

**Điểm quan trọng
(milestones), sản phẩm chuyển
giao (deliverables)**

**Điểm Kiểm Tra Chất Lượng
(SQA points)**

**Các hoạt động giám sát, đánh giá kỹ thuật, đảm bảo chất lượng phần mềm, quản lý cấu hình, quản lý rủi ro, ...
(Umbrella activities)**



4. Các Mô hình Quy trình Phần mềm

- **Mô hình Thác nước (Waterfall Model):**
 - Đặc trưng bởi sự phân tách các giai đoạn và tiến trình tuần tự qua chúng.
 - Còn được gọi là mô hình phát triển truyền thống hoặc mô hình tuần tự.
 - Đây là một phương pháp hợp lý khi các yêu cầu được xác định rõ ràng.
 - Nó thường bao gồm việc viết ít nhất một số đặc tả trước khi viết mã.
- **Mô hình Nguyên mẫu (Prototype Model):**
 - Bao gồm các giai đoạn như thu thập yêu cầu, thiết kế nhanh, xây dựng nguyên mẫu, đánh giá của khách hàng, tinh chỉnh yêu cầu, dẫn đến sản phẩm cuối cùng.
 - Nó hữu ích khi các yêu cầu của khách hàng không được xác định rõ ràng.
- **Các Mô hình Quy trình Tiến hóa (Evolutionary Process Models):**
 - Có tính chất lặp đi lặp lại.
 - Chúng có thể dễ dàng đáp ứng các yêu cầu sản phẩm thay đổi.
 - Nguyên mẫu tiến hóa (nơi nguyên mẫu phát triển thành hệ thống cuối cùng) thường được ưu tiên hơn so với nguyên mẫu bỏ đi vì nó cho phép tái sử dụng nguyên mẫu ban đầu và thường không tạo ra các sản phẩm bỏ đi.
- **Mô hình Xoắn ốc (Spiral Model):**
 - Cho phép kiểm soát rủi ro ở mỗi cấp độ tiến hóa.
 - Bao gồm việc đánh giá rủi ro phần mềm trong mỗi vòng lặp.
 - Nó được mô tả là hỗn loạn hơn các mô hình gia tăng.
 - Nó không nhất thiết phải kết thúc bằng việc phát hành sản phẩm phần mềm.
- **Phát triển Ứng dụng Nhanh (RAD - Rapid Application Development):**
 - Được mô tả là sự lắp ráp tốc độ cao của mô hình tuần tự tuyến tính.
- **Phát triển Dựa trên Thành phần (Component-Based Development):**
 - Bao gồm việc xây dựng phần mềm từ các thành phần được thiết kế sẵn.
 - Nó có thể hỗ trợ phát triển các thành phần có thể tái sử dụng.
 - Nó dựa trên các kỹ thuật hỗ trợ hướng đối tượng.
- **Agile/SCRUM:**
 - Một khung công việc được sử dụng trong phát triển linh hoạt.
 - So với các mô hình truyền thống, nó mang lại mức độ sáng tạo và linh hoạt cao, tạo điều kiện thuận lợi cho việc chuyển giao kiến thức và phản ứng nhanh với môi trường. Nó không đảm bảo chi phí dự án được quản lý hoặc giao hàng đúng thời hạn.



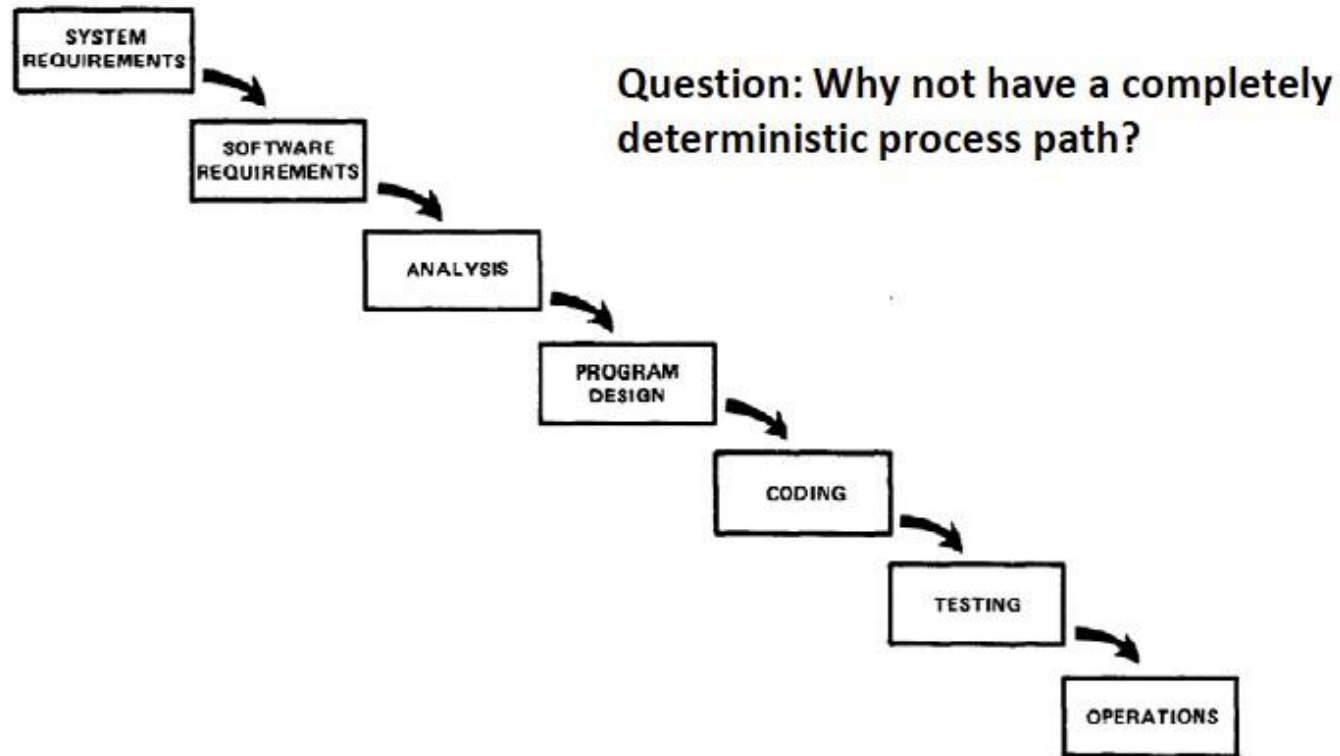


4.1. Mô hình thác nước

- 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 thác nước vì nó thường được vẽ với một **chuỗi các hoạt động qua các giai đoạn** của vòng đời “xuống dốc” từ trái sang phải:
 - **phân tích, yêu cầu, đặc tả, thiết kế, cài đặt, kiểm thử, bảo trì**
- Có nhiều 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 hơn hoặc ít hơn

4.1. Vòng đời lý tưởng - Thác nước (Ng nghiêm ngặt) không có phản hồi

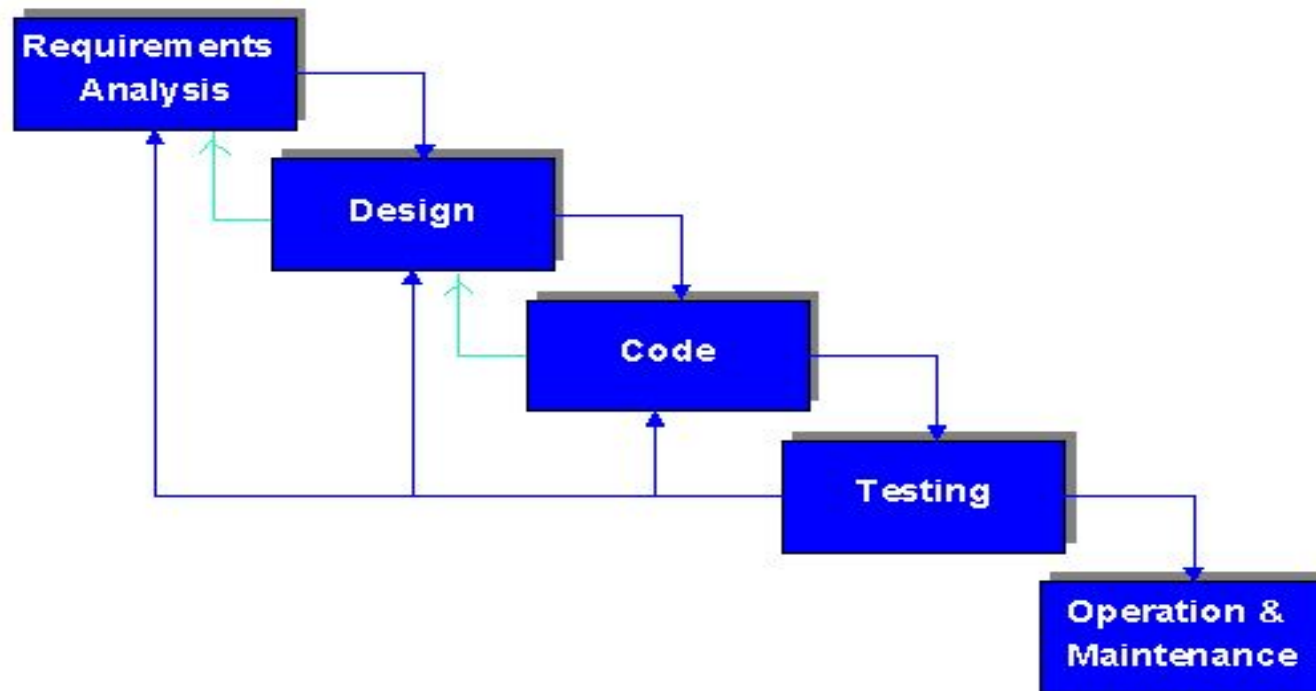
Life Cycle Ideal - (Strict) Waterfall With No Feedback





4.1. Mô hình thác nước (Non- stric)

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





4.1. Mô hình thác nước

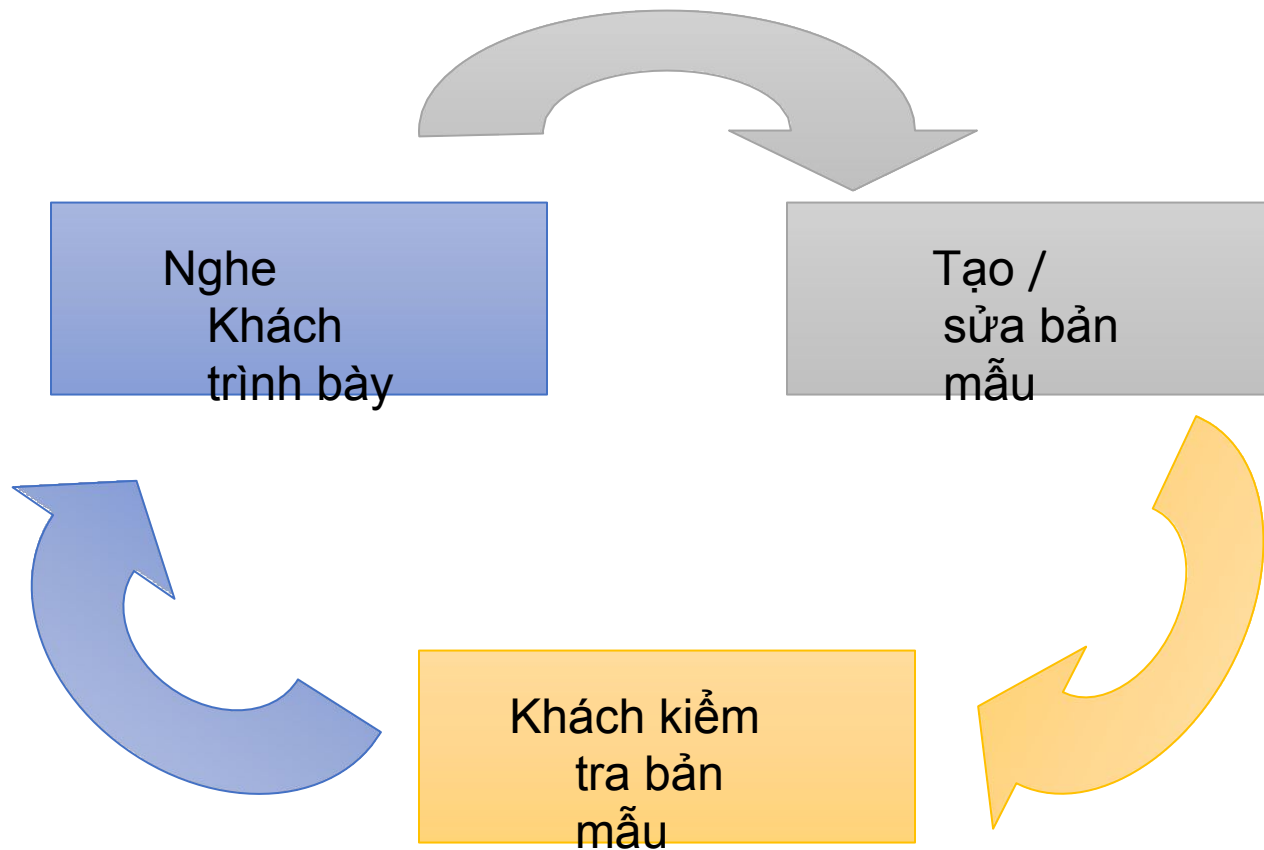
- Điểm mạnh:

- Hoàn thành một **giai đoạn** trước khi tiếp tục giai đoạn tiếp
- Nhấn mạnh việc lập kế hoạch sớm, đầu vào của khách hàng và thiết kế
- Nhấn mạnh kiểm tra như một phần không thể thiếu của vòng đời
- Cung cấp các chất lượng ở mỗi giai đoạn vòng đời

- Điểm yếu:

- **Phụ thuộc vào các yêu cầu** được xác định sớm từ đầu
- Phụ thuộc vào việc **tách các yêu cầu** khỏi thiết kế
- Không khả thi trong một số trường hợp đòi hỏi có nhiều **thay đổi**
- Nhấn mạnh vào **sản phẩm hơn là quy trình**

4.2. Mô hình mẫu thử (Prototyping model)





4.2. Mô hình mẫu thử: Khi nào ?

- Khi mới rõ mục đích chung chung của phần mềm, chưa rõ chi tiết đầu vào hay xử lý ra sao hoặc chưa rõ yêu cầu đầu ra
- Dùng để thu thập yêu cầu qua các thiết kế nhanh
- Các giải thuật, kỹ thuật dùng làm bản mẫu có thể chưa nhanh, chưa tốt, miễn là có mẫu để thảo luận gợi ý yêu cầu của người dùng



4.3. Các mô hình tăng dần

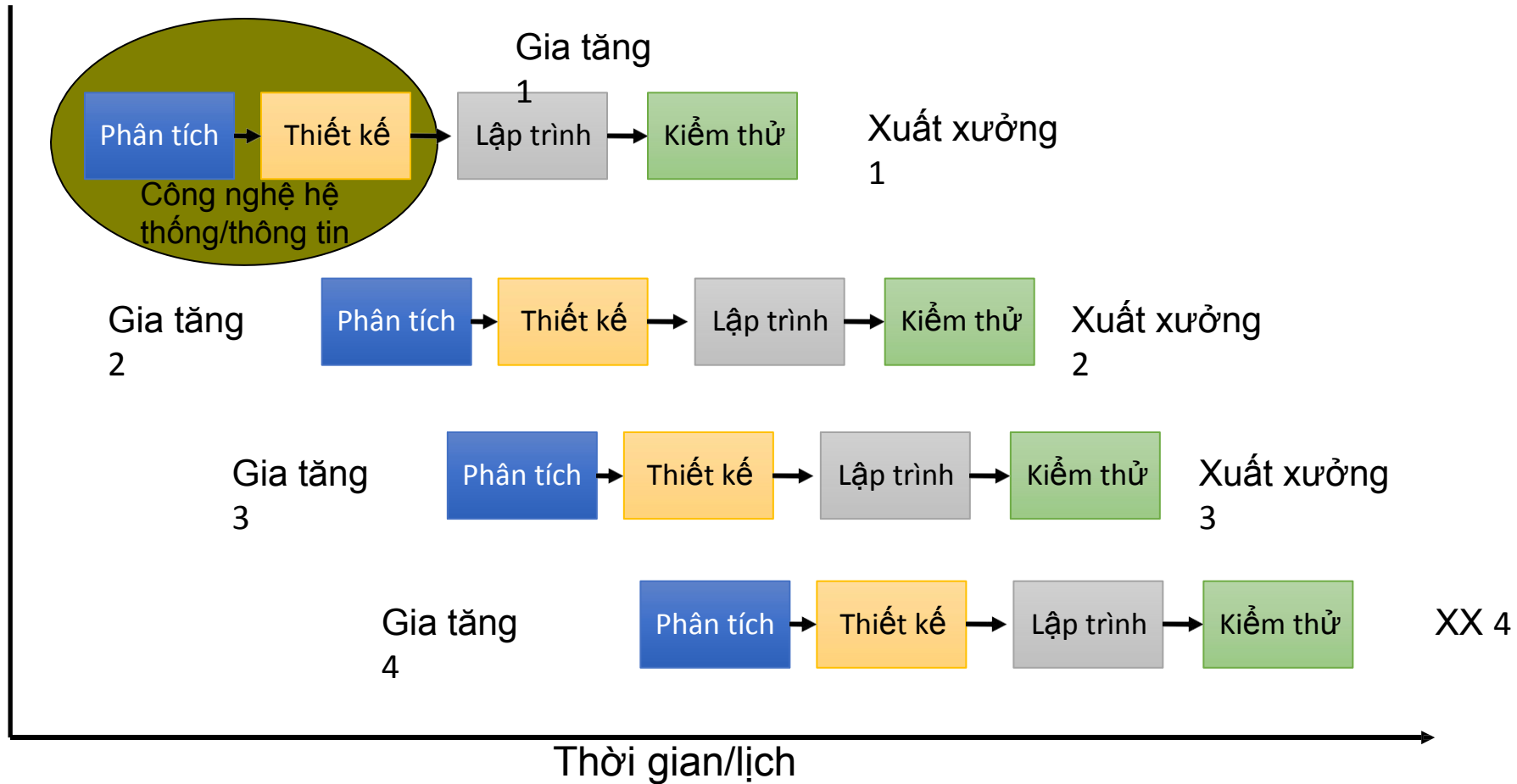
- Phần lớn các hệ phần mềm phức tạp đều **tiến hóa theo thời gian**: môi trường thay đổi, yêu cầu phát sinh thêm, hoàn thiện thêm chức năng, tính năng
- Các mô hình tiến hóa (evolutionary models) có tính **lặp lại**. Kỹ sư phần mềm tạo ra các phiên bản (versions) ngày càng hoàn thiện hơn, phức tạp hơn
- Các mô hình tiêu biểu:
 - Gia tăng (Incremental)
 - Xoắn Ốc (Spiral)
 - Xoắn Ốc WINWIN (WINWIN spiral)
 - Phát triển đồng thời (Concurrent development)

4.4. Mô hình gia tăng (The incremental model)

- Kết hợp mô hình **tuần tự** và ý tưởng **lặp lại** của chế bản mẫu
- Sản phẩm với những yêu cầu cơ bản nhất của hệ thống được phát triển
- Các chức năng với những yêu cầu khác được **phát triển thêm sau (gia tăng)**
- **Lặp lại quy trình** để hoàn thiện dần



4.4. Mô hình gia tăng



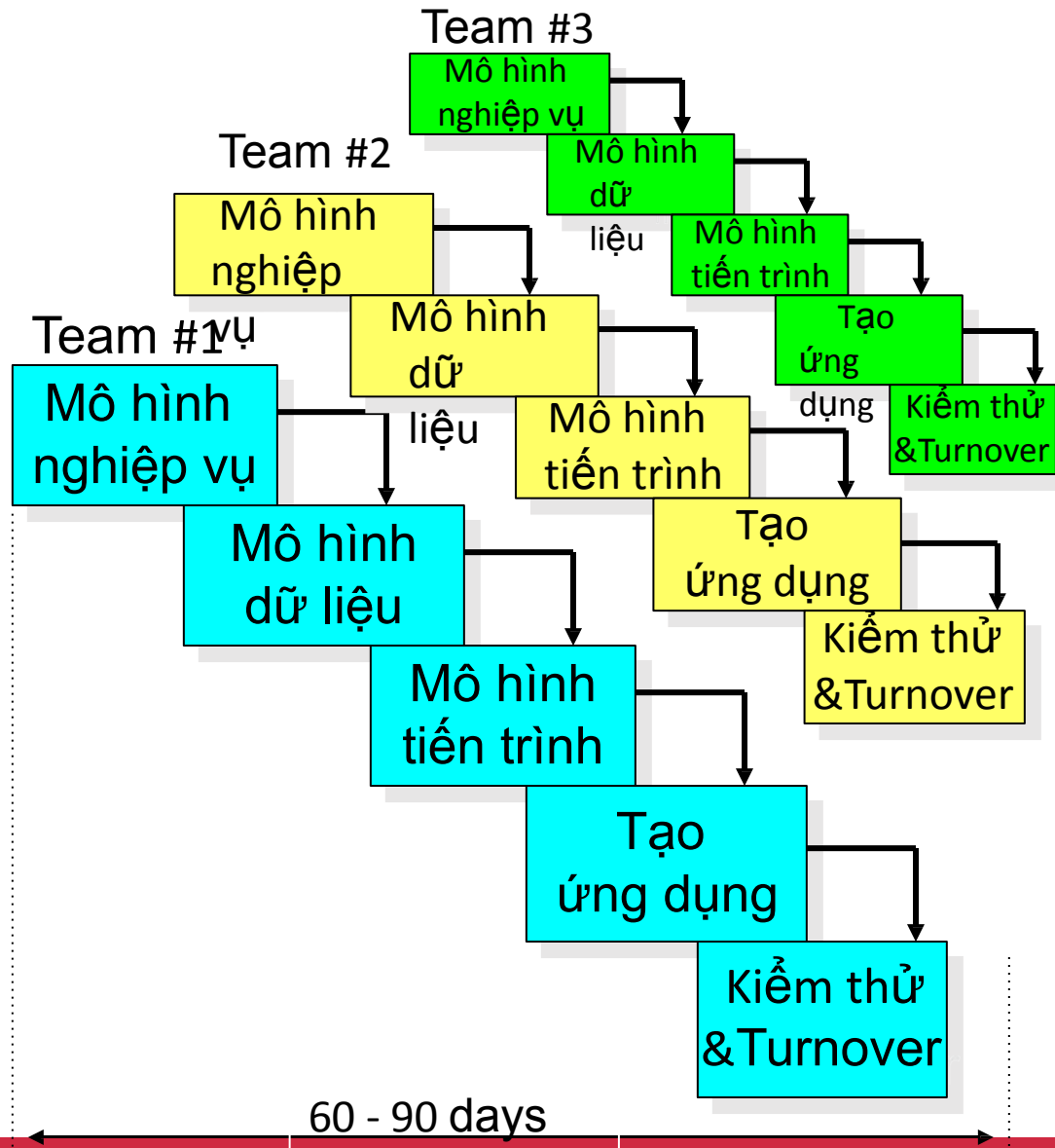


4.5. Mô hình phát triển ứng dụng

nhanh (Rapid Application Development: RAD)

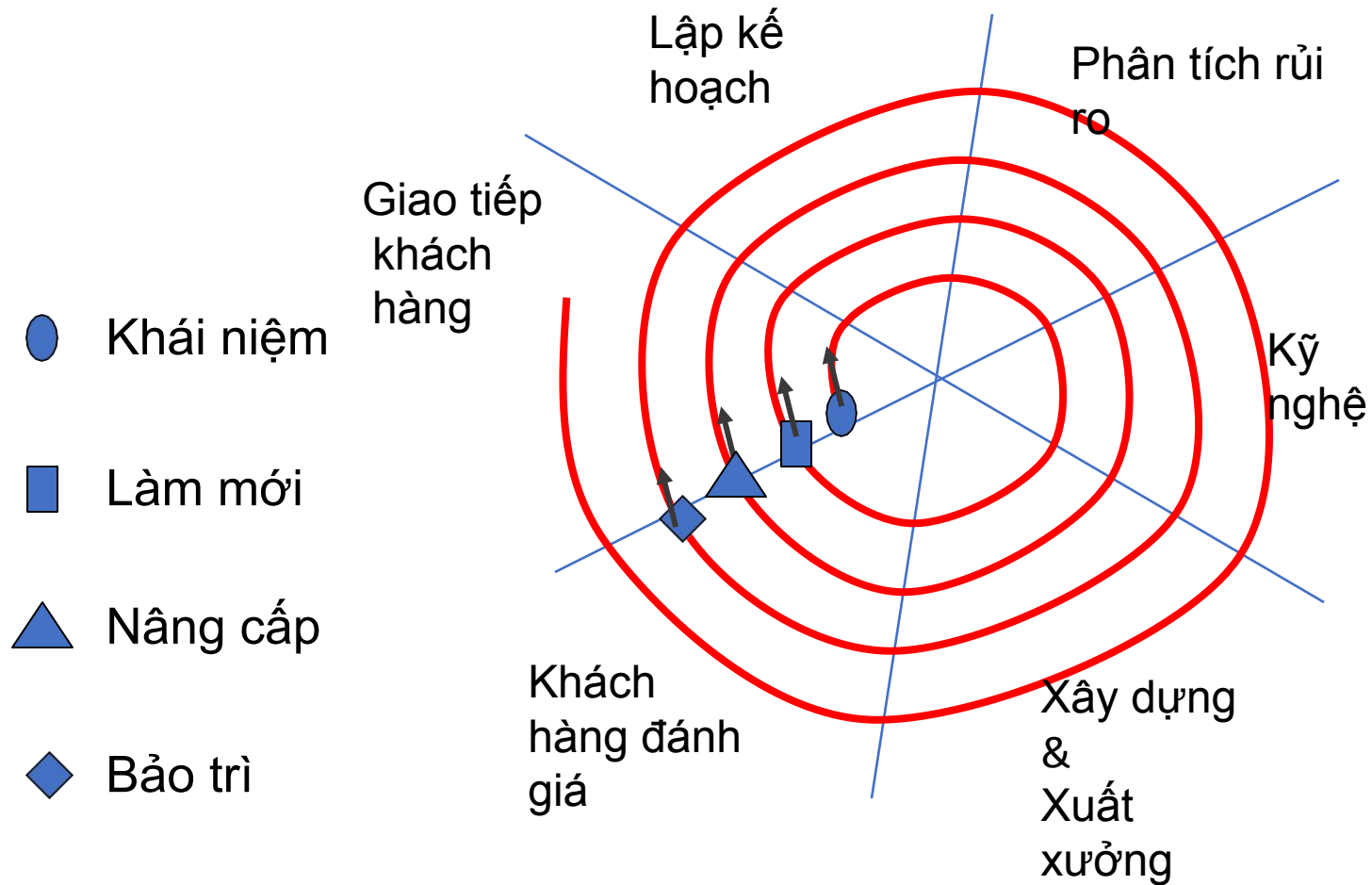
- Là quy trình phát triển phần mềm **gia tăng**, tăng dần từng bước (Incremental software development) với mỗi chu trình phát triển rất ngắn (60-90 ngày)
- Xây dựng dựa trên **hướng thành phần** (Component-based construction) với khả năng **tái sử dụng** (reuse)
- Gồm một số nhóm (teams), mỗi nhóm làm 1 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 phát triển ứng dụng nhanh





4.6. Mô hình xoắn ốc (spiral)





4.6. Mô hình xoắn ốc (tiếp)

- **Giao tiếp khách hàng:** giữa người phát triển và khách hàng để tìm hiểu yêu cầu, ý kiến
- **Lập kế hoạch:** Xác lập tài nguyên, thời hạn và những thông tin khác
- **Phân tích rủi ro:** Xem xét mạo hiểm kỹ thuật và mạo hiểm quản lý
- **Kỹ nghệ:** Xây dựng một hay một số biểu diễn của ứng dụng



4.6. Mô hình xoắn ốc (tiếp)

- **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 người sử dụng về biểu diễn phần mềm trong giai đoạn kỹ nghệ và cài đặt

4.6. Mô hình xoắn ốc: Mạnh và yếu?

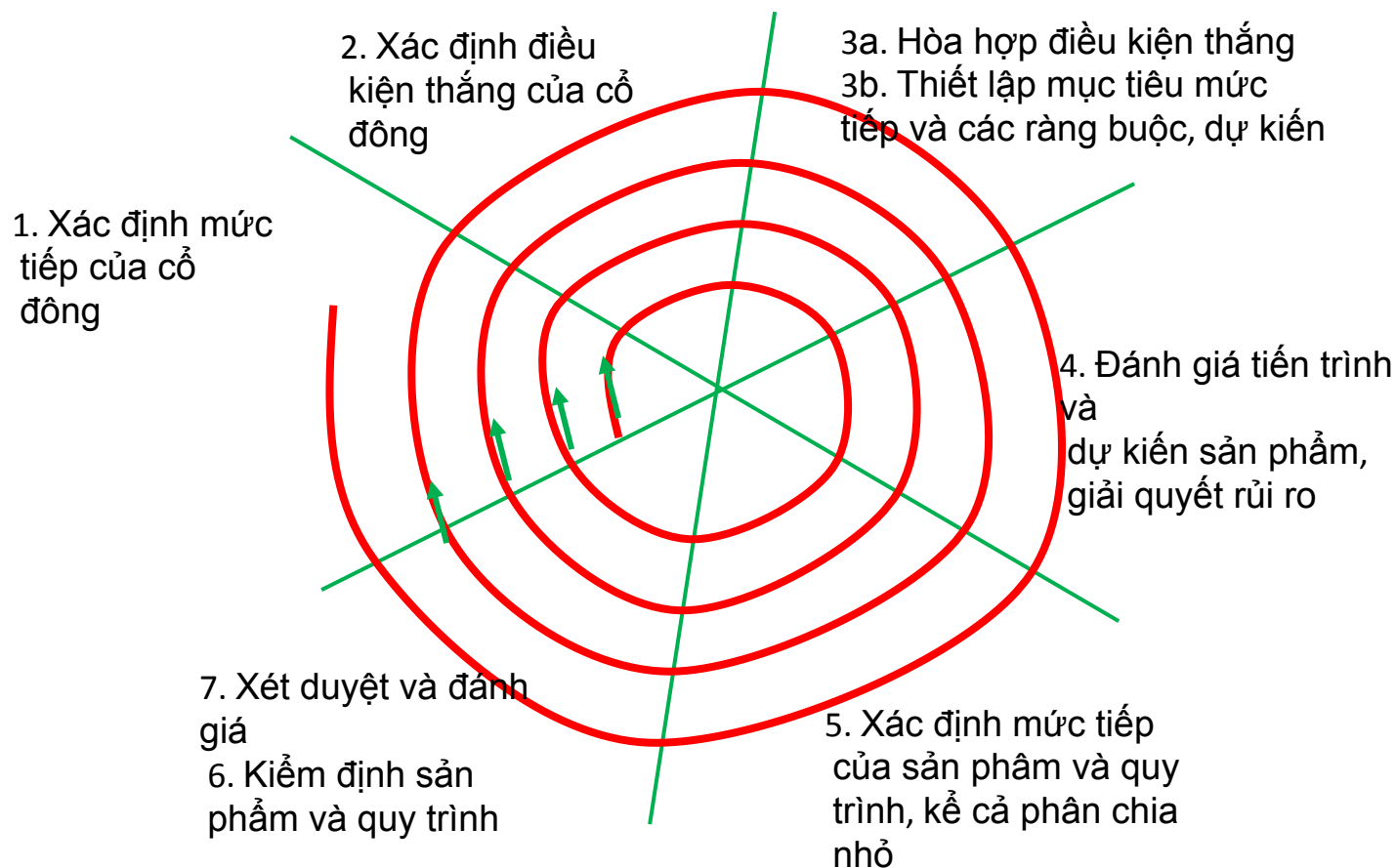
- Tốt cho các hệ phần mềm **quy mô lớn**
- Dễ **kiểm soát các mạo hiểm** ở từng mức tiến hóa
- **Khó thuyết phục** khách hàng là phương pháp tiến hóa xoắn ốc có thể kiểm soát được
- **Chưa được dùng rộng rãi** như các mô hình tuyến tính hoặc chế thử



4.7. Mô hình xoắn ốc WINWIN

- Nhằm **thỏa hiệp** giữa người phát triển và khách hàng, cả hai cùng “Thắng” (win-win)
 - Khách thì có phần mềm thỏa mãn yêu cầu chính
 - Người phát triển thì có kinh phí thỏa đáng và thời gian hợp lý
- Các hoạt động chính trong xác định hệ thống:
 - Xác định cổ đông (stakeholders)
 - Xác định điều kiện thắng của cổ đông
 - Thỏa hiệp điều kiện thắng của các bên liên quan

4.7. Mô hình xoắn ốc WINWIN





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**



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.