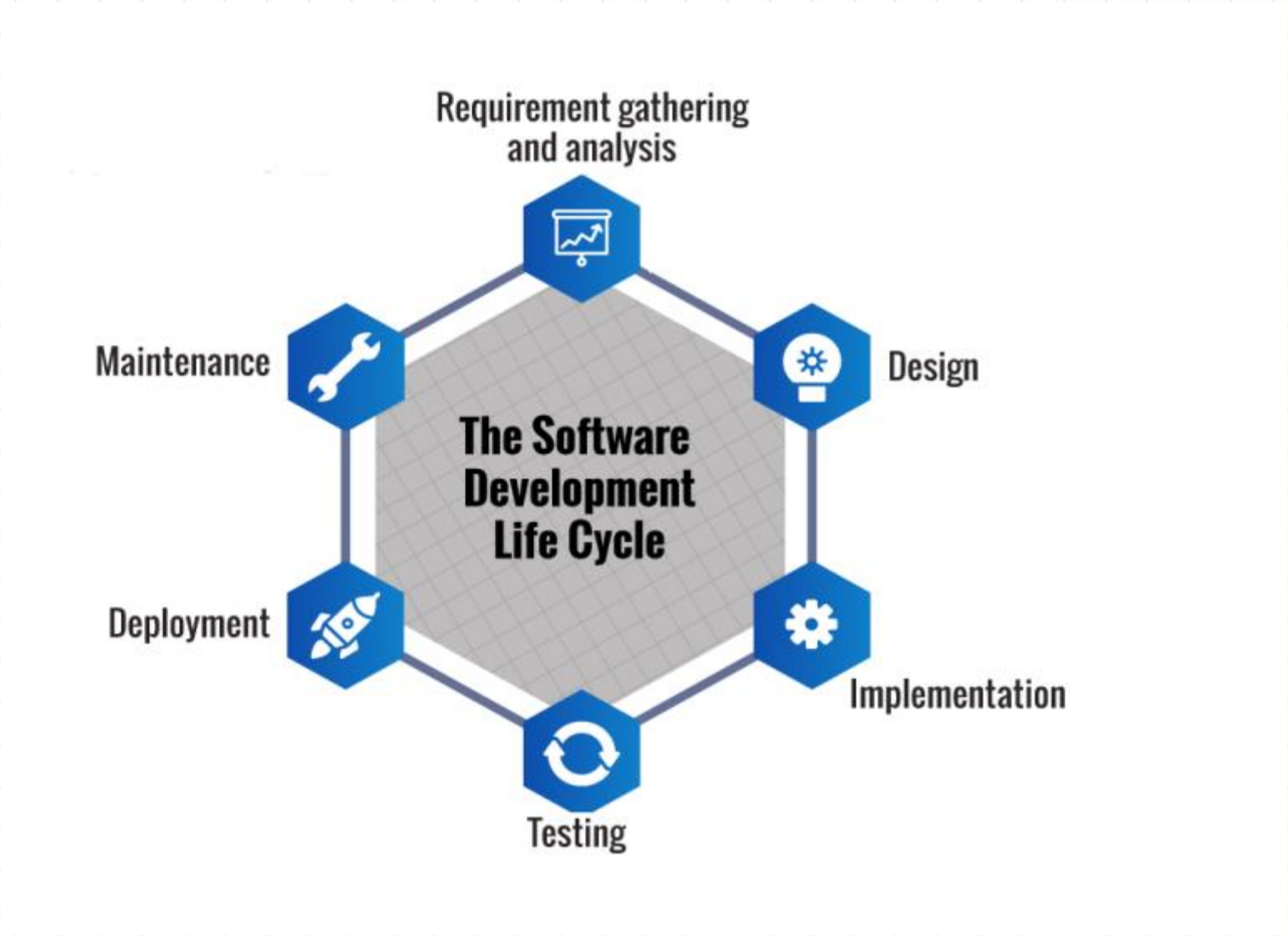
MỘT SỐ MÔ HÌNH PHÁT TRIỂN PHẦN MỀM

GV: VÕ TUẤN KIỆT

29/04/2025

Vòng Đời Phát Triển Phần Mềm



SDLC - Requirement Gathering & Analysis

- **Mục tiêu:**
  - ✓ Hiểu rõ nhu cầu của khách hàng, người dùng cuối.
  - ✓ Xác định các yêu cầu chức năng và phi chức năng.
- **Hoạt động chính:**
  - ✓ Phỏng vấn khách hàng, khảo sát, phân tích hệ thống hiện tại (nếu có).
  - ✓ Ghi nhận yêu cầu nghiệp vụ.
  - ✓ Xác định các ràng buộc (ngân sách, thời gian, kỹ thuật, pháp lý...).
- **Kết quả:**
  - ✓ Tài liệu đặc tả yêu cầu (SRS – Software Requirement Specification).
  - ✓ Use cases, sơ đồ luồng dữ liệu (DFD), sơ đồ hoạt động...

SDLC – System Design

- **Mục tiêu:**
  - ✓ Biến yêu cầu thành thiết kế kỹ thuật cụ thể..
- **Hoạt động chính:**
  - ✓ Thiết kế kiến trúc hệ thống.
  - ✓ Thiết kế cơ sở dữ liệu.
  - ✓ Thiết kế giao diện người dùng (UI/UX).
  - ✓ Lựa chọn công nghệ, nền tảng, ngôn ngữ lập trình.
- **Kết quả:**
  - ✓ Tài liệu thiết kế hệ thống (Software Design Document – SDD).
  - ✓ Sơ đồ kiến trúc (Architecture Diagram), UML, ERD...

SDLC – Implementation & Coding

- **Mục tiêu:**
  - ✓ Hiện thực phần mềm theo thiết kế đã được duyệt.
- **Hoạt động chính:**
  - ✓ Phân chia công việc cho các lập trình viên.
  - ✓ Lập trình theo module (theo thiết kế).
  - ✓ Tích hợp giữa các module.
- **Kết quả:**
  - ✓ Mã nguồn.
  - ✓ Tài liệu hướng dẫn code, cấu trúc thư mục, build tool...

SDLC – Testing

- **Mục tiêu:**
  - ✓ Phát hiện lỗi, đảm bảo phần mềm hoạt động đúng theo yêu cầu.
- **Các cấp độ kiểm thử:**
  - ✓ **Integration Testing**: kiểm thử khi kết hợp các module.
  - ✓ **System Testing**: kiểm thử toàn hệ thống.
  - ✓ **Acceptance Testing**: kiểm thử chấp nhận (người dùng tham gia).
- **Kết quả:**
  - ✓ Báo cáo lỗi (bug report).
  - ✓ Tài liệu test case, test plan.
  - ✓ Phần mềm sẵn sàng triển khai.

SDLC – Deployment

- **Mục tiêu:**
  - ✓ Đưa phần mềm vào môi trường thực tế.
- **Mức độ triển khai:**
  - ✓ Triển khai thử nghiệm (beta release).
  - ✓ Triển khai chính thức (production).
- **Hoạt động:**
  - ✓ Cấu hình môi trường server, database, domain...
  - ✓ Đào tạo người dùng.
  - ✓ Tài liệu cài đặt, hướng dẫn sử dụng.

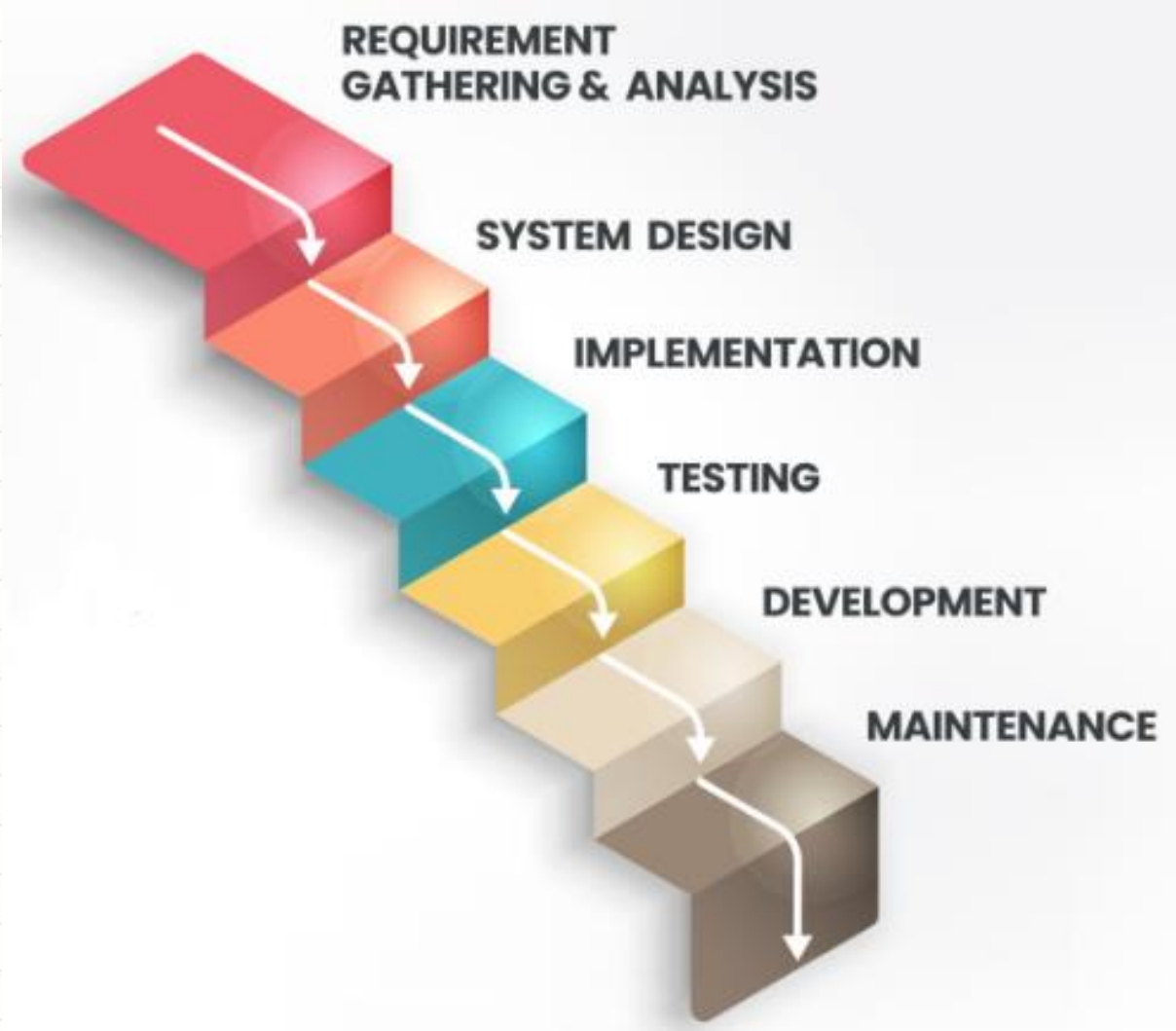
SDLC – Maintenance

- **Mục tiêu:**
  - ✓ Đảm bảo phần mềm hoạt động lâu dài, ổn định.
  - ✓ Phản hồi và cập nhật theo yêu cầu mới.
- **Các loại bảo trì:**
  - ✓ **Sửa lỗi (Corrective)**.
  - ✓ **Thích nghi (Adaptive)**: thay đổi theo môi trường mới.
  - ✓ **Nâng cấp (Perfective)**: cải tiến tính năng.
  - ✓ **Ngăn ngừa (Preventive)**: bảo trì phòng ngừa lỗi.

Một Số Mô Hình SDLC

- Mô hình thác nước (Waterfall Model)
- Mô hình lặp (Iterative Model)
- Mô hình linh hoạt (Agile Model)
- Mô hình DevOps (DevOps Model)

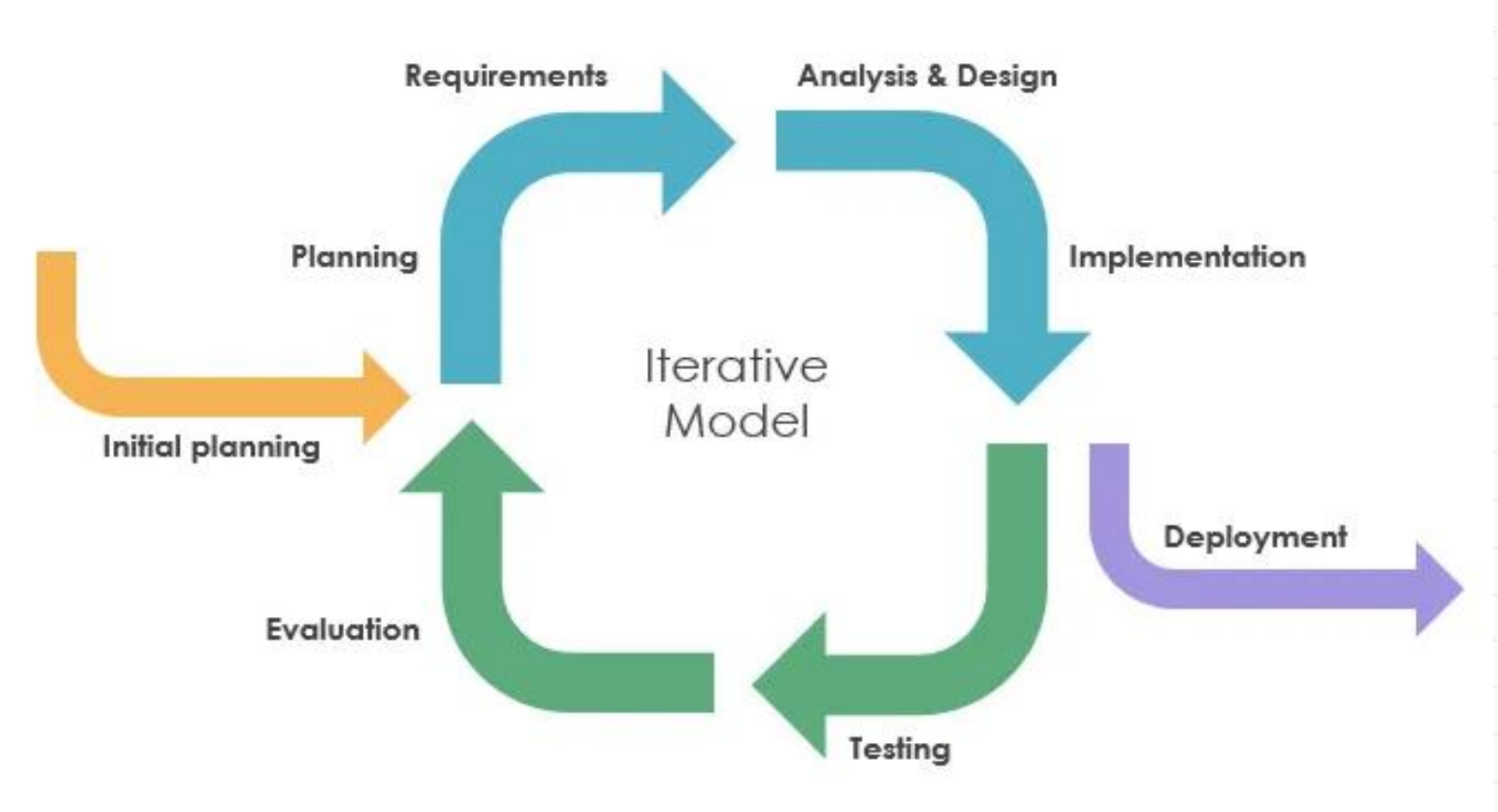
Mô Hình Thác Nước (Waterfall)



Mô Hình Thác Nước (Waterfall)

- **Ưu điểm:**
  - ❑ Quy trình rõ ràng, dễ quản lý
  - ❑ Tài liệu đầy đủ
- **Nhược điểm:**
  - ❑ Khó thay đổi yêu cầu
  - ❑ Phát hiện lỗi trễ
- **Phù hợp:**
  - ❑ Dự án yêu cầu ổn định, ít thay đổi
  - ❑ Dự án nhỏ hoặc trung bình

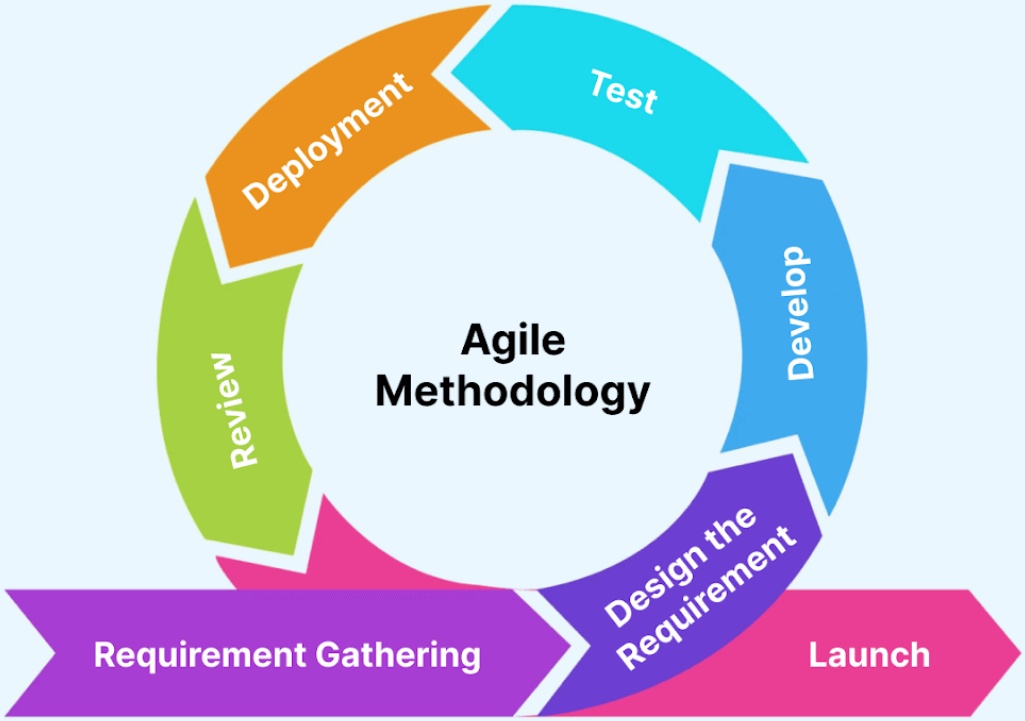
Mô Hình Lặp (Iterative)



Mô Hình Lặp (Iterative)

- **Ưu điểm:**
  - ❑ Phát hiện và sửa lỗi sớm
  - ❑ Dễ thích ứng với yêu cầu thay đổi
  - ❑ Giảm rủi ro trong dự án lớn
- **Nhược điểm:**
  - ❑ Tốn nhiều tài nguyên
  - ❑ Quản lý phức tạp hơn Waterfall
- **Phù hợp:**
  - ❑ Dự án cần phản hồi liên tục từ khách hàng
  - ❑ Yêu cầu ban đầu chưa rõ ràng

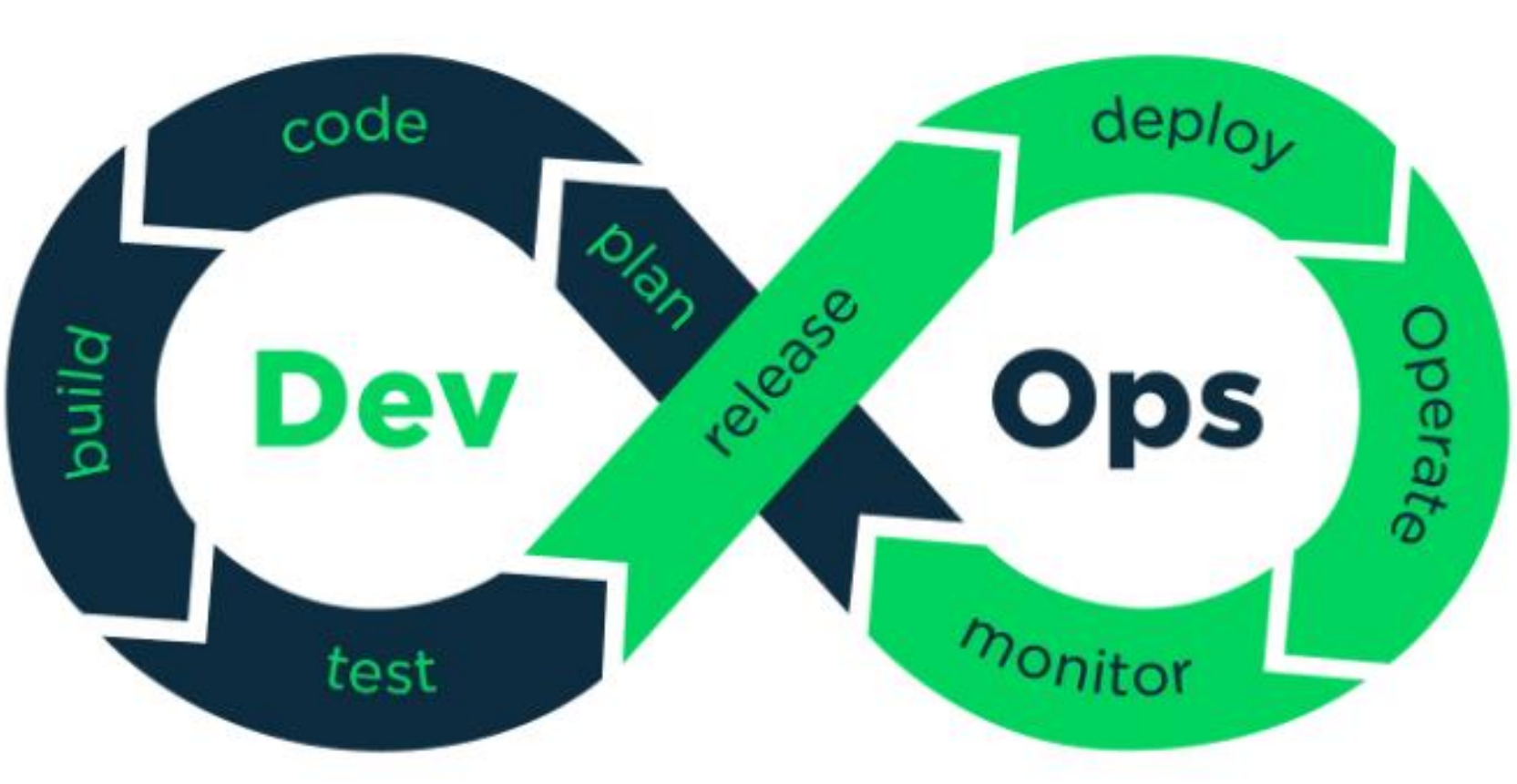
Mô Hình Linh Hoạt (Agile)



Mô Hình Linh Hoạt (Agile)

- **Ưu điểm:**
  - ❑ Thích ứng nhanh với thay đổi
  - ❑ Sản phẩm có thể kiểm thử sớm
  - ❑ Khách hàng tham gia xuyên suốt
- **Nhược điểm:**
  - ❑ Cần đội nhóm có kỷ luật cao
  - ❑ Khó ước lượng chính xác chi phí, thời gian
- **Phù hợp:**
  - ❑ Dự án yêu cầu linh hoạt, thay đổi liên tục
  - ❑ Startup, phần mềm theo yêu cầu khách hàng

Mô Hình DevOps



Mô Hình DevOps

- **Ưu điểm:**
  - Rút ngắn thời gian đưa sản phẩm ra thị trường
  - Phát hiện lỗi sớm, cải thiện chất lượng phần mềm
  - Phản hồi nhanh, cập nhật liên tục
- **Nhược điểm:**
  - Đòi hỏi hệ thống hạ tầng mạnh
  - Yêu cầu đội ngũ thành thạo công cụ DevOps
- **Phù hợp:**
  - Dự án yêu cầu triển khai nhanh, liên tục
  - Các hệ thống, nền tảng dịch vụ lớn

So Sánh Các Mô Hình

Tiêu chí	Waterfall	Iterative	Agile	DevOps
Đặc điểm chính	Tuần tự, từng bước, cố định	Phát triển lặp, cải tiến qua từng vòng	Linh hoạt, phản hồi nhanh, chia thành sprint	Liên tục phát triển - tích hợp - triển khai
Tiến trình	Một chiều, không lùi bước	Lặp đi lặp lại, mỗi vòng hoàn thiện dần	Nhiều sprint ngắn liên tục	Chu trình CI/CD không ngừng
Tính linh hoạt	Rất thấp	Trung bình	Rất cao	Rất cao
Thời điểm kiểm thử	Sau khi hoàn thành toàn bộ phát triển	Sau mỗi vòng lặp	Trong từng sprint	Tích hợp kiểm thử liên tục
Yêu cầu thay đổi	Khó thay đổi khi đang làm	Có thể thích nghi sau mỗi vòng	Thay đổi nhanh chóng theo yêu cầu mới	Thay đổi và cập nhật liên tục
Phù hợp cho	Dự án nhỏ, yêu cầu rõ ràng từ đầu	Dự án vừa, cần hoàn thiện dần	Dự án thay đổi liên tục, yêu cầu linh hoạt	Dự án cần release liên tục, vận hành tự động
Rủi ro	Rủi ro cao nếu yêu cầu sai	Rủi ro trung bình, có thể điều chỉnh	Rủi ro thấp nhờ thích ứng nhanh	Rủi ro thấp nhờ tự động hóa
Ví dụ thực tế	Phát triển hệ thống ngân hàng cũ	Phát triển hệ thống ERP	Phát triển app mobile, startup	SaaS platforms như Netflix, Amazon