

# Review chapter 1

1. Định nghĩa Công nghệ phần mềm: Phần mềm là tập hợp các hướng dẫn, chương trình được viết để máy tính thực thi nhằm thực hiện các chức năng hoặc nhiệm vụ cụ thể.  
Gồm 3 loại phần mềm: Hệ thống, ứng dụng và nhúng.  
Mục tiêu: Tạo ra phần mềm đúng yêu cầu khách hàng, đảm bảo phần mềm có thể bảo trì và nâng cấp dễ dàng, giảm thiểu thời gian và chi phí phát triển.
  2. Quy trình phát triển phần mềm: Lấy yêu cầu, thiết kế, lập trình, kiểm thử, triển khai, bảo trì.
  3. Phạm vi phát triển phần mềm: Bảo trì, bù đắp chi phí tương đối của các giai đoạn trong vòng đời phần mềm
  4. Các vấn đề thường gặp khi phát triển phần mềm: Vấn đề vượt chi phí, vấn đề trễ thời gian, vấn đề vẫn còn lỗi sau khi bàn giao. Tùy thuộc vào vấn đề gặp phải gây ra hậu quả xác định.
  5. Các lỗi dẫn đến các vấn đề trong phát triển phần mềm: Không đồng bộ, thiếu quyết đoán, yêu cầu kém, phạm vi mở rộng không kiểm soát, thiếu giao tiếp
-

# Review chapter 2

## Phần 1: Giới thiệu về

### các mô hình quy trình phần mềm

- **Mô hình thác nước (Waterfall Model):** Mô hình này là một mô hình tuyến tính, tuần tự, trong đó mỗi giai đoạn của quy trình phát triển phần mềm phải được hoàn thành trước khi giai đoạn tiếp theo bắt đầu. Mô hình này đơn giản và dễ hiểu, nhưng nó không linh hoạt và khó thích ứng với các thay đổi.
- **Mô hình chữ V (V-Model):** Mô hình này là một biến thể của mô hình thác nước, trong đó giai đoạn kiểm thử được nhấn mạnh hơn. Mô hình chữ V liên kết mỗi giai đoạn phát triển với một giai đoạn kiểm thử tương ứng.
- **Mô hình nguyên mẫu (Prototyping Model):** Mô hình này cho phép tạo ra một nguyên mẫu của phần mềm trước khi phát triển phần mềm hoàn chỉnh. Nguyên mẫu này được sử dụng để thu thập phản hồi từ người dùng và để xác định các yêu cầu của phần mềm.
- **Mô hình xoắn ốc (Spiral Model):** Mô hình này là một mô hình lặp, trong đó mỗi vòng lặp bao gồm các giai đoạn lập kế hoạch, phân tích rủi ro, phát triển và đánh giá. Mô hình xoắn ốc phù hợp cho các dự án lớn và phức tạp, trong đó rủi ro là một yếu tố quan trọng.

## Phần 2: Các khía cạnh thực tế của phát triển phần mềm

- **Thay đổi yêu cầu:** Yêu cầu của phần mềm có thể thay đổi trong quá trình phát triển. Điều này có thể là do người dùng thay đổi nhu cầu của họ, hoặc do các yếu tố bên ngoài khác.
- **Rủi ro:** Phát triển phần mềm là một hoạt động rủi ro. Có nhiều yếu tố có thể dẫn đến thất bại của dự án, chẳng hạn như yêu cầu không rõ ràng, ước tính chi phí và thời gian không chính xác, và các vấn đề kỹ thuật.
- **Quản lý dự án:** Quản lý dự án hiệu quả là rất quan trọng để đảm bảo thành công của dự án phát triển phần mềm. Điều này bao gồm việc lập kế hoạch, tổ chức, lãnh đạo và kiểm soát dự án.

## Phần 3: Các phương pháp phát triển phần mềm linh hoạt (Agile)

- **Lập trình cực đoan (Extreme Programming - XP):** XP là một phương pháp phát triển phần mềm linh hoạt, trong đó nhấn mạnh vào việc giao tiếp, phản hồi và đơn giản.

- **Scrum:** Scrum là một khung làm việc linh hoạt, trong đó dự án được chia thành các sprint ngắn. Mỗi sprint có một mục tiêu cụ thể và một khoảng thời gian cố định.
- **Kanban:** Kanban là một phương pháp quản lý công việc trực quan, trong đó công việc được chia thành các giai đoạn và được hiển thị trên một bảng Kanban.

## **Kết luận**

Chương 2 cung cấp một cái nhìn tổng quan về các mô hình quy trình phần mềm khác nhau và các khía cạnh thực tế của phát triển phần mềm. Việc lựa chọn mô hình quy trình phần mềm phù hợp và áp dụng các phương pháp quản lý dự án hiệu quả là rất quan trọng để đảm bảo thành công của dự án phát triển phần mềm.

# Review chapter 3

## Tổng quan về nội dung chương

Chương "The Software Process" tập trung vào việc giới thiệu và giải thích quy trình phát triển phần mềm, nhấn mạnh tầm quan trọng của việc quản lý và cải tiến quy trình này. Nội dung được chia thành nhiều phần, bao gồm:

1. **Khái niệm cơ bản về quy trình phần mềm:** Quy trình phần mềm được định nghĩa là cách thức tổ chức các hoạt động, kỹ thuật, công cụ, và con người để sản xuất phần mềm. Chương chỉ ra sự khác biệt giữa các tổ chức trong cách tiếp cận quy trình, từ việc coi phần mềm là tự tài liệu hóa (self-documenting) đến các tổ chức yêu cầu tài liệu chi tiết và kiểm tra kỹ lưỡng.
2. **Quy trình Thống nhất (Unified Process):** Đây là một phương pháp phát triển phần mềm hướng đối tượng (object-oriented) chính, được mô tả chi tiết với năm luồng công việc cốt lõi (core workflows): yêu cầu (requirements), phân tích (analysis), thiết kế (design), thực hiện (implementation), và kiểm thử (testing). Chương cũng giải thích bốn pha (phases) của Unified Process: khởi tạo (inception), chi tiết hóa (elaboration), xây dựng (construction), và chuyển giao (transition).
3. **Tính chất lặp lại và tăng dần (Iteration and Incrementation):** Phương pháp này nhấn mạnh rằng phát triển phần mềm thường được thực hiện qua nhiều lần lặp lại và tăng dần, sử dụng Ngôn ngữ Mô hình Thống nhất (UML - Unified Modeling Language) để mô phỏng các khía cạnh của sản phẩm phần mềm, giúp giao tiếp hiệu quả hơn giữa các thành viên đội ngũ phát triển.
4. **Cải tiến quy trình phần mềm:** Chương đề cập đến các sáng kiến cải tiến quy trình phần mềm ở cấp quốc gia và quốc tế, như Mô hình Độ trưởng thành Khả năng (Capability Maturity Model - CMM) của Viện Kỹ thuật Phần mềm (Software Engineering Institute - SEI), cùng các tiêu chuẩn như ISO 9000 và ISO/IEC 15504 (SPICE).
5. **Các thách thức và giải pháp:** Chương nêu bật các vấn đề thường gặp trong quy trình phát triển phần mềm, như sự mơ hồ, mâu thuẫn, hoặc thiếu sót trong yêu cầu, cũng như việc bảo trì sau khi giao hàng (postdelivery maintenance) và việc nghỉ hưu của phần mềm (retirement).

## Ý nghĩa và giá trị

- **Tầm quan trọng của quy trình có cấu trúc:** Quy trình phần mềm không chỉ là một chuỗi các bước đơn giản, mà cần được quản lý chặt chẽ để đảm bảo chất lượng, giảm thiểu rủi ro, và tối ưu hóa hiệu suất. Unified Process, với tính lặp lại và tăng

dẫn, là một phương pháp linh hoạt và hiệu quả để đối phó với sự phức tạp của các dự án phần mềm lớn.

- **Vai trò của UML và mô hình hóa:** Sử dụng UML giúp cải thiện giao tiếp và giảm thiểu hiểu lầm giữa các bên liên quan, đặc biệt là giữa khách hàng và đội phát triển, nhờ vào khả năng thể hiện trực quan các khái niệm phức tạp.
- **Cải tiến liên tục:** Các mô hình như CMM và tiêu chuẩn ISO nhấn mạnh rằng cải tiến quy trình phần mềm là cần thiết để nâng cao chất lượng sản phẩm, giảm chi phí, và tăng năng suất. Dữ liệu từ các tổ chức như Hughes Aircraft, Raytheon, và Motorola cho thấy lợi ích kinh tế rõ rệt khi nâng cao mức độ trưởng thành của quy trình.
- **Xử lý các thách thức thực tế:** Chương giải quyết các vấn đề thực tế như sự thay đổi yêu cầu của khách hàng (mowing-target problem), thiếu tài liệu, và rủi ro kỹ thuật, cung cấp nền tảng để tìm kiếm giải pháp trong các phần khác của sách.

## **Điểm mạnh và hạn chế**

- **Điểm mạnh:**
  - Nội dung được trình bày chi tiết, có cấu trúc rõ ràng, với các ví dụ thực tế (như lỗi phần mềm trong các dự án không gian) và các "Just in Case You Wanted to Know" Box giúp làm sáng tỏ các khái niệm phức tạp.
  - Chương cung cấp cái nhìn toàn diện về cả lý thuyết (như các luồng công việc và pha của Unified Process) và thực tiễn (như cải tiến quy trình qua CMM và ISO).
  - Các tài liệu tham khảo phong phú, bao gồm các nghiên cứu và tiêu chuẩn quốc tế, giúp người đọc mở rộng kiến thức.
- **Hạn chế:**
  - Một số khái niệm, như chi tiết về UML, chỉ được đề cập sơ qua và yêu cầu tham khảo các chương khác, có thể khiến người đọc mới cảm thấy thiếu liên mạch.
  - Nội dung tập trung nhiều vào các dự án phần mềm lớn, có thể không hoàn toàn phù hợp với các dự án nhỏ hoặc vừa, dù tác giả đã đề cập đến việc điều chỉnh Unified Process cho quy mô khác nhau.

## **Kết luận**

Chương The Software Process cung cấp nền tảng lý thuyết vững chắc về Unified Process và các mô hình cải tiến mà còn nhấn mạnh giá trị thực tiễn của việc áp dụng các phương pháp này trong thế giới thực.