

BÁO CÁO ASSIGNMENT.3

Bùi Huỳnh Tây

MSSV: 24521589

04/05/2025

Cấu trúc dữ liệu và giải thuật
IT003.P21.CTTN

BÁO CÁO: CÁC MÔ HÌNH TRONG VÒNG ĐỜI PHÁT TRIỂN PHẦN MỀM (SDLC)

Mục lục

1	Giới thiệu về SDLC	3
2	Các mô hình phát triển chính của SDLC	3
3	Bảng so sánh các mô hình phát triển phần mềm	4
4	Kết luận	4

1. Giới thiệu về SDLC

Software Development Life Cycle (SDLC) – Vòng đời phát triển phần mềm – là một quy trình có hệ thống nhằm đảm bảo chất lượng và hiệu quả trong việc phát triển phần mềm. SDLC giúp kiểm soát từng giai đoạn của việc xây dựng phần mềm, từ khi hình thành ý tưởng cho đến khi triển khai và bảo trì.

6 Phases of the Software Development Life Cycle



Hình 1: Mô hình vòng đời phát triển phần mềm (SDLC)

2. Các mô hình phát triển chính của SDLC

Từ SDLC cơ bản, người ta đã phát triển ra nhiều mô hình khác nhau nhằm tối ưu hóa hiệu suất và phù hợp với từng loại dự án. Bốn mô hình tiêu biểu nhất bao gồm:

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

Mỗi mô hình mang đặc trưng riêng và được áp dụng tùy theo yêu cầu, độ phức tạp, quy mô dự án và mức độ thay đổi yêu cầu.

3. Bảng so sánh các mô hình phát triển phần mềm

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, lặp 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	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 hóa
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

4. Kết luận

Mỗi mô hình SDLC đều có điểm mạnh và điểm yếu riêng. Do đó, việc lựa chọn mô hình nào phù hợp phải dựa vào tính chất dự án, mức độ rõ ràng của yêu cầu, thời gian, ngân sách và khả năng của đội ngũ phát triển.

- **Waterfall** phù hợp với các dự án truyền thống, có yêu cầu cố định.
- **Iterative** thích hợp khi cần cải tiến dần theo từng giai đoạn.
- **Agile** là lựa chọn hàng đầu cho các dự án có yêu cầu thay đổi liên tục.
- **DevOps** tối ưu cho hệ thống yêu cầu triển khai và cập nhật thường xuyên.

Trên đây là bài báo cáo của em, em cảm ơn thầy/cô đã đọc ạ!