



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY



Viện Toán ứng dụng và Tin học

# Phân tích thiết kế hệ thống

## Quy trình phát triển phần mềm



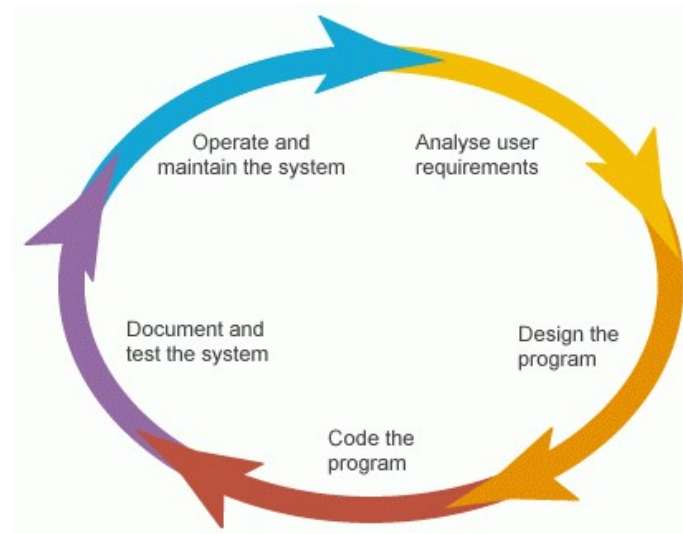
## Quy trình phát triển HTTT

- Quy trình là tuần tự các bước, các thao tác được thực hiện để xây dựng hệ thống
- Quy trình cho biết phải bắt đầu từ đâu, làm gì và làm xong thì làm việc gì tiếp



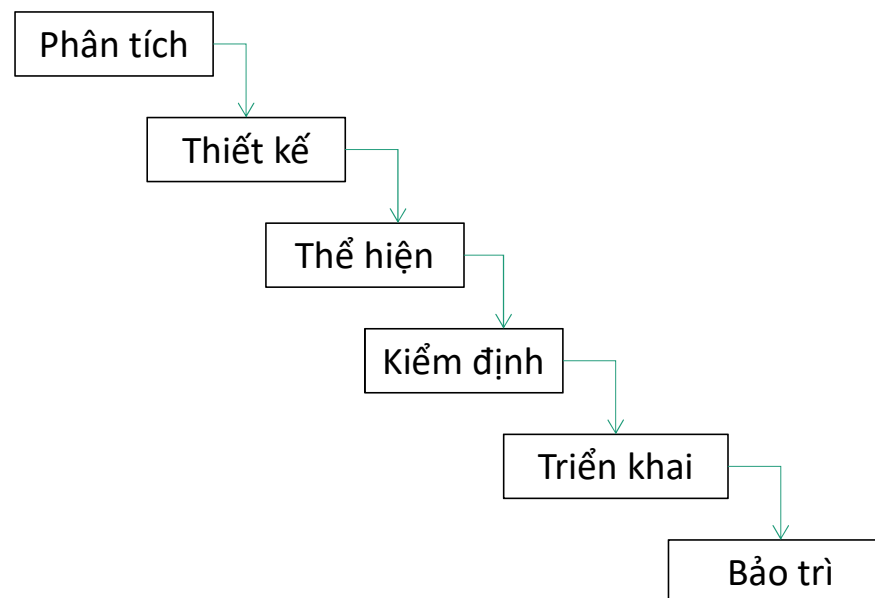
# Quy trình phát triển

- Software Development Life Cycle (SDLC)



# Quy trình thác nước (waterfall)

- Do Winston W. Royce đề xuất năm 1970



## Quy trình thác nước

- Bước 1 – Thu thập và phân tích yêu cầu: tất cả các yêu cầu của phần mềm đang được phát triển phải được thu thập, phân tích và viết tài liệu trong tài liệu “đặc tả yêu cầu”
- Bước 2 – Thiết kế hệ thống: giải pháp xây dựng hệ thống được đề xuất. Thiết kế này xác định yêu cầu phần cứng, phần mềm và kiến trúc tổng thể hệ thống
- Bước 3 – Thể hiện/lập trình: các chương trình nhỏ (các đơn vị) được phát triển. Các đơn vị được phát triển và kiểm định chức năng của nó (gọi là unit test)

## Quy trình thác nước

- Bước 4 – Tích hợp và kiểm thử: tất cả các đơn vị được phát triển trong bước 3 được tích hợp vào hệ thống và kiểm thử
- Bước 5 – Triển khai hệ thống: một khi các kiểm thử chức năng và phi chức năng đã được thực hiện, sản phẩm được triển khai ở môi trường của khách hàng hay công bố ra thị trường.
- Bước 6 – Bảo trì: khi có lỗi được phát hiện trong môi trường khách hàng. Các bản vá lỗi cần được công bố. Sản phẩm với phiên bản tốt hơn cũng có thể được công bố.

## Ứng dụng của quy trình thác nước

- Các phần mềm khác nhau thường đòi hỏi các quy trình phát triển khác nhau (tùy biến) phụ thuộc vào nhiều nhân tố bên trong và bên ngoài.
- Mô hình thác nước phù hợp nhất khi:
  - Các yêu cầu được tài liệu rõ ràng và không thay đổi
  - Xác định sản phẩm ổn định
  - Công nghệ được hiểu rõ ràng và ổn định
  - Không có yêu cầu mơ hồ
  - Các tài nguyên phong phú sẵn sàng hỗ trợ
  - Dự án ngắn

## Ưu điểm của quy trình thác nước

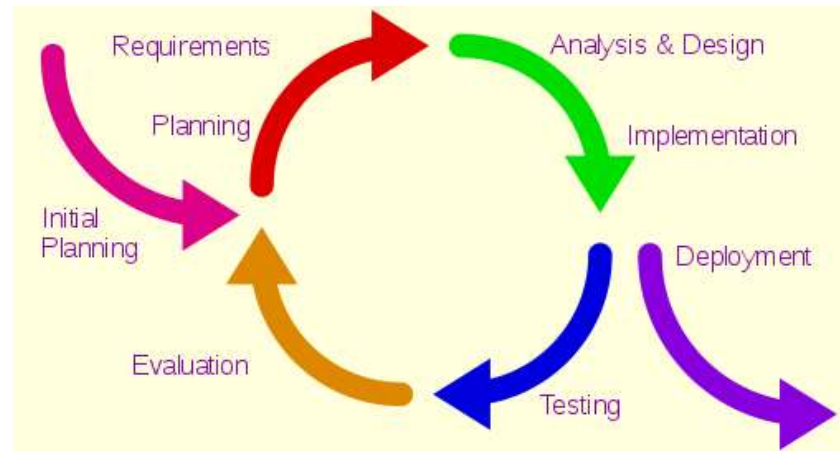
- Đơn giản, dễ hiểu và dễ sử dụng
- Dễ quản lý.
- Mỗi giai đoạn có một số sản phẩm xác định
- Các giai đoạn được thực hiện và hoàn thành không chồng lấn nhau
- Dễ tổ chức các công việc
- Tiến trình và các kết quả được tài liệu tốt



## Nhược điểm của quy trình thác nước

- Phần mềm không được tạo ra cho tới cuối quy trình
- Rủi ro cao và không chắc chắn
- Không phù hợp cho các thay đổi yêu cầu
- Điều chỉnh phạm vi chỉ ở cuối quy trình
- Tích hợp là một công việc khó khăn
- Không cho phép xác định các điểm nghẽn về công nghệ và kinh doanh sớm

## Quy trình lặp, tăng trưởng



- Lập kế hoạch: phân tích rủi ro, xác định công việc trong vòng lặp
- Đánh giá kết quả của vòng lặp

## Quy trình lặp, tăng trưởng

- Quy trình lặp bắt đầu với những thể hiện đơn giản của tập nhỏ các yêu cầu phần mềm, sau đó tiến triển dần tới khi hoàn chỉnh hệ thống.
- Không cố để hoàn chỉnh tất cả các đặc tả yêu cầu trước khi bắt đầu thể hiện
- Ở mỗi vòng lặp, thiết kế được hiệu chỉnh, chức năng mới được bổ sung
- Mỗi vòng lặp module được phát triển qua các giai đoạn: yêu cầu, thiết kế, thể hiện và kiểm định

# Ứng dụng của quy trình lặp, tăng trưởng

- Tiến trình được sử dụng trong các điều kiện sau:
  - Các yêu cầu của toàn bộ hệ thống được hiểu và xác định rõ ràng
  - Các yêu cầu chính được xác định, tuy nhiên các chức năng và yêu cầu nâng cao sẽ tiến triển theo thời gian
  - Sử dụng công nghệ mới và đang được học bởi nhóm phát triển
  - Nguồn lực với các kỹ năng cần thiết chưa sẵn sàng và được lập kế hoạch sử dụng dựa trên các vòng lặp xác định
  - Có một số rủi ro cao và mục tiêu cũng có thể thay đổi

## Ưu điểm của quy trình lắp, tăng trưởng

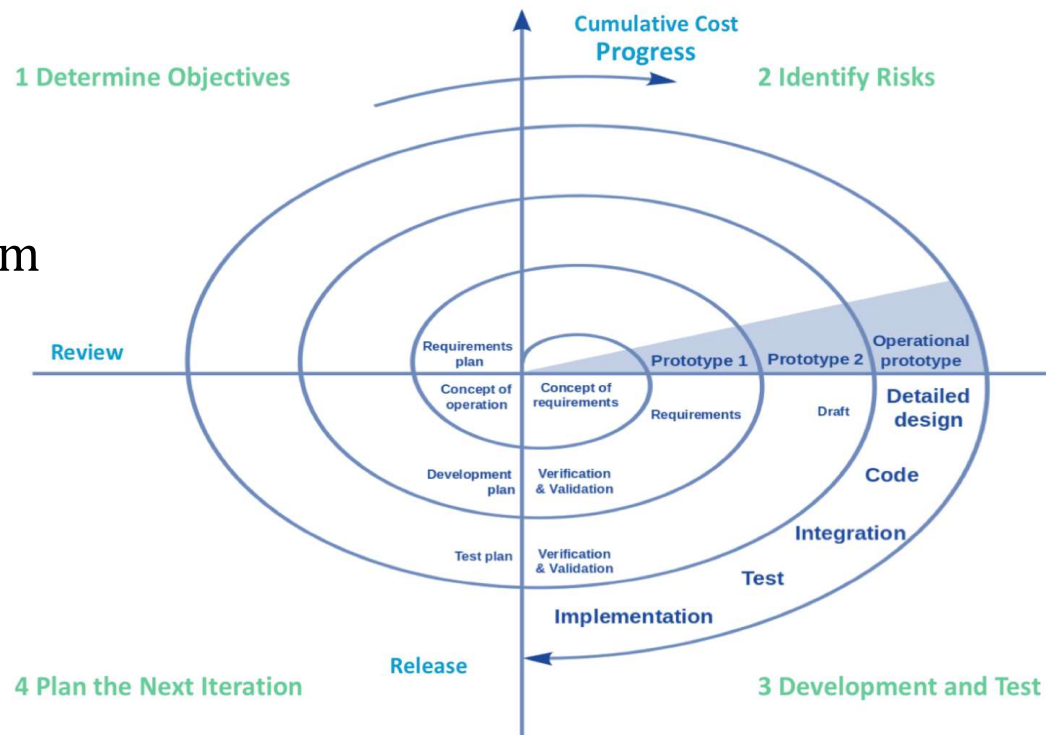
- Một số chức năng có thể được phát triển nhanh và sớm
- Có kết quả theo từng chu kỳ phát triển
- Có thể phát triển song song
- Tiến trình có thể được đánh giá, quản lý
- Giảm giá phải trả cho việc thay đổi yêu cầu
- Quản lý rủi ro dễ hơn
- Phù hợp hơn với các dự án lớn và quan trọng
- Sớm đáp ứng các đánh giá và phản hồi của khách hàng

## Nhược điểm của quy trình lặp, tăng trưởng

- Yêu cầu nhiều nguồn lực
- Đòi hỏi nhiều nỗ lực quản lý
- Tồn tại các vấn đề về thiết kế và kiến trúc hệ thống
- Không phù hợp với các dự án nhỏ
- Yêu cầu nguồn lực kỹ năng cao cho phân tích rủi ro
- Tiến trình dự án phụ thuộc pha phân tích rủi ro

# Mô hình xoắn ốc

- Là mô hình risk-driven
- Đề xuất bởi Barry Boehm năm 1986



## Mô hình xoắn ốc

- Mỗi vòng đại diện cho 1 pha phát triển
- Vòng trong cùng tập trung vào tính khả thi
- Vòng tiếp theo tập trung vào xác định các yêu cầu
- Vòng kế tiếp tập trung vào thiết kế, vận hành
- Mỗi vòng gồm 4 phần:
  - Xác định đối tượng
  - Định lượng và giảm thiểu rủi ro
  - Phát triển và đánh giá
  - Lập kế hoạch



## Ứng dụng của quy trình xoắn ốc

- Quy trình xoắn ốc được sử dụng trong các điều kiện sau:
  - Khi có ràng buộc về tài chính và việc đánh giá rủi ro là quan trọng
  - Dự án với độ rủi ro từ trung bình tới cao
  - Cam kết dự án dài hơi
  - Khách hàng không chắc chắn về yêu cầu
  - Các yêu cầu phức tạp và cần được đánh giá kỹ càng
  - Sản phẩm cần nhận phản hồi của khách hàng ở các giai đoạn
  - Mong muốn các thay đổi đáng kể trong quá trình phát triển

## Ưu điểm của quy trình xoắn ốc

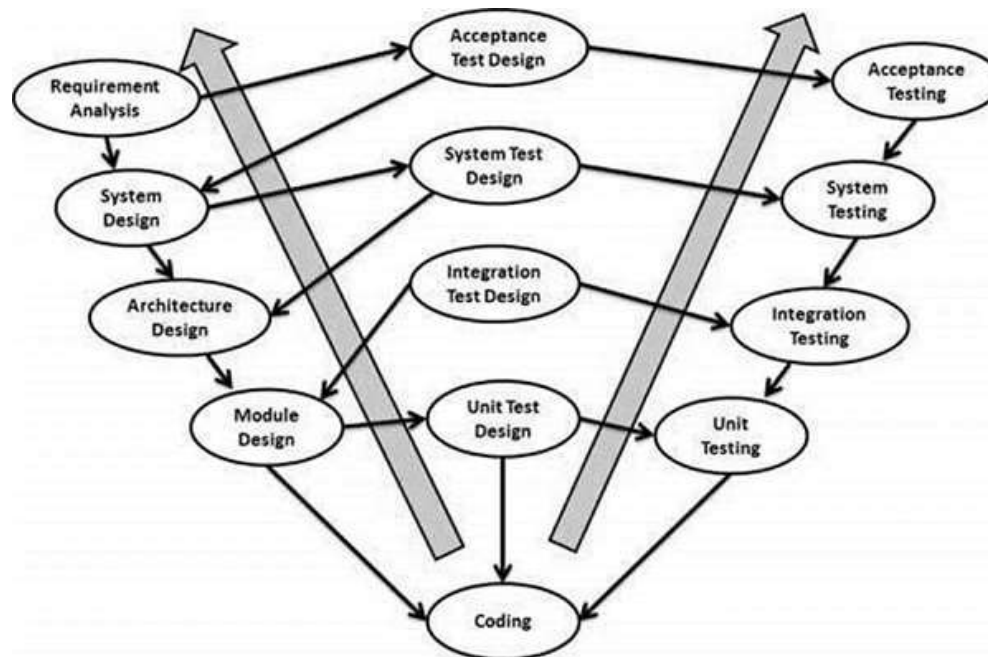
- Thích ứng các yêu cầu thay đổi
- Cho phép sử dụng nguyên mẫu
- Thu thập các yêu cầu chính xác hơn
- Người dùng được thấy hệ thống sớm
- Việc phát triển được chia vào các phần nhỏ, các phần rủi ro được thực hiện sớm giúp quản lý rủi ro tốt hơn.

## Nhược điểm của quy trình xoắn ốc

- Quản lý phức tạp
- Không biết sớm điểm kết thúc của dự án
- Không phù hợp với dự án nhỏ, rủi ro thấp
- Quy trình phức tạp
- Vòng lặp có thể vô hạn
- Số lớn các giai đoạn trung gian đòi hỏi nhiều công sức viết tài liệu

# Quy trình chữ V

- Còn gọi là quy trình kiểm định và xác thực
- Pha kiểm định và pha phát triển được lập kế hoạch song song



## Quy trình chữ V

- **Phân tích yêu cầu:** xác định chính xác mong muốn và đòi hỏi của khách hàng; thiết kế kiểm định chấp nhận (acceptance test) cũng được tạo ra.
- **Thiết kế hệ thống:** xây dựng tài liệu đặc tả và sơ lược các thành phần kỹ thuật như các tầng dữ liệu, logic nghiệp vụ. Thiết kế kiểm định hệ thống cũng được tạo ra.
- **Thiết kế kiến trúc:** chi tiết liên kết giữa các thành phần của ứng dụng thông qua việc tích hợp trong hay ngoài. Thiết kế kiểm định tích hợp cũng được thực hiện.

## Quy trình chữ V

- **Thiết kế module:** thiết kế chi tiết bên trong cho tất cả các module (thiết kế mức thấp). Thiết kế cần tương thích giữa các module và với các hệ thống bên ngoài. Thiết kế kiểm định đơn nguyên cũng được thực hiện
- **Mã hóa:** code thực sự của hệ thống được viết tuân thủ các chuẩn và quy ước mã
- **Kiểm định đơn nguyên:** thực hiện kiểm định các đơn nguyên đã được thiết kế

## Quy trình chữ V

- **Kiểm định tích hợp:** kiểm định tích hợp được thực hiện để kiểm định sự giao tiếp giữa các module trong hệ thống
- **Kiểm định hệ thống:** kiểm định toàn bộ chức năng của hệ thống, giao tiếp giữa hệ thống và các hệ thống bên ngoài. Kiểm định sự tương thích phần cứng và phần mềm
- **Kiểm định chấp nhận:** kiểm định sản phẩm trong môi trường của người dùng cuối. Nó có thể phát hiện các vấn đề tương thích với các hệ thống khác cũng như các vấn đề phi chức năng khác như tải trọng, hiệu năng trong môi trường vận hành thực.

## Ứng dụng mô hình chữ V

- Các yêu cầu được xác định, tài liệu hóa rõ ràng và không thay đổi.
- Định nghĩa sản phẩm ổn định
- Công nghệ không động và được hiểu tốt bởi đội ngũ phát triển
- Không có các yêu cầu mơ hồ hay không xác định
- Dự án ngắn



## Ưu điểm của quy trình chữ V

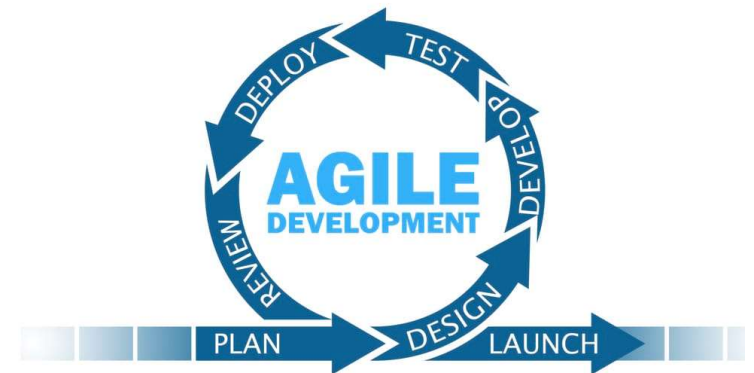
- Quy trình chặt chẽ, các giai đoạn được thực hiện tuần tự
- Tốt cho các dự án nhỏ với các yêu cầu được xác định và hiểu rõ ràng
- Đơn giản, dễ hiểu và sử dụng
- Dễ quản lý, mỗi giai đoạn xác định rõ các kết quả đầu ra cũng như việc xem xét đánh giá

## Nhược điểm của quy trình chữ V

- Rủi ro cao và không chắc chắn
- Không tốt cho các dự án phức tạp
- Không tốt cho dự án dài hạn
- Không phù hợp với các dự án mà các yêu cầu gánh chịu sự thay đổi ở mức trung bình và cao
- Một khi ứng dụng ở giai đoạn kiểm định, khó để thay đổi chức năng của nó
- Không tạo sản phẩm phần mềm cho tới khi kết thúc giai đoạn sau cùng.

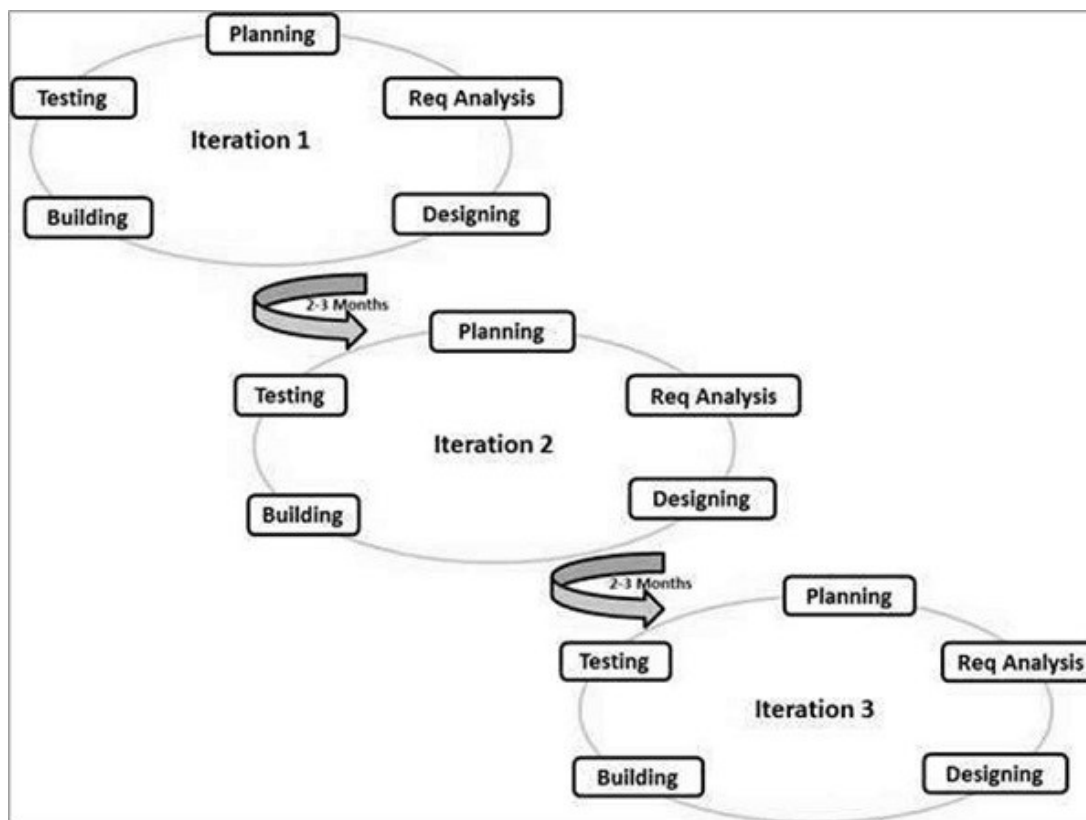
## Quy trình phát triển linh hoạt

- Quy trình phát triển linh hoạt (Agile) và các phương pháp luận linh hoạt bắt nguồn từ tuyên ngôn linh hoạt năm 2001
- Agile là sự kết hợp của các mô hình lặp có tăng trưởng tập trung vào việc thích ứng và thỏa mãn khách hàng bằng việc phát hành nhanh các sản phẩm phần mềm hoạt động.
- Mỗi vòng lặp nên ngắn khoảng từ 1 tới 3 tuần



# Quy trình phát triển linh hoạt

- Tiếp cận thích ứng
- Tương tác khách hàng
- Agile phát hành sản phẩm theo những khoảng thời gian cố định



# Quy trình phát triển linh hoạt

## Tuyên ngôn Phát triển phần mềm linh hoạt

*Chúng tôi đã phát hiện ra cách phát triển phần mềm tốt hơn bằng cách thực hiện nó và giúp đỡ người khác thực hiện.*

*Qua công việc này, chúng tôi đã đi đến việc đánh giá cao:*

**Cá nhân và sự tương tác** hơn là quy trình và công cụ;

**Phần mềm chạy tốt** hơn là tài liệu đầy đủ;

**Cộng tác với khách hàng** hơn là đàm phán hợp đồng;

**Phản hồi với các thay đổi** hơn là bám sát kế hoạch.

*Mặc dù các điều bên phải vẫn còn giá trị, nhưng chúng tôi đánh giá cao hơn các mục ở bên trái.*

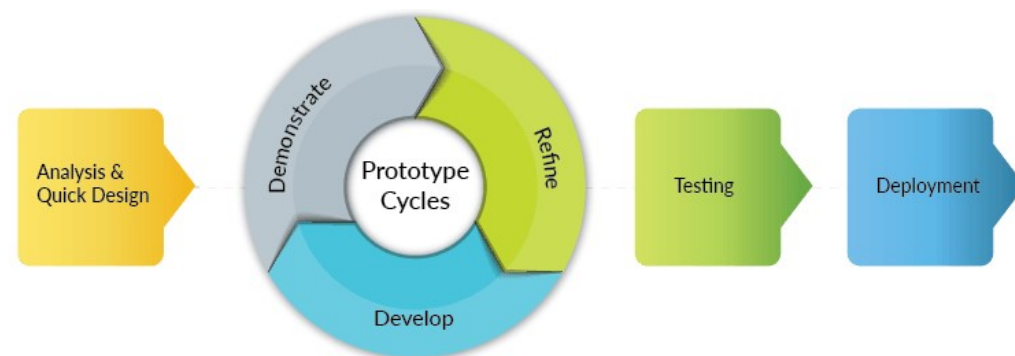
## Ưu điểm của quy trình phát triển linh hoạt

- Phù hợp cho cả các yêu cầu cố định và thay đổi
- Thúc đẩy nhóm phát triển và đào tạo chéo
- Chức năng được phát triển nhanh và giới thiệu với khách hàng
- Không đòi hỏi nhiều tài nguyên
- Phát hành sản phẩm từng phần sớm
- Cho phép phát triển đồng thời

## Nhược điểm của quy trình phát triển linh hoạt

- Không phù hợp cho các vấn đề có sự phụ thuộc phức tạp
- Phụ thuộc nhiều vào khách hàng, nếu khách hàng không rõ ràng, nhóm phát triển có thể đi sai hướng
- Phụ thuộc nhiều vào các cá nhân
- Khó chuyển giao công nghệ cho nhóm mới do thiếu vắng tài liệu

## Quy trình phát triển nhanh



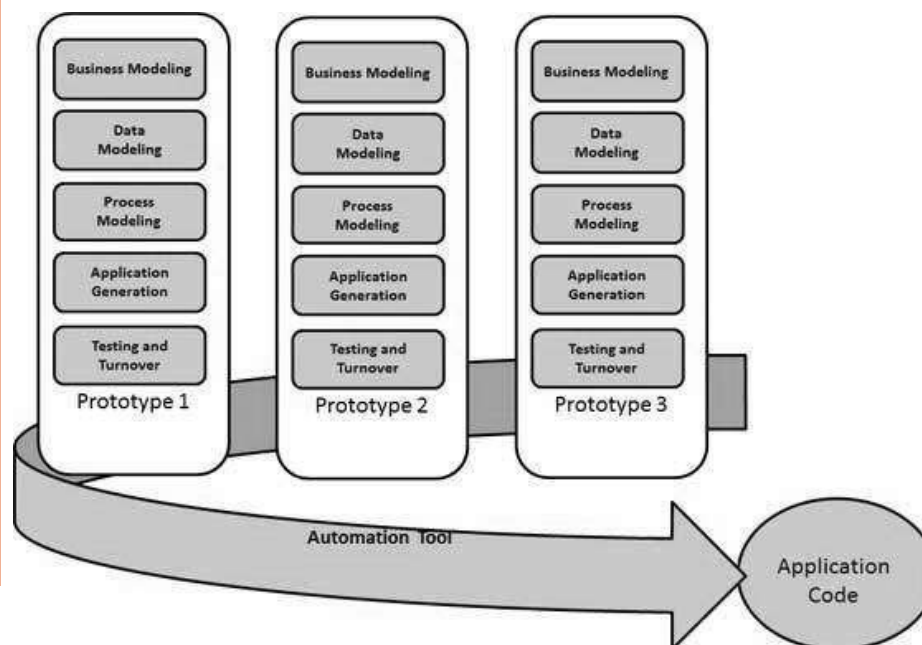
- Quy trình phát triển nhanh (Rapid Application Development - RAD) được đề xuất bởi James Martin, năm 1991
- RAD là quy trình dựa trên mô hình phát triển nguyên mẫu và lặp mà không lập kế hoạch chi tiết.
- RAD tập trung vào việc thu thập yêu cầu người dùng thông qua các hội thảo, nhóm làm việc
- Kiểm định sớm các nguyên mẫu bởi người dùng
- Sử dụng khái niệm lặp với sự sử dụng lại các nguyên mẫu (các thành phần)
- Tích hợp liên tục và phát hành sản phẩm nhanh



# Quy trình phát triển nhanh

- Tạo nguyên mẫu nhanh, giảm thiểu lập kế hoạch
- Các nguyên mẫu (module chức năng, thành phần) được phát triển song song và được tích hợp vào sản phẩm cuối

- RAD tuân theo mô hình **lặp có tăng trưởng**, có các nhóm nhỏ người phát triển, chuyên gia nghiệp vụ, đại diện khách hàng và các nguồn lực IT khác xây dựng nguyên mẫu hay thành phần
- Điểm mấu chốt là các nguyên mẫu sẽ được sử dụng lại



## Quy trình phát triển nhanh

**RAD** phân phối các pha phân tích, thiết kế, xây dựng, kiểm định vào chuỗi các vòng lặp phát triển ngắn gồm:

- **Mô hình hóa nghiệp vụ**: mô tả luồng thông tin và sự phân phối của thông tin giữa các kênh nghiệp vụ
- **Mô hình hóa dữ liệu**: mô tả tập các đối tượng dữ liệu gồm thuộc tính của đối tượng và mối quan hệ giữa các đối tượng

## Quy trình phát triển nhanh

- **Mô hình hóa xử lý:** mô tả luồng thông tin cần thiết để đạt mục tiêu nghiệp vụ cụ thể, mô tả bao gồm việc thêm, hủy bỏ, thu thập và biến đổi đối tượng dữ liệu
- **Xây dựng ứng dụng:** viết mã và xây dựng nguyên mẫu
- **Kiểm định và chuyển giao:** kiểm định nguyên mẫu của mỗi vòng lặp và kiểm định luồng dữ liệu và giao diện giữa các thành phần. Chuyển giao nguyên mẫu

Các nguyên mẫu được phát triển và sử dụng lại

## Ứng dụng quy trình phát triển nhanh

- Hệ thống có thể được module hóa và có thể phân phối tăng dần
- Đội ngũ thiết kế viên sẵn sàng cho việc mô hình hóa
- Đội ngũ chuyên gia nghiệp vụ sẵn sàng
- Các yêu cầu thay đổi trong quá trình phát triển

## Ưu điểm của quy trình RAD

- Các yêu cầu thay đổi được đáp ứng
- Quy trình có thể đo đạc được
- Hiệu quả với dự án ít người, thời gian ngắn
- Giảm thời gian phát triển
- Tăng khả năng sử dụng lại các thành phần
- Nhanh chóng có đánh giá ban đầu
- Khuyến khích phản hồi của khách hàng
- Tích hợp sớm nên giải quyết được các khó khăn của việc tích hợp

## Nhược điểm của quy trình RAD

- Phụ thuộc nhiều vào khả năng của đội phát triển trong việc xác định yêu cầu
- Chỉ phù hợp với các hệ thống có khả năng module hóa
- Đòi hỏi kỹ năng lập trình và thiết kế cao
- Quản lý phức tạp
- Chỉ thích hợp với các hệ thống dựa trên thành phần và có khả năng mở rộng
- Đòi hỏi người dùng cuối tham gia vào quy trình phát triển
- Chỉ phù hợp với các dự án ngắn