

TỔNG QUAN CHƯƠNG 10: QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

1. Lý thuyết và thực tế trong phát triển phần mềm

- Lý tưởng nhất, phần mềm được phát triển theo mô hình thác nước (waterfall) với các bước: Yêu cầu → Phân tích → Thiết kế → Triển khai → Cài đặt.
- Tuy nhiên, thực tế có nhiều yếu tố làm mô hình này không khả thi:
 - **Lỗi của con người:** Lập trình viên có thể mắc sai sót trong mọi giai đoạn, buộc phải quay lại sửa lỗi.
 - **Thay đổi yêu cầu (Moving-target problem):** Khách hàng thường xuyên thay đổi yêu cầu trong quá trình phát triển.

2. Quá trình lặp và gia tăng trong phát triển phần mềm

- **Lặp (Iteration):** Một thành phần phần mềm được cải tiến qua nhiều phiên bản.
- **Gia tăng (Incrementation):** Phần mềm được xây dựng theo từng phần nhỏ và mở rộng dần.
- **Kết hợp lặp và gia tăng:**
 - Gia tăng giúp bổ sung tính năng.
 - Lặp giúp nâng cao chất lượng phần mềm.
- Quy trình này giúp thích ứng với thay đổi và kiểm soát chất lượng tốt hơn so với mô hình tuyến tính.

3. Quy trình hợp nhất (Unified Process - UP)

- Là phương pháp phát triển phần mềm có thể tùy chỉnh theo từng dự án.
- Sử dụng **UML (Unified Modeling Language)** để mô hình hóa phần mềm.
- **Quy trình hướng đối tượng:**
 - Mô hình hóa phần mềm qua nhiều phiên bản UML.
 - Lặp lại và mở rộng dần các mô hình đến khi hoàn thiện.

4. Tổng quan về các quy trình làm việc (Workflows)

- **Yêu cầu (Requirements):** Xác định chính xác nhu cầu của khách hàng.
- **Phân tích (Analysis):** Làm rõ yêu cầu để dễ dàng phát triển phần mềm.
- **Thiết kế (Design):** Xây dựng kiến trúc phần mềm dựa trên phân tích.
- **Triển khai (Implementation):** Viết mã nguồn để thực hiện thiết kế.

- **Kiểm thử (Testing):** Đảm bảo phần mềm hoạt động đúng và không có lỗi.
- **Lập kế hoạch & tài liệu hóa:** Xuyên suốt vòng đời phần mềm.

5. Làm việc nhóm trong phát triển phần mềm

- Các dự án phần mềm thường quá phức tạp để một cá nhân có thể tự thực hiện.
- Mỗi nhóm có thể phụ trách một workflow cụ thể (yêu cầu, thiết kế, kiểm thử...).

6. Phân tích chi phí - lợi ích trong phần mềm

- **Chi phí phát triển phần mềm** gồm:
 - Chi phí nhân lực.
 - Phần cứng, phần mềm hỗ trợ.
 - Chi phí vận hành, bảo trì.
- **Lợi ích:** Xác định giá trị kinh doanh mà phần mềm mang lại.
- So sánh giữa chi phí và lợi ích để quyết định hướng phát triển tối ưu.

7. Đo lường và đánh giá hiệu suất phần mềm (Metrics)

- **5 chỉ số quan trọng:**
 1. Kích thước (số dòng code hoặc module).
 2. Chi phí (tính bằng tiền).
 3. Thời gian thực hiện (tháng).
 4. Nỗ lực (số nhân công/tháng).
 5. Chất lượng (số lỗi phát hiện).
- Các chỉ số giúp theo dõi tiến độ và phát hiện vấn đề sớm.

8. Công cụ hỗ trợ phát triển phần mềm (CASE - Computer-Aided Software Engineering)

- **Công cụ CASE:** Hỗ trợ một số tác vụ cụ thể như vẽ UML, quản lý mã nguồn.
- **Workbench:** Tập hợp các công cụ CASE phục vụ một hoặc hai hoạt động.
- **Môi trường CASE:** Hệ thống hỗ trợ toàn bộ quy trình phát triển phần mềm.

9. Quản lý phiên bản và cấu hình

- **Phiên bản (Version):** Khi một thành phần phần mềm thay đổi, cần lưu lại phiên bản cũ.
- **Cấu hình (Configuration):** Tập hợp các phiên bản của các thành phần phần mềm.

- **Công cụ kiểm soát phiên bản:** Giúp quản lý lịch sử thay đổi mã nguồn.
- **Công cụ kiểm soát cấu hình:** Giúp theo dõi trạng thái của phần mềm trong từng giai đoạn.

10. Kiểm thử phần mềm

- **Lỗi (Fault):** Lập trình viên mắc sai sót khi viết mã.
- **Thất bại (Failure):** Phần mềm hoạt động sai do lỗi.
- **Chất lượng phần mềm:** Mức độ đáp ứng yêu cầu.
- **Hai loại kiểm thử:**
 - **Kiểm thử thực thi (Execution-based):** Chạy phần mềm với các bộ kiểm thử.
 - **Kiểm thử không thực thi (Non-execution-based):** Đọc mã nguồn, tài liệu để phát hiện lỗi.

11. Nguyên tắc thiết kế phần mềm: Tính mô-đun

- **Mô-đun (Module):** Đơn vị nhỏ trong phần mềm.
- **Nguyên tắc thiết kế tốt:**
 - **Giảm ghép nối (Low Coupling):** Giảm sự phụ thuộc giữa các mô-đun.
 - **Tăng kết gắn (High Cohesion):** Mỗi mô-đun chỉ thực hiện một nhiệm vụ cụ thể.
 - **Che giấu thông tin (Information Hiding):** Hạn chế quyền truy cập vào chi tiết bên trong mô-đun.

12. Tái sử dụng phần mềm (Software Reuse)

- Giảm thời gian và chi phí phát triển bằng cách sử dụng lại mã nguồn, tài liệu, thiết kế.
- Áp dụng trong nhiều lĩnh vực như phần mềm nhúng, hệ thống doanh nghiệp, AI.

13. Quản lý dự án phần mềm

- **3 thành phần chính của kế hoạch quản lý dự án:**
 1. **Công việc cần làm (Work):** Các nhiệm vụ cụ thể trong dự án.
 2. **Nguồn lực (Resources):** Nhân lực, phần cứng, phần mềm.
 3. **Ngân sách (Money):** Chi phí cần thiết cho dự án.
- **Lập kế hoạch theo IEEE 1058:**
 - Quản lý tài nguyên, thời gian, rủi ro.
 - Kiểm soát chất lượng dự án.