

Các mô hình Waterfall, Iterative, Agile, DevOps

1. Waterfall (Mô hình Thác nước)

Mô tả Waterfall là mô hình phát triển phần mềm tuyến tính, từng giai đoạn thực hiện nối tiếp:

1. **Yêu cầu (Requirements):** Thu thập, phân tích và ghi chép đầy đủ các yêu cầu của khách hàng.
2. **Thiết kế (Design):** Lên kiến trúc hệ thống, thiết kế chi tiết các module.
3. **Triển khai (Implementation):** Lập trình, xây dựng mã nguồn.
4. **Kiểm thử (Testing):** Thực hiện kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống.
5. **Bảo trì (Maintenance):** Sửa lỗi, cập nhật và nâng cấp khi phần mềm vào vận hành.

Đặc điểm

- Tuyến tính, không quay lại giai đoạn trước.
- Tài liệu đầy đủ, rõ ràng.
- Phản hồi khách hàng chỉ có sau khi hoàn thành toàn bộ.

Ưu điểm

- Dễ quản lý, dễ theo dõi tiến độ.
- Từng bước rõ ràng, phù hợp với dự án có yêu cầu cố định, ít thay đổi.

Nhược điểm

- Khó linh hoạt khi cần thay đổi yêu cầu.
- Khách hàng phải chờ đến cuối mới thấy sản phẩm thực tế, dễ dẫn đến không đáp ứng đúng nhu cầu.

2. Iterative (Mô hình Lặp)

Mô tả Iterative là mô hình phát triển qua nhiều vòng lặp (iteration). Mỗi vòng lặp bao gồm:

- Phân tích → Thiết kế → Thực hiện → Kiểm thử → Đánh giá

Sau mỗi vòng, phần mềm được cải tiến, mở rộng dựa trên phản hồi của khách hàng và nhóm.

Đặc điểm

- Có thể quay lại điều chỉnh trong các vòng sau.
- Mỗi bản dựng (build) sau một vòng lặp là phần mềm có thể chạy được.

Ưu điểm

- Giảm rủi ro nhờ phát hiện sớm lỗi, vấn đề về yêu cầu.
- Khách hàng có thể xem và đánh giá sản phẩm ngay từ các vòng đầu.

Nhược điểm

- Yêu cầu quản lý phức tạp hơn so với Waterfall.
- Có thể tốn kém về thời gian, nguồn lực nếu không kiểm soát tốt phạm vi mỗi iteration.

3. Agile (Mô hình Linh hoạt)

Mô tả Agile là phương pháp phát triển linh hoạt, nhấn mạnh vào:

- **Sprint** (chu kỳ ngắn, thường 1–4 tuần): mỗi Sprint tạo ra một phiên bản phần mềm có tính năng hoàn chỉnh.
- **Tương tác liên tục** trong nhóm và với khách hàng.
- **Phản hồi nhanh**, sẵn sàng thay đổi yêu cầu.

Đặc điểm

- Út tiên phần mềm chạy được hơn tài liệu chi tiết.
- Làm việc theo nhóm tự tổ chức, tự chịu trách nhiệm.
- Cập nhật backlog (danh sách yêu cầu) liên tục.

Ưu điểm

- Rất linh hoạt, thích ứng nhanh với thay đổi.
- Tăng tính minh bạch, sự hài lòng của khách hàng.
- Tốc độ phát triển nhanh, có giá trị thực tế sau mỗi Sprint.

Nhược điểm

- Cần nhóm phát triển có kỷ luật, kinh nghiệm.
- Dự án không rõ ràng, yêu cầu cố định có thể gây mất ổn định.

4. DevOps

Mô tả DevOps là văn hóa và tập hợp thực hành kết hợp giữa **Development** (phát triển) và **Operations** (vận hành) nhằm:

- Tự động hóa quá trình xây dựng, kiểm thử, triển khai (CI/CD).
- Giám sát, đo lường và phản hồi nhanh khi xảy ra sự cố.
- Rút ngắn chu kỳ phát hành, nâng cao chất lượng và độ ổn định.

Đặc điểm

- Sử dụng các công cụ tự động hóa (Jenkins, GitLab CI, CircleCI...).
- Kết hợp giám sát (monitoring) và logging trong suốt vòng đời phần mềm.
- Văn hóa “Infrastructure as Code” (hệ thống được quản lý như mã nguồn).

Ưu điểm

- Phát hành nhanh, thường xuyên và an toàn.
- Giảm thiểu lỗi nhờ tự động hóa kiểm thử và triển khai.
- Tăng cường hợp tác, chia sẻ trách nhiệm giữa phát triển và vận hành.

Nhược điểm

- Yêu cầu đầu tư hạ tầng, công cụ phức tạp.
- Cần thay đổi văn hóa tổ chức, đào tạo nhân sự.

Bảng so sánh tổng hợp

Tiêu chí	Waterfall	Iterative	Agile	DevOps
Cách tiếp cận	Tuyến tính	Lặp lại nhiều vòng	Lặp nhanh, linh hoạt	Tích hợp Dev & Ops, CI/CD
Giai đoạn phát triển	Cố định, tuần tự	Lặp lại: phân tích → thiết kế → ...	Chia nhỏ theo Sprint	Liên tục: build → test → deploy → monitor
Khả năng thay đổi	Rất thấp	Trung bình	Rất cao	Cao nhờ tự động hóa và phản hồi nhanh
Phản hồi khách hàng	Cuối dự án	Sau mỗi vòng lặp	Liên tục mỗi Sprint	Liên tục từ hệ thống và người dùng
Tài liệu	Đầy đủ, chi tiết	Đủ dùng, cập nhật thường xuyên	Ưu tiên code & tính năng	Tài liệu kỹ thuật + scripts tự động hóa
Tốc độ phát triển	Chậm	Trung bình	Nhanh	Rất nhanh
Phù hợp với dự án	Yêu cầu cố định, ít thay đổi	Phát triển dần, yêu cầu có thể thay đổi	Yêu cầu biến động, khách hàng tham gia liên tục	Hệ thống lớn, yêu cầu phát hành liên tục
Yêu cầu nhóm	Trung bình	Trung bình	Cao, tự tổ chức	Cao, kỹ năng Dev & Ops đa dạng
Khả năng mở rộng	Thấp	Trung bình	Trung bình	Rất cao

