

ỦY BAN NHÂN DÂN THÀNH PHỐ CẦN THƠ
TRƯỜNG CAO ĐẲNG KINH TẾ - KỸ THUẬT CẦN THƠ



GIÁO TRÌNH

CÔNG NGHỆ PHẦN MỀM
NGÀNH CÔNG NGHỆ THÔNG TIN
TRÌNH ĐỘ: CAO ĐẲNG

*Ban hành kèm theo Quyết định số: /QĐ-... ngày.....tháng....năm
của.....*

Cần Thơ, ngày ... tháng ... năm 2019

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình, nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Công nghệ phần mềm là một trong những lĩnh vực then chốt phục vụ cuộc cách mạng công nghiệp 4.0 thông qua việc khuyến khích ứng dụng công nghệ thông tin trong đời sống và công việc. Với vai trò và mục đích trọng tâm nhằm giúp tạo ra các sản phẩm phần mềm có chất lượng cao, môn học Công nghệ phần mềm sẽ cung cấp kiến thức, hỗ trợ rèn luyện kỹ năng và giúp phát huy tư duy cần thiết để người học có khả năng thực hiện được vai trò và mục đích trên.

Giáo trình Công nghệ phần mềm trình bày các nội dung cơ bản của tất cả giai đoạn trong quy trình phát triển phần mềm qua 11 chương, bao gồm các mô hình, kỹ thuật phổ biến, đặc điểm các công cụ, đặc trưng của các nghiệp vụ, những thuận lợi và thách thức...trong các công việc có liên quan. Cuối mỗi chương, các bài tập và câu hỏi thảo luận sẽ là dàn ý giúp người học hệ thống, mở rộng kiến thức và vận dụng chúng bằng các kỹ năng thích hợp.

Giáo trình này được biên soạn nhằm mục đích giảng dạy môn Công nghệ phần mềm ở bậc cao đẳng, phù hợp với độc giả đã có kiến thức cơ sở về công nghệ thông tin.

Nhóm tác giả chân thành cảm ơn những ý kiến đóng góp thiết thực nhằm xây dựng và hoàn thiện giáo trình, góp phần vào việc nâng cao chất lượng đào tạo nguồn nhân lực công nghệ thông tin của nhà trường nói riêng và của địa phương nói chung.

Cần Thơ, ngày...tháng... năm 2019

Tham gia biên soạn

1. Chủ biên - TS. Phạm Phi Giang
2. Ths. Lê Thị Phương Phi
3. Ths. Châu Lê Salin

MỤC LỤC

Nội dung	Trang
Danh mục bảng	i
Danh mục hình	ii
Danh mục từ viết tắt.....	iv
 Chương 1 Tổng quan CNPM	 1
1.1 Tổng quan	
1.2 Khái niệm	
1.3 Một số thuật ngữ liên quan	2
1.4 Vai trò của phần mềm	3
Bài tập	4
Chủ đề thảo luận	
Chương 2 Quy trình phát triển phần mềm	6
2.1 Khái quát	
2.1.1 Khái niệm	
2.1.2 Các giai đoạn trong quy trình	
2.2 Nội dung cơ bản các giai đoạn	
2.2.1 Xác định yêu cầu	
2.2.2 Phân tích và đặc tả	8
2.2.3 Thiết kế.....	9
2.2.4 Cài đặt	10
2.2.5 Kiểm thử	
2.2.6 Bảo trì.....	11
Bài tập	
Chủ đề thảo luận.....	12
Chương 3 Các mô hình chu trình phát triển phần mềm	13
3.1 Mô hình thác nước	
3.2 Mô hình bản mẫu	15
3.3 Mô hình tăng trưởng	17
3.4 Mô hình xoắn ốc.....	18

3.5 Mô hình quy trình hợp nhất (RUP)	21
3.5 Phát triển hướng đối tượng	25
Bài tập	26
Chủ đề thảo luận	
Chương 4 Nhân lực & Công cụ CASE	27
4.1 Giới thiệu	
4.2 Tiếp cận nhóm làm việc	
4.2.1 Phát triển ứng dụng	28
4.2.2 Hỗ trợ ứng dụng	
4.2.3 Một số chuyên ngành kỹ thuật	29
4.2.4 Một số lĩnh vực khác	
4.3 Công cụ CASE	31
Bài tập	32
Chủ đề thảo luận.....	33
Chương 5 Thu thập yêu cầu	34
5.1 Giới thiệu	
5.2 Khởi động khảo sát	
5.3 Kỹ thuật khảo sát cơ bản	35
5.3.1 Phỏng vấn	
5.3.2 Bảng câu hỏi.....	38
5.3.3 Quan sát	
Bài tập	39
Chủ đề thảo luận	
Chương 6 Phân tích & đặc tả yêu cầu	41
6.1 Giới thiệu	
6.2 Phân tích yêu cầu	
6.2.1 Yêu cầu chức năng	
6.2.2 Yêu cầu phi chức năng	
6.3 Đặc tả yêu cầu	42
6.4 Xác nhận đặc tả	45
Bài tập	46
Chương 7 Thiết kế	47

7.1 Giới thiệu	
7.2 Thiết kế kiến trúc	
7.3 Thiết kế hệ thống.....	51
7.3.1 Mô hình Thực thể - Quan hệ	
7.3.2 Mô hình hướng đối tượng với UML	53
7.4 Thiết kế giao diện.....	57
Bài tập	60
Chủ đề thảo luận	
Chương 8 Cài đặt	61
8.1 Giới thiệu	
8.2 Công cụ	
8.3 Giải thuật & dữ liệu.....	63
8.4 Kỹ thuật lập trình	64
8.5 Nhân lực	65
Bài tập	66
Chủ đề thảo luận	
Chương 9 Kiểm thử	67
9.1 Giới thiệu	
9.2 Kiểm thử đáp ứng yêu cầu	
9.3 Kiểm thử lỗi	68
9.3.1 Kỹ thuật kiểm thử hộp đen.....	69
9.3.2 Kỹ thuật kiểm thử hộp trắng	
9.4 Kiểm thử giai đoạn	
9.4.1 Giai đoạn cài đặt	
9.4.2 Giai đoạn bàn giao	70
9.4.3 Kiểm thử từ người dùng.....	71
Bài tập	72
Chủ đề thảo luận	
Chương 10 Bàn giao & bảo trì.....	74
10.1 Giới thiệu	
10.2 Bàn giao	
10.2.1 Tiếp cận sản phẩm mới	

10.2.2 Kế hoạch bàn giao	75
10.2.3 Một số lưu ý trong công đoạn bàn giao	76
10.3 Bảo trì	
Bài tập	79
Chủ đề thảo luận	
Chương 11 Quản lý dự án phần mềm	80
11.1 Giới thiệu	
11.2 Quản lý rủi ro	81
11.3 Quản lý nhân lực	83
11.4 Nhóm làm việc	85
11.5 Lập kế hoạch dự án	
11.6 PERT	86
11.7 Gantt	88
Bài tập	89
Chủ đề thảo luận.....	91
Tài liệu tham khảo	92
Phụ lục 1 Một số tiêu chuẩn quốc tế	93
Phụ lục 2 Khung đặc tả yêu cầu phần mềm (IEEE).....	96
Phụ lục 3 Tài liệu đặc tả yêu cầu Phần mềm bán hàng.....	104
Phụ lục 4 Các tiêu chuẩn liên quan đến đảm bảo chất lượng và kiểm thử phần mềm ISO/IEC/IEEE 29119.....	116

DANH MỤC BẢNG

	Trang
Bảng 2.1 Ví dụ về việc tóm tắt quá trình thiết kế, cài đặt một ứng dụng web	8
Bảng 5.1 Tóm tắt 2 dạng câu hỏi đóng và mở.....	30
Bảng 5.2 Tóm tắt các bước cơ bản của một buổi phỏng vấn	30
Bảng 6.1 Ví dụ về việc đặc tả yêu cầu Phần mềm điều khiển pha café	37
Bảng 6.2 Ví dụ đặc tả yêu cầu dạng ngôn ngữ tự nhiên có cấu trúc	38
Bảng 6.3 Tóm tắt hai kiểu đặc tả hình thức và phi hình thức.....	39
Bảng 7.1 Ví dụ một thiết kế kiến trúc từ mức độ tổng quát đến chuyên biệt cấp 1 .	42
Bảng 7.2 Kiến trúc một phần điều khiển người máy đóng gói kiện hàng.....	42
Bảng 8.1 Phân nhóm các công cụ hỗ trợ lập trình.....	56
Bảng 11.1 Những rủi ro phổ biến	75

DANH MỤC HÌNH

	Trang
Hình 1.1. Phần mềm hệ thống và phần mềm ứng dụng	3
Hình 2.1. Các nội dung cơ bản trong giai đoạn thiết kế	9
Hình 3.1. Mô hình thác nước	13
Hình 3.2. Mô hình bản mẫu	16
Hình 3.3. Mô hình tăng trưởng	17
Hình 3.4. Mô hình xoắn ốc	19
Hình 3.5. Một trường hợp áp dụng mô hình xoắn ốc	20
Hình 3.6. Mô hình quy trình hợp nhất (RUP)	24
Hình 6.1. Một trường hợp trình bày bằng đồ họa trong đặc tả yêu cầu trong phần mềm quản lý thông tin bệnh viện	44
Hình 7.1. Kiến trúc một phần mềm dạng Client – Server	48
Hình 7.2. Kiến trúc một mô-đun điều khiển người máy đóng gói kiện hàng	49
Hình 7.3. Kiến trúc phần mềm thông tin thư viện dạng phân lớp	50
Hình 7.4. Kiến trúc phần mềm hệ thống máy rút tiền ATM dạng hướng ứng dụng ..	50
Hình 7.5. Một trường hợp áp dụng mô hình Thực thể - Quan hệ	51
Hình 7.6. Tổng hợp các bảng dữ liệu dạng quan hệ	52
Hình 7.7. Ví dụ về một lớp đối tượng xe hơi (car class)	54
Hình 7.8. Một ví dụ về lớp kế thừa	55
Hình 7.9. Một mối quan hệ trong sơ đồ lớp	55
Hình 7.10. Sơ đồ trường hợp sử dụng thể hiện tương tác giữa khách hàng và hệ thống khi sử dụng chức năng tìm kiếm	56
Hình 7.11. Sơ đồ trình tự thể hiện sự cộng tác thông qua chuỗi thông điệp giữa khách hàng, người bán vé và hệ thống	56
Hình 7.12. Sơ đồ hoạt động diễn giải một quy trình nướng bánh.	57
Hình 7.13. Giao diện chính của một phần mềm quản lý nhà hàng	58
Hình 7.14. Giao diện chính của một phần mềm quản lý nhà hàng dạng sơ đồ	58

Hình 7.15. Giao diện chính của một phần mềm quản lý dạng đối tượng.....	59
Hình 7.16. Giao diện chính của một phần mềm quản lý dịch vụ Nails.....	59
Hình 9.1. Quy trình cơ bản của một đợt kiểm thử.....	68
Hình 9.2. Một quy trình cơ bản của kiểm thử chấp nhận	72
Hình 10.1. Biểu đồ tỉ lệ chi phí trong quy trình phát triển phần mềm	77
Hình 10.2. Biểu đồ so sánh chi phí phát triển và bảo trì	77
Hình 10.3. Biểu đồ tỉ lệ chi phí trong giai đoạn bảo trì.....	78
Hình 11.1. Tóm tắt quy trình quản lý rủi ro	83
Hình 11.2. Một trường hợp áp dụng PERT	88
Hình 11.3. Một trường hợp áp dụng biểu đồ Gantt	89

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ đầy đủ
SQA	Software Quality Assurance
SPMP	Software Project Management Plan
CMS	Content Management System
UML	Unified Modeling Language
GUI	Graphical User Interface
SWIG	Simplified Wrapper and Interface Generator
PERT	Program Evaluation Review Technique

CHƯƠNG 1. TỔNG QUAN CÔNG NGHỆ PHẦN MỀM

Chương 1 giới thiệu những nội dung tổng quan về bối cảnh hình thành lĩnh vực công nghệ phần mềm và trình bày các khái niệm, thuật ngữ cơ bản có liên quan trong lĩnh vực này. Qua đó, người học có thể hiểu và giải thích được vai trò công nghệ phần mềm trong ngành công nghệ thông tin nói riêng và trong cuộc sống nói chung.

1.1 Tổng quan

Thế hệ máy tính điện tử đầu tiên ra đời vào năm 1946, được thương mại hóa vào những năm 1950 và từ đó những chương trình máy tính bắt đầu phát triển. Sau mười năm phát triển, ngành nghề xây dựng và kinh doanh các chương trình máy tính đã xuất hiện nhiều vấn đề gây khó khăn và làm sụt giảm toàn bộ quá trình sản xuất cũng như tiêu thụ phần mềm, điển hình như:

- Về chất lượng phần mềm: không đáp ứng yêu cầu thực tế, khó sử dụng, không tin cậy, khó bảo trì, khách hàng không hài lòng...
- Về việc quản lý quy trình sản xuất: kế hoạch không cụ thể, không tính đúng giá giá thành, không đúng tiến độ, biến động các chi phí phát triển, vấn đề nhân sự...
- Về mặt pháp lý: hợp đồng không rõ ràng, không chặt chẽ...
- Về mặt nhân lực: tay nghề, trình độ không đồng đều, khả năng làm việc nhóm chưa tốt...
- Về quản lý tiêu chuẩn: thiếu tiêu chuẩn đánh giá sản phẩm, thiếu quy trình quản lý...

Năm 1986 hội thảo của NATO về công nghệ phần mềm đã đưa ra thuật ngữ “Khủng hoảng phần mềm” để mô tả ngắn gọn tình hình ngành công nghiệp sản xuất phần mềm ở thời điểm đó.

1.2 Khái niệm

Thuật ngữ “Phần mềm - Software” được sử dụng lần đầu tiên năm 1935; tuy nhiên, sau 1946 thuật ngữ này trở nên phổ biến hơn để chỉ những chương trình được lưu trữ và thực thi trên máy tính.

Thuật ngữ “Công nghệ phần mềm – Software engineering” được sử dụng chính thức năm 1965 để chỉ việc áp dụng các công nghệ và kỹ thuật phục vụ cho quá trình

phát triển các sản phẩm phần mềm. Ở một góc nhìn khác, công nghệ phần mềm có liên quan đến việc nghiên cứu và đề xuất các phương pháp và công cụ trong việc xây dựng các phần mềm đạt chất lượng và quá trình sản xuất phần mềm có hiệu quả kinh tế.

Chẳng hạn, chất lượng của một phần mềm cần thỏa mãn ít nhất hai yêu cầu:

- Phần mềm tạo ra phải đáp ứng được yêu cầu người sử dụng
- Phần mềm phải đạt được các tiêu chuẩn chất lượng được áp dụng

Một quá trình sản xuất phần mềm hiệu quả cần đáp ứng tối thiểu hai yêu cầu:

- Giá thành phải nằm trong giới hạn dự kiến
- Tiến độ xây dựng phần mềm phải đảm bảo theo kế hoạch

Hiện nay, thuật ngữ “Phần mềm” được sử dụng để khái quát tất cả các ứng dụng, chương trình thực thi trên các thiết bị điện tử như các thiết bị di động (điện thoại - phones, thiết bị mang theo – wearable devices,...), các thiết bị điều khiển (máy tự động).

IEEE định nghĩa thuật ngữ “Công nghệ phần mềm” là việc áp dụng các phương pháp có tính hệ thống (systematic), tính kiểm soát (disciplined) và tính lượng hóa (quantifiable) vào việc phát triển, vận hành và bảo trì phần mềm; đồng thời, “Công nghệ phần mềm” là một ngành học nhằm tìm hiểu và nghiên cứu các phương pháp đó.

1.3 Một số thuật ngữ liên quan

Thuật ngữ “Chương trình máy tính - computer programs” được dùng phổ biến cùng với thuật ngữ Phần mềm, nhưng nó nhấn mạnh hơn về các tập hợp lệnh bên trong một phần mềm. Các lệnh trong một chương trình máy tính sẽ điều khiển máy tính thực hiện một hoặc nhiều chức năng cụ thể.

Thuật ngữ “Phần cứng máy tính – computer hardware” được dùng để chỉ tất cả các bộ phận vật chất hoặc các thiết bị của máy tính (Ví dụ: bảng mạch, ổ đĩa cứng, bàn phím, chuột...) và các thiết bị liên quan đến máy tính (Ví dụ: dây dẫn điện, dây mạng...).

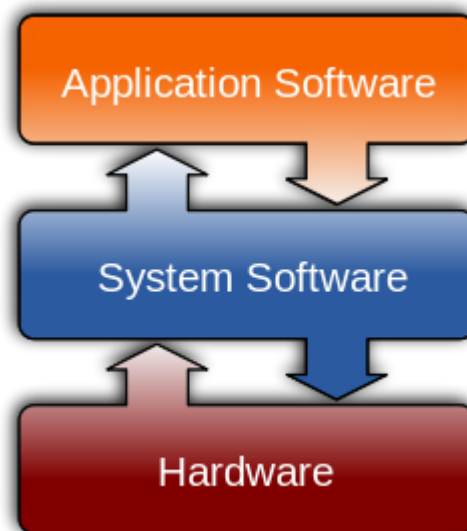
Thuật ngữ “Đảm bảo chất lượng phần mềm - Software Quality Assurance” xác định các hoạt động nhằm đảm bảo sản phẩm được xây dựng đúng theo hợp đồng đặt hàng (phù hợp với yêu cầu sử dụng và các tiêu chuẩn). Công việc này phải thực hiện

đúng chức năng ngay từ đầu của quy trình và hoạt động trong tất cả giai đoạn phát triển phần mềm. Vai trò của công việc đảm bảo chất lượng có thể kết thúc khi việc kiểm tra phiên bản cuối cùng được khách hàng hài lòng.

Như vậy, công nghệ phần mềm là ngành học nhằm tìm hiểu và nghiên cứu cách thức sản xuất phần mềm sao cho thỏa mãn nhu cầu của khách hàng (hoặc yêu cầu được đặt ra), phần mềm không có lỗi, được bàn giao đúng thời hạn và có chi phí phù hợp.

1.4 Vai trò của phần mềm

Phần mềm chính là cầu nối giữa con người và máy tính. Trong cuộc sống hiện đại, máy tính nói riêng và các thiết bị điện tử nói chung hiện diện ở mọi nơi, mọi lĩnh vực; chúng phục vụ nhu cầu của con người và do con người điều khiển. Thật sự, phần mềm chính là phương tiện giúp con người thực hiện việc điều khiển đó. Có hai dạng điều khiển gồm điều khiển hệ thống (system) và điều khiển ứng dụng (application) (Hình 1.1).



Hình 1.1 Phần mềm hệ thống và phần mềm ứng dụng

(https://en.wikibooks.org/wiki/Alevel_Computing/AQA/Paper_2/Fundamentals_of_computer_systems/Classification_of_software)

Phần mềm hệ thống (System software) có vai trò chủ yếu vận hành các thiết bị phần cứng và tạo ra một môi trường để các phần mềm ứng dụng hoạt động. Con người sử dụng phần mềm hệ thống để vận hành các thiết bị điện tử và phần cứng máy tính (Ví dụ: các hệ điều hành - Operating systems Windows, Ubuntu, iOS, Android). Trong

khi đó, phần mềm ứng dụng (application software) tập trung phục vụ các nhu cầu cụ thể cho con người hoặc môi trường ngoài máy tính. Chẳng hạn như,

- Giải trí: phần mềm xem phim, nghe nhạc, chụp hình, trò chơi điện tử,...
- Giáo dục: phần mềm dạy học, lớp học điện tử (Elearning class),...
- Y tế: phần mềm quản lý bệnh viện, phần mềm theo dõi, đánh giá sức khỏe,...
- Truyền thông: hệ thống trang tin điện tử (websites), mạng xã hội trực tuyến (Facebook, Zalo),...
- Thương mại: phần mềm quản trị kinh doanh, websites thương mại điện tử,...

Bài tập

1. Tìm hiểu, trình bày khái niệm và cho ví dụ của những thuật ngữ gần nghĩa với thuật ngữ Phần mềm. (Gợi ý: chương trình máy tính – computer programs, ứng dụng – computer based application, ứng dụng web ...)
2. Thuật ngữ Khủng hoảng phần mềm vào những năm 1986 có ý nghĩa gì?
3. Trình bày sự cần thiết ngành học công nghệ phần mềm trong nền công nghiệp sản xuất phần mềm.

Chủ đề thảo luận

1. Sự phát triển của phần cứng máy tính hoặc các thiết bị kỹ thuật (tốc độ xử lý, tốc độ truyền dẫn thông tin, dung lượng lưu trữ, ...) có tác động hoặc ảnh hưởng như thế nào đến sự phát triển phần mềm?
2. Các đặc điểm sau đây có ảnh hưởng như thế nào đến người sử dụng các phần mềm hiện nay
 - Giao diện đẹp: màn hình độ phân giải cao, màu sắc hài hòa
 - Dễ sử dụng: thao tác đơn giản, dễ thực hiện
 - Tốc độ xử lý nhanh: các chức năng hoạt động mượt mà
 - Giá thành hợp lý: miễn phí hoặc giá hợp lý
 - Thương hiệu: nổi tiếng hoặc có nhiều người sử dụng

- An toàn: rủi ro về mất an toàn thông tin, vi-rút máy tính, chương trình quảng cáo

...

(Gợi ý: có thể bổ sung các tiêu chí khác, đánh giá với điểm số và xếp hạng)

CHƯƠNG 2. QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

Chương 2 giới thiệu khái quát quy trình phát triển phần mềm và trình bày các nội dung cơ bản của toàn bộ những giai đoạn trong quy trình. Dựa vào nội dung này, người học có thể hiểu và giải thích được vai trò và mục đích cốt lõi của từng giai đoạn và việc kết hợp các giai đoạn trong tổng thể một quy trình phát triển phần mềm.

2.1 Khái quát

2.1.1 Khái niệm

“Quy trình phát triển phần mềm” (Software development processes) là tập hợp các công việc có trật tự nhằm sản xuất phần mềm có chất lượng [1]. Nội dung chính của quy trình này gồm các mô hình vòng đời phần mềm (Software life-cycle models), các công cụ CASE (CASE tools), và yếu tố con người (Individuals) vừa được xem xét ở góc nhìn kỹ thuật và cũng được xem xét ở phương diện quản lý.

Vì vậy, quy trình phát triển phần mềm cụ thể ở các tổ chức, các doanh nghiệp khác nhau thông thường sẽ không giống nhau (Ví dụ: khác về khả năng tài chính, khác về đặc điểm nhân sự, khác về đặc thù nghề nghiệp).

2.1.2 Các giai đoạn trong quy trình

Một quy trình phát triển phần mềm cơ bản bao gồm các giai đoạn sau:

- Xác định yêu cầu (Requirement phase)
- Phân tích và đặc tả (Specification phase)
- Thiết kế (Design phase)
- Cài đặt (Implementation phase)
- Kiểm thử (Testing phase)
- Bảo trì (Maintenance phase)

Tùy vào phương pháp và kỹ thuật áp dụng quy trình phát triển phần mềm mà các giai đoạn của quy trình có thể được thực hiện tuần tự hoặc đồng thời, có thể được thực hiện dứt điểm kiểu “cuốn chiếu” hoặc thực hiện lặp lại.

2.2 Nội dung cơ bản của các giai đoạn

2.2.1 Xác định yêu cầu

Đây là giai đoạn tìm hiểu và xác định những yêu cầu của khách hàng đối với dự án hoặc sản phẩm mềm mà họ cần.

Quy trình phát triển phần mềm bắt đầu khi khách hàng liên hệ với công ty phần mềm và đặt vấn đề về một sản phẩm phần mềm phù hợp với yêu cầu của khách hàng và có giá cả hợp lý.

❖ **Tình huống**

Đầu tiên, khách hàng có thể phát họa một sản phẩm theo ý muốn của mình. Tuy nhiên, theo quan điểm của người phụ trách kỹ thuật của công ty, mô tả này không rõ ràng, chưa hợp lý, không thể hoặc khó có thể xây dựng phần mềm đúng như ý muốn ban đầu của khách hàng.

❖ **Nhận định**

Người phụ trách kỹ thuật chưa làm tốt nhiệm vụ ở giai đoạn này do khách hàng đã mô tả không đúng theo nhu cầu của họ hoặc người phụ trách chưa hiểu đúng các yêu cầu đó.

Công việc cần làm là xác định yêu cầu của khách hàng về sản phẩm, về thời gian, về chi phí.., mà chưa phải là thời điểm để xác định giải pháp hoặc xác định tính khả thi của sản phẩm.

Vấn đề khó khăn trong giai đoạn này là tình huống khách hàng không biết cụ thể sản phẩm mà họ cần. Từ đó, khách hàng không thể trình bày rõ ràng yêu cầu về sản phẩm, hoặc các mô tả về yêu cầu của họ sẽ dễ thay đổi không nhất quán.

❖ **Giải pháp**

- Thu thập yêu cầu với bản mẫu

Bản mẫu phần mềm (Software prototype): Là các phần mềm mẫu hoặc giả lập có kết hợp một hoặc nhiều chức năng của sản phẩm cuối cùng và tất nhiên không phải là một phần mềm hoàn chỉnh thực thụ. Việc thay đổi các chức năng của bản mẫu phần mềm được thực hiện đơn giản và nhanh chóng, giúp khách hàng, người sử dụng phần mềm tương lai xác định cụ thể, chính xác yêu cầu của mình.

- Viết tài liệu

Viết tài liệu là công việc gần như xuyên suốt trong tất cả các giai đoạn của quy trình phát triển phần mềm. Đây là một yêu cầu bắt buộc nhằm đảm bảo tính chuyên nghiệp trong khi triển khai kỹ thuật và vận hành quản lý.

Trường hợp có bản mẫu nhanh, một biên bản giữa công ty với khách hàng và người sử dụng về việc xác định cụ thể tất cả các yêu cầu có trong bản mẫu. Trường hợp không có bản mẫu nhanh, một tài liệu mô tả yêu cầu (bằng chữ viết, hình ảnh, số liệu,...không bao gồm phần mềm mẫu) được khách hàng và người sử dụng xác nhận.

2.2.2 Phân tích và đặc tả

Dựa vào những yêu cầu của khách hàng, nhóm phân tích sẽ phân tích và đặc tả những yêu cầu đó theo cách thức và quy chuẩn đặc thù của chuyên môn.

Sau khi tài liệu mô tả các yêu cầu của khách hàng và người sử dụng được thông qua, một tài liệu đặc tả các chức năng của sản phẩm được viết, tài liệu này phải mô tả chi tiết tất cả những gì sản phẩm cần có và các ràng buộc cần có.

Việc đặc tả thường bao gồm những dữ liệu đầu vào (input) của sản phẩm hoặc của một chức năng, tác vụ và những kết quả đầu ra (output).

Ví dụ: Chức năng tính lương hàng tháng trong một công ty, đầu vào bao gồm định mức chi trả cho mỗi nhân viên, thông tin cá nhân của nhân viên, các hệ số phụ cấp, mức đóng bảo hiểm xã hội,..., và đầu ra là số tiền lương cụ thể nhận được đối với từng nhân viên.

Yêu cầu của đặc tả:

- Đầy đủ: thể hiện mọi yêu cầu của khách hàng
- Không mâu thuẫn: không có ý nghĩa đối lập hoặc phủ định lẫn nhau
- Cụ thể (không dùng từ nhiều nghĩa)
- Dễ rà soát: có thể sử dụng đặc tả để đối chiếu trở lại với yêu cầu của khách hàng.

Nếu đặc tả được trình bày đúng đắn so với yêu cầu của khách thì nhóm đảm bảo chất lượng phần mềm sẽ ít gặp khó khăn. Nếu đặc tả có nội dung không đối chiếu được với yêu cầu khách hàng, sẽ có hai trường hợp. Một là, trong trường hợp có sử dụng bản mẫu thì những suy luận hoặc phán đoán liên quan của đặc tả phải phù hợp

với bản mẫu. Hai là, khi không có bản mẫu, cần sự xác minh trở lại đối với khách hàng.

Ví dụ: Đặc tả cho yêu cầu in ấn chỉ bao gồm việc in ấn có kết nối trực tiếp cá nhân (không chia sẻ máy in). Tài liệu đặc tả có thể phát sinh chức năng in ấn qua hệ thống mạng cục bộ. Bản mẫu được sử dụng để đối chiếu xem có chức năng in ấn qua mạng cục bộ hay không.

Sau khi toàn bộ việc đặc tả được hoàn thành và thông qua, kế hoạch quản lý dự án phần mềm (Software Project Management Plan - SPMP) sẽ được hình thành. Đây là khâu then chốt của dự án nhằm xác định các yếu tố thời gian, chi phí, nhân lực...và các yếu tố về kỹ thuật, công nghệ.

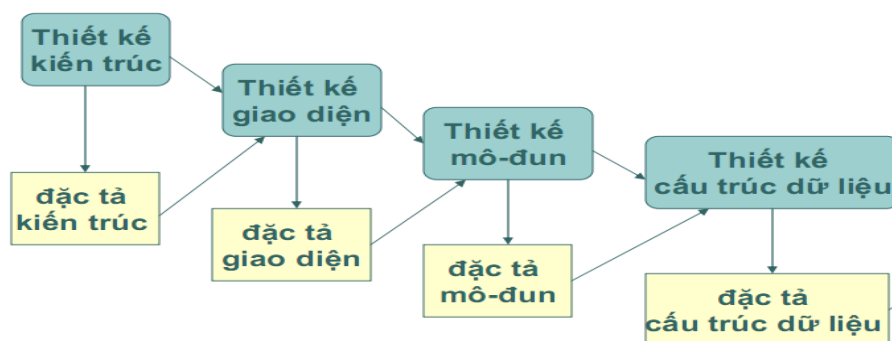
2.2.3 Thiết kế

Thiết kế là việc chuyển đổi từ tài liệu đặc tả yêu cầu thành tài liệu mô tả các chức năng hoặc công việc có cấu trúc hợp lý để chuẩn bị cho giai đoạn cài đặt.

Thiết kế có thể bao gồm các nội dung sau:

- Thiết kế kiến trúc: các thành phần cấu thành hệ thống;
- Thiết kế giao diện: các khung nhìn, thành phần trực quan;
- Thiết kế thành phần: các mô-đun, chức năng, giải thuật;
- Thiết kế cơ sở dữ liệu: mô hình, các ràng buộc, cấu trúc dữ liệu.

Sản phẩm của giai đoạn này là các tài liệu thiết kế (Hình 2.1).



Hình 2.1. Các nội dung cơ bản trong giai đoạn thiết kế

2.2.4 Cài đặt

Cài đặt là giai đoạn tạo ra sản phẩm phần mềm dựa theo tài liệu thiết kế, bên cạnh các tài liệu đã có khác. Các công cụ hỗ trợ lập trình, ngôn ngữ lập trình, công cụ thiết kế giao diện,... sẽ được sử dụng để tạo ra các giao diện, hình thành các chức năng trong từng mô-đun hoặc thành phần của một phần mềm tổng thể.

Ví dụ: Tóm tắt quá trình thiết kế, cài đặt một ứng dụng Web cho thương mại điện tử

Bảng 2.1 Ví dụ về việc tóm tắt quá trình thiết kế, cài đặt một ứng dụng web

<div>Thành phần</div> <div>Giai đoạn</div>	Kiến trúc	Giao diện	Mô-đun/Chức năng	Cơ sở dữ liệu/cấu trúc dữ liệu	Thuật toán
Thiết kế	Thiết kế (server, client)	Thiết kế (trang chủ, trang liên kết)	Thiết kế (tĩnh, động)	Bảng dữ liệu, ràng buộc...	Mã hóa nội dung giao dịch
Cài đặt	Cài đặt (back-end, front-end)	Cài đặt (banner, menu, button, frame,...)	Cập nhật nội dung, quản trị người dùng, an ninh mạng	Cập nhật dữ liệu, thủ tục – hàm truy vấn	Mã hóa thông tin thẻ tín dụng

2.2.5 Kiểm thử

Kiểm thử là việc kiểm tra tính đúng đắn các chức năng của phần mềm đối với tài liệu thiết kế, yêu cầu của khách hàng và yêu cầu về kỹ thuật... Vì vậy, giai đoạn này giữ vai trò cực kỳ quan trọng và không thể thiếu để bảo đảm chất lượng phần mềm, cũng như chất lượng của cả dự án sản xuất phần mềm. Làm tốt công đoạn này sẽ giúp giảm thiểu hoặc hạn chế các rủi ro trong các công việc vận hành và bảo trì phần mềm về sau.

Các vấn đề thường gặp trong giai đoạn kiểm tra:

- Thời gian và chi phí đối với các hệ thống, phần mềm phức tạp
- Quy trình phát triển phần mềm luôn trải qua nhiều giai đoạn và từng giai đoạn sử dụng một dạng thông tin đặc trưng. Ý nghĩa của thông tin được chuyển đổi qua các giai đoạn là yếu tố chính làm cho hoạt động rà soát, đối chiếu sản phẩm cuối cùng so với yêu cầu ban đầu gặp khó khăn.
- Yêu cầu cụ thể của việc kiểm tra không được quy định rõ ràng và đầy đủ
- Kiểm tra mang tính chất không tuyệt đối do tính đúng đắn của phần mềm còn phụ thuộc vào thời gian.

2.2.6 Bảo trì

Bảo trì là công việc bảo dưỡng và duy trì phần mềm hoạt động ổn định và đúng đắn theo thời gian. Đây là hoạt động phức tạp nhưng cần thiết nhằm đảm bảo phần mềm vẫn phù hợp với yêu cầu của người sử dụng và các nền tảng cơ sở kỹ thuật. Giai đoạn cuối cùng này chiếm khá nhiều thời gian trong một dự án phần mềm, vì vậy vẫn còn nhiều tranh luận về việc cách thức tính thời gian và chi phí bảo trì phần mềm trong một dự án. Các hoạt động bảo trì thường gặp bao gồm sửa lỗi mã nguồn, điều chỉnh chức năng phù hợp với thực tế sử dụng, cập nhật các ràng buộc hoặc tiêu chí vận hành phần mềm.

Bài tập

1. Hãy trình bày mối quan hệ giữa quy trình phát triển phần mềm và vòng đời phần mềm.
2. Nhân sự nào là phù hợp cho lần gặp đầu tiên với khách hàng để tiếp nhận yêu cầu? Tại sao?
3. Hãy kể tên ít nhất 3 công cụ hoặc phần mềm có thể sử dụng để tạo bản mẫu nhanh.
4. Giai đoạn thiết kế có thể được bỏ qua hay không? Tại sao?
5. Có phải công việc của giai đoạn cài đặt chỉ là lập trình? Giải thích câu trả lời?

Chủ đề thảo luận

1. Hãy đề xuất bổ sung hoặc bỏ qua một hoặc một vài giai đoạn trong quy trình phát triển phần mềm nêu trên, trình bày lý do việc bổ sung hoặc bỏ qua đó.

2. Trong những yêu cầu của việc đặc tả nêu trên, yêu cầu nào quan trọng nhất? Tại sao?

3. Trong quy trình phát triển phần mềm nêu trên, các giai đoạn nào có thể lặp lại, có thể được kết hợp với nhau? Tại sao?

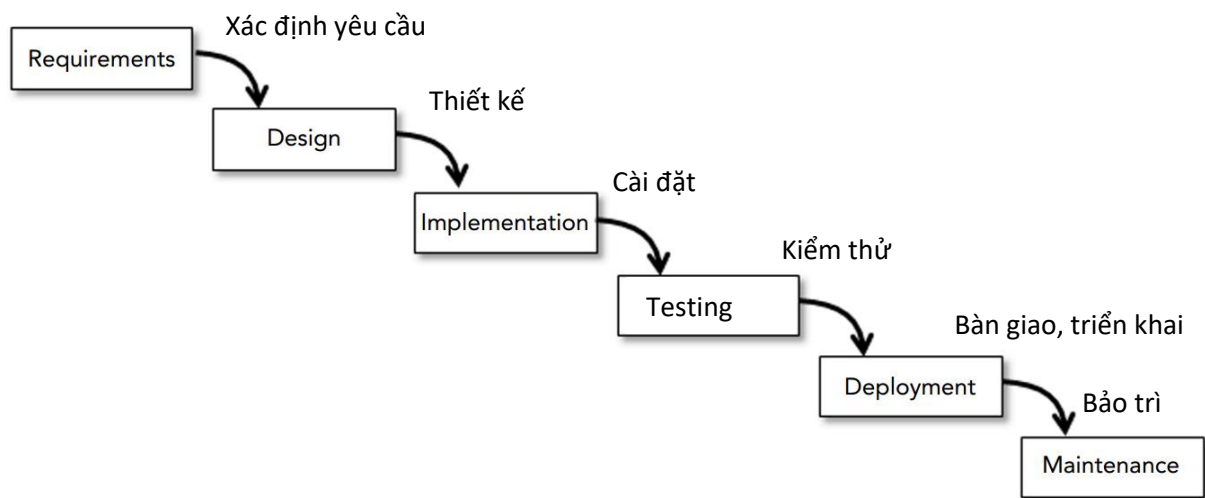
CHƯƠNG 3. CÁC MÔ HÌNH QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

Một sản phẩm phần mềm để được ra đời phải trải qua tất cả các giai đoạn của quy trình phát triển phần mềm (xác định yêu cầu, đặc tả, thiết kế, cài đặt, kiểm tra, bảo trì). Các giai đoạn này có thể được thực hiện theo các thứ tự khác nhau hoặc được kết hợp theo các quy cách khác nhau, chính là mô hình quy trình phát triển phần mềm (software development process models).

Chương này tập trung giới thiệu năm mô hình phát triển phần mềm phổ biến hiện nay, bao gồm mô hình thác nước (waterfall model), bản mẫu (prototyping model), tăng trưởng (incremental model), xoắn ốc (spiral model) và hướng đối tượng (object-oriented model). Kiến thức của chương này giúp người học có thể áp dụng hoặc triển khai từng mô hình thích hợp vào các dự án vừa và nhỏ với mức độ cơ bản.

3.1 Mô hình thác nước (Waterfall model)

Mô hình thác nước lần đầu tiên được đề nghị bởi Royce (1970), với tên gọi của mô hình chỉ ra cách thức thực hiện tuần tự từ giai đoạn trên cùng đến giai đoạn cuối cùng (Hình 3.1).



Hình 3.1 Mô hình thác nước (Stephens, R. 2015, tr.270)

Đặc trưng của mô hình thác nước là một mô hình hướng kế hoạch. Mỗi giai đoạn chỉ được bắt đầu khi giai đoạn trước đó hoàn thành và sản phẩm được nhóm SQA thông qua. Ngoài ra, đến trước giai đoạn bảo trì, thì giữa các giai đoạn đều có vòng lặp. Vòng lặp giữa các giai đoạn nhằm đảm bảo quá trình cập nhật những thay đổi hoặc bổ sung các phát sinh từ yêu cầu của khách hàng hoặc do yếu tố kỹ thuật.

- Trước hết yêu cầu được xác định và kiểm tra bởi khách hàng và các thành viên của nhóm SQA.
- Sau đó, tài liệu đặc tả được xây dựng và được kiểm tra bởi nhóm SQA rồi đưa cho khách hàng xem xét.
- Khi khách hàng chấp nhận tài liệu đặc tả, thì nhóm phát triển tiến hành lập kế hoạch quản lý dự án phần mềm và lập lịch trình chi tiết để triển khai. Nhóm SQA kiểm tra kế hoạch.
- Khi khách hàng tán thành thời gian và thương lượng xong giá thành sản phẩm thì giai đoạn thiết kế bắt đầu. Trong giai đoạn thiết kế, thường dễ phát hiện được lỗi trong tài liệu đặc tả, có thể là:
 - Không đầy đủ: một số đặc điểm của sản phẩm đã chưa được đề cập
 - Mâu thuẫn: có những phát sinh trong tài liệu đặc tả mâu thuẫn nhau
 - Không rõ ràng: tài liệu đặc tả có nhiều hơn một cách hiểu hoặc ý nghĩa

Khi phát hiện được những vấn đề thì cần phải hiệu chỉnh tài liệu đặc tả trước khi tiến hành các bước tiếp theo.

❖ Ưu điểm

Thế mạnh của mô hình thác nước là việc tài liệu luôn đầy đủ và luôn được kiểm tra bởi nhóm SQA. Mô hình này, vì thế có thuận lợi trong việc quản lý tiến độ dự án và phù hợp áp dụng trong các dự án nhỏ.

Xét các tình huống sau:

- Tình huống 1: ông Minh muốn xây một căn nhà và đã thuê một kiến trúc sư thiết kế. Vị này đưa cho ông Minh một xấp vài chục trang tài liệu mô tả ngôi nhà theo thuật ngữ chuyên ngành xây dựng. Ông Minh chỉ hiểu phần bản vẽ bên ngoài căn nhà, còn lại toàn bộ các nội dung khác không hiểu. Nhưng ông Minh vẫn đồng ý tiếp tục triển khai dự án.

- Tình huống 2: Lan mua một cái áo bằng cách đặt hàng qua trực tuyến. Thay vì được xem video hoặc ảnh của chiếc áo, thì Lan chỉ được xem tài liệu mô tả chiếc áo đó, và Lan đã đặt mua trên cơ sở mô tả như vậy.

Mặc dù các trường hợp trên thực tế không phải thường xuyên xảy ra, nhưng nó thể hiện cách xây dựng phần mềm khi dùng mô hình thác nước.

Trong quy trình phát triển phần mềm, tài liệu đặc tả thường rất nhiều nội dung, lại được mô tả với ngôn ngữ kỹ thuật chuyên ngành. Khách hàng thì không có kiến thức chuyên sâu và cũng không có kinh nghiệm khi đọc các tài liệu kiểu như thế. Vì thế, việc không hiểu phần lớn nội dung là đương nhiên. Tuy nhiên, do khách hàng vẫn hiểu một ít hoặc được diễn giải một cách chung chung, khái quát nên vẫn chấp nhận và thông qua các tài liệu này. Với mô hình thác nước, khách hàng chỉ nhìn thấy sản phẩm cụ thể sau khi nó được hoàn tất cài đặt. Điều gì xảy ra nếu khách hàng nói “tôi biết đây là cái tôi yêu cầu, nhưng không phải là cái tôi cần”?

❖ Nhược điểm

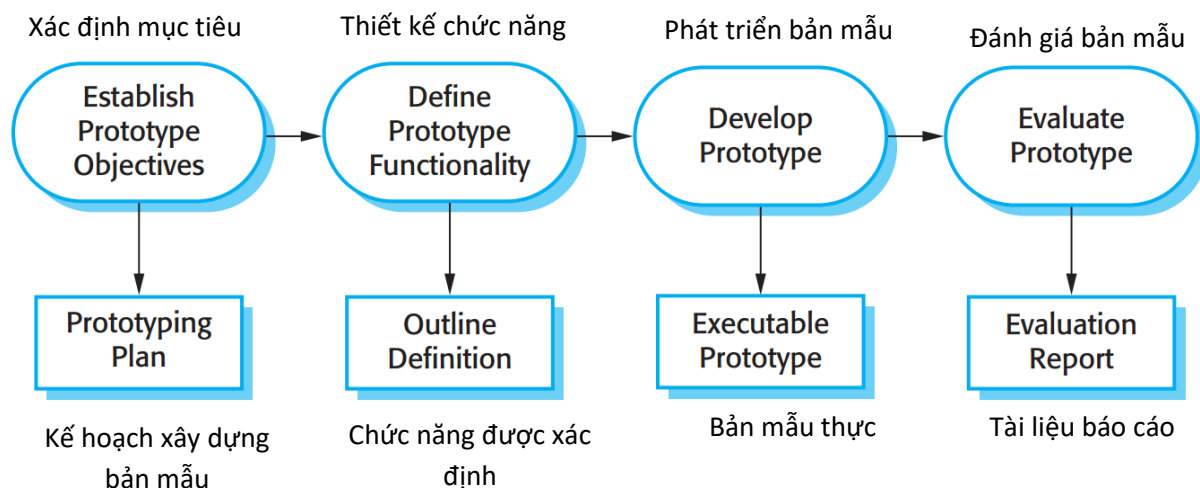
Hạn chế của mô hình thác nước là phụ thuộc quá nhiều vào việc đặc tả và việc ràng buộc cứng nhắc trình tự các giai đoạn. Khi phát hiện vấn đề tại một giai đoạn, rất khó để quay trở lại các giai đoạn trước đó để khắc phục. Vì vậy, có thể dẫn đến xây dựng sản phẩm không thỏa mãn nhu cầu thực sự hoặc thỏa mãn đầy đủ yêu cầu của khách hàng.

3.2 Mô hình bản mẫu (Prototyping model)

Trong nhiều trường hợp gặp hạn chế khi áp dụng các phương pháp lấy thông tin từ khách hàng như phỏng vấn, sử dụng bảng câu hỏi và quan sát, thì hiệu quả của thông tin chỉ đạt mức tổng quát để đặc tả các chức năng của phần mềm. Điều này gây khó khăn cho bước phân tích và đặc tả chi tiết. Phương pháp thu nhận thông tin bằng cách sử dụng mô hình bản mẫu có thể giúp giải quyết vấn đề trên.

Mô hình bản mẫu là việc áp dụng hoặc dựa trên các mô hình mẫu có sẵn trước đó hoặc được xây dựng nhanh, dự kiến đề xuất cho khách hàng và người sử dụng dùng thử và phản hồi. Ví dụ: một số các giao diện phần mềm có các chức năng giả lập (các nút, các menu), giải lập dữ liệu vào, và giả lập kết quả đầu ra. Ưu điểm của mô hình này là tiết kiệm thời gian lấy yêu cầu của khách hàng và giúp họ cảm thấy thú vị hơn trong việc cung cấp thông tin.

Sau khi khách hàng đồng ý rằng bản mẫu thực sự là sản phẩm họ cần thì việc đặc tả và các bước tiếp theo được tiến hành.



Hình 3.2 Mô hình bản mẫu (Sommerville, I. 2011)

Mô hình bản mẫu mang tính tuần tự cao và vòng lặp không được áp dụng. Việc đặc tả hầu như chỉ dựa vào bản mẫu mà ít chú ý đến yêu cầu thực sự của khách hàng. Đây là nhược điểm của mô hình bản mẫu.

Khía cạnh chính của mô hình này là “nhanh chóng” nếu việc áp dụng thành công. Người phát triển mong muốn làm nhanh các bản mẫu để tăng tốc độ phát triển sản phẩm. Điều quan trọng là các bản mẫu được xây dựng và sửa đổi nhanh sao cho phản ánh đúng nhu cầu khách hàng.

❖ Ưu điểm

- Các yêu cầu thường xuyên được kiểm tra và cải thiện. Các bản mẫu trong suốt quá trình phát triển sẽ giúp khách hàng hoặc người sử dụng quan sát được sản phẩm tương lai đáp ứng như thế nào yêu cầu của họ. Qua đó, những phản hồi từ khách hàng hoặc người sử dụng sẽ giúp cải thiện các bản mẫu, hướng đến sản phẩm tương lai cuối cùng thỏa mãn các yêu cầu một cách đầy đủ.

- Việc thống nhất quan điểm của khách hàng và các bên then chốt của dự án được diễn ra dễ dàng và rõ ràng. Điều này giúp giảm thiểu những mâu thuẫn có thể xuất hiện những giai đoạn bàn giao.

- Giai đoạn thiết kế và cài đặt sẽ có nhiều sự lựa chọn, cơ sở đánh giá linh hoạt và toàn diện.

❖ Nhược điểm

- Việc chỉ sử dụng bản mẫu hoặc quá tập trung vào chúng có thể làm giảm hoặc triệt tiêu sự sáng tạo.

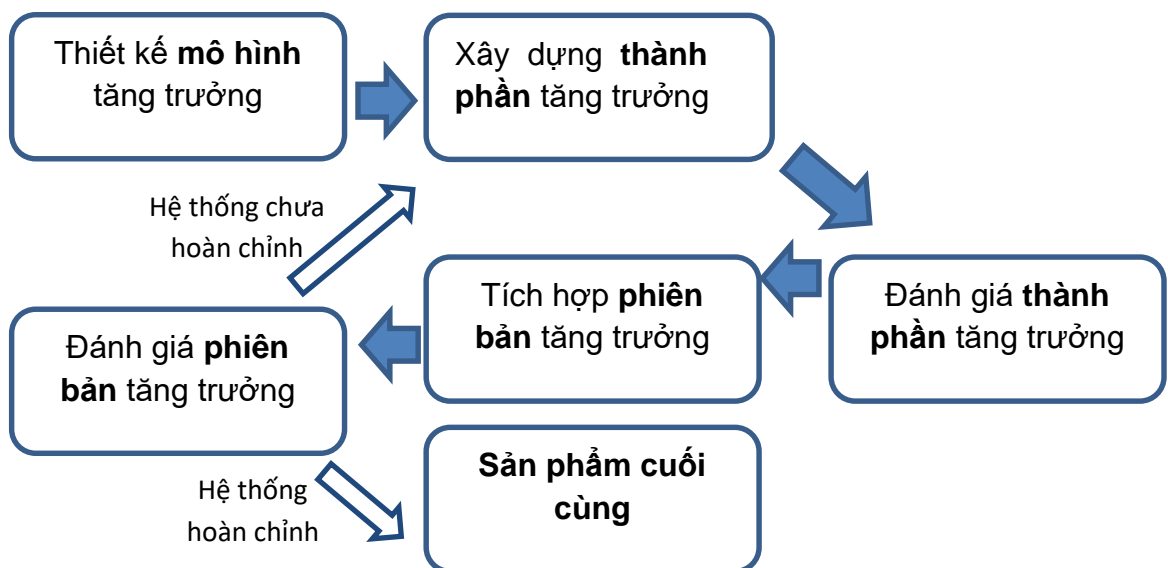
- Khách hàng có thể đánh giá không đúng tiến độ của dự án. Các bản mẫu có thể gây cho khách hàng hiểu rằng sản phẩm cuối cùng đã gần như sẵn sàng.

- Công đoạn cài đặt có thể theo “lối mòn” của mã lệnh trong các bản mẫu.

- Các bản mẫu có thể gây lãng phí thời gian và chi phí nếu được thiết kế không phù hợp với mục tiêu đặt ra.

3.3 Mô hình tăng trưởng (Incremental model)

Trong mô hình tăng trưởng, một phần mềm hoàn chỉnh được xây dựng thông qua nhiều phiên bản phần mềm quá trình. Từng phiên bản sẽ cung cấp một vài chức năng theo yêu cầu của khách hàng, được khách hàng phản ánh những chức năng nào phù hợp hoặc không. Dựa vào những phản ánh đó, trong phiên bản tiếp theo, các chức năng sẽ được hiệu chỉnh hoặc loại bỏ, bổ sung một vài chức năng mới để tiếp tục nhận phản hồi từ khách hàng. Quá trình các phiên bản tiếp theo cũng diễn ra tương tự cho đến khi phiên bản cuối cùng được hoàn thành.



Hình 3.3 Mô hình tăng trưởng (Sommerville, I. 2011)

Đặc trưng của mô hình này là giải quyết một bài toán lớn thông qua quá trình giải quyết các bài toán nhỏ trong đó. Một sản phẩm phần mềm được chia ra thành nhiều chức năng, yêu cầu cụ thể, có thể được xếp hạng theo mức độ quan trọng. Mỗi phiên

bản sẽ tập trung đáp ứng việc giải quyết một vài yêu cầu hoặc chức năng nào đó. Vì vậy, việc phân chia các chức năng và yêu cầu cần thực hiện hợp lý sao cho phù hợp với yếu tố thời gian, nhân lực của dự án. Nếu phân chia quá chi tiết sẽ làm tăng thời gian tích hợp các phiên bản (do quá nhiều phiên bản), còn nếu phân chia quá tổng quát sẽ làm tăng khối lượng công việc trong từng phiên bản (số lượng phiên bản quá ít).

❖ Ưu điểm

- Với mô hình này, tại mỗi giai đoạn khách hàng có được sản phẩm đáp ứng một phần công việc theo yêu cầu. Ngay từ phiên bản ban đầu khách hàng đã có thể thấy được sự ích lợi của sản phẩm, và từng phiên bản của sản phẩm có thể được sử dụng ngay. Điều này phù hợp các dự án lớn khi thời gian và kinh phí phải được tận dụng triệt để.

- Nhận được phản hồi từ khách hàng thường xuyên trong quá trình phát triển các phiên bản. Do đó, khả năng đáp ứng và hiệu chỉnh theo yêu cầu khách hàng là khá cao.

❖ Nhược điểm

Khó khăn cho người quản trị dự án trong việc tính toán tiến độ và phân bổ nguồn nhân lực, đặc biệt với các dự án phức tạp. Các phiên bản phần mềm có thể do nhiều đội phụ trách độc lập, khi này sẽ khó khăn trong quá trình tích hợp các phiên bản với nhau. Tiến độ nếu quá nhanh cũng sẽ không hiệu quả đối với dự án do chi phí đầu tư sẽ bị ảnh hưởng.

3.4 Mô hình xoắn ốc (Spiral model)

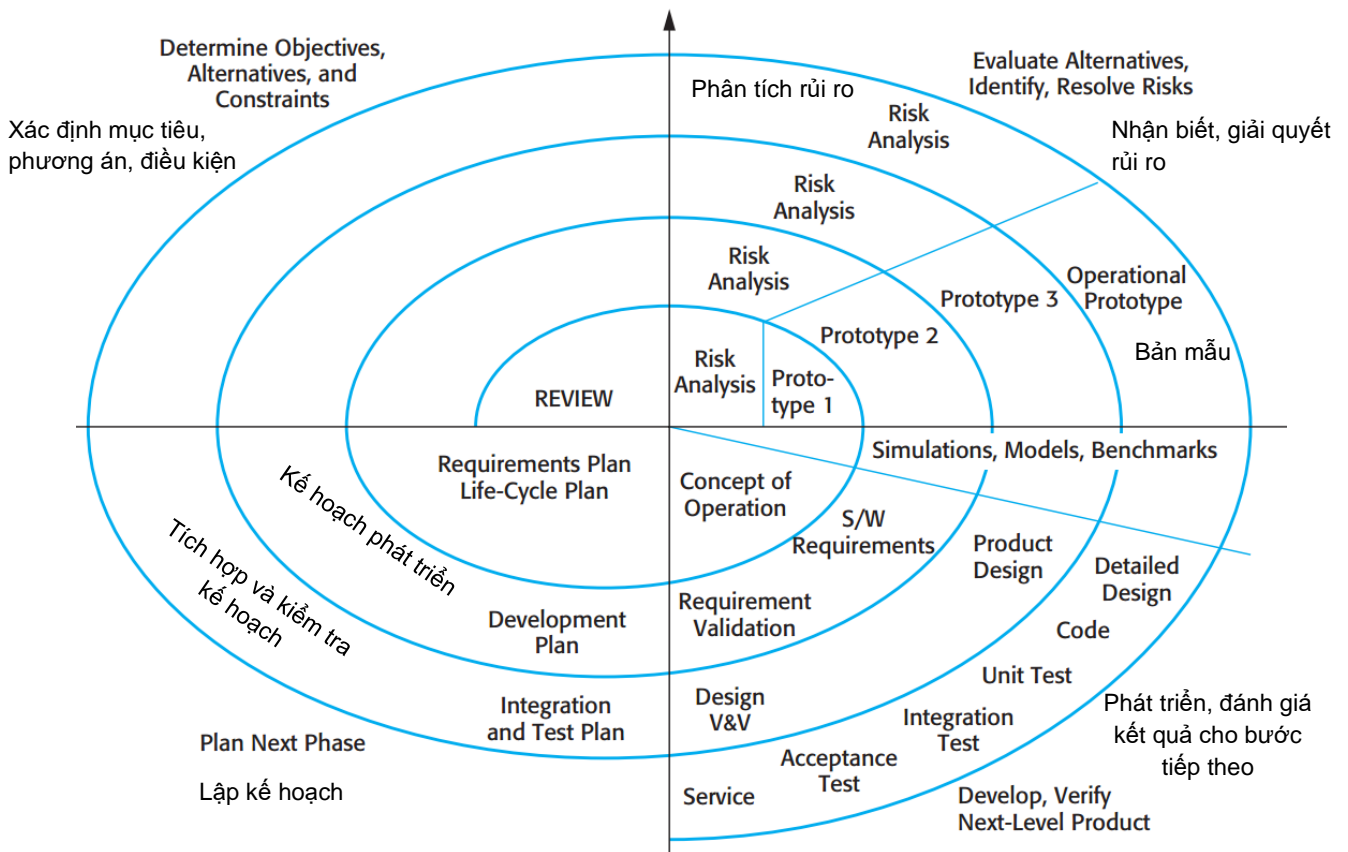
Mô hình xoắn ốc tập trung giảm thiểu rủi ro của quy trình phát triển phần mềm bằng cách sử dụng lặp lại bốn bước, bao gồm: xác định mục tiêu hiện tại và rủi ro tiềm năng, xây dựng bản mẫu đối với rủi ro, đánh giá bản mẫu giải quyết rủi ro và đạt mục tiêu, cập nhật tiến độ và kế hoạch tiếp theo.

Trong suốt quy trình phát triển phần mềm, rủi ro có thể xuất hiện ở bất kỳ giai đoạn nào, chẳng hạn như:

- Xác định yêu cầu: khách hàng có thể không hiểu các yêu cầu về kỹ thuật xử lý, vận hành bên trong phần mềm
- Đặc tả: không đầy đủ yêu cầu của khách hàng
- Thiết kế: không phù hợp với đặc tả

- Cài đặt: không đủ thời gian như kế hoạch
- Kiểm thử: không phát hiện được các lỗi tiềm tàng

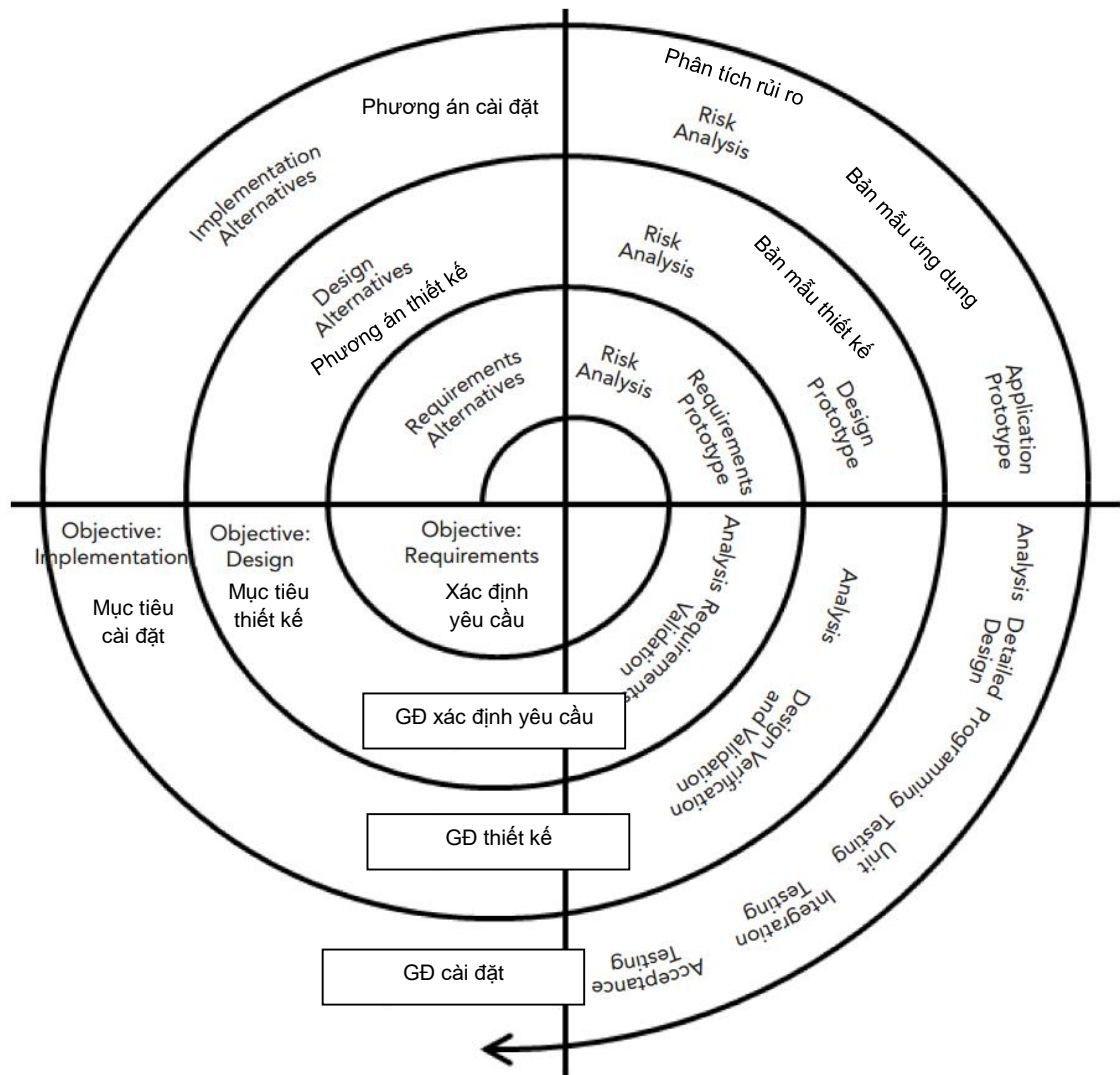
Ngoài ra, các rủi ro liên quan yếu tố nhân sự và kinh phí cũng nên được quan tâm.



Hình 3.4 Mô hình xoắn ốc (©IEEE 1988)

Mỗi vòng xoắn ốc tương ứng một giai đoạn trong quy trình phát triển phần mềm. Trong mỗi giai đoạn, các mục tiêu ngắn hạn được xác định, các rủi ro tiềm năng ngăn cản việc đạt được các mục tiêu cũng được dự đoán. Đặc điểm của mô hình này là việc luôn xem xét và phân tích các rủi ro (risk analysis), kể đến là xây dựng phương án và giải pháp để triệt tiêu hoặc hạn chế rủi ro đó nhằm đạt được mục tiêu của giai đoạn. Tùy vào công việc của mỗi giai đoạn, phương án và giải pháp sẽ khác nhau.

Xem xét ví dụ sau: (Hình 3.5)



Hình 3.5 Một trường hợp áp dụng mô hình xoắn ốc (Stephens, R. 2015)

Sau khi các yêu cầu ban đầu của dự án được thiết lập, một danh sách các rủi ro tương ứng được chỉ ra để xem xét. Một số các rủi ro được đánh giá và chọn lọc để tìm ra phương án đối phó. Giải pháp cho các rủi ro đó được ghi nhận, và bản mẫu tương ứng các yêu cầu của dự án được triển khai. Bản mẫu này sẽ được đánh giá mức độ giải quyết các rủi ro đã đề cập và khả năng đáp ứng các yêu cầu đặt ra. Các yêu cầu nào được đáp ứng tốt sẽ chính thức thông qua cho giai đoạn thiết kế. Trong giai đoạn thiết kế, các yêu cầu về thiết kế cũng như các rủi ro cũng được xem xét, phân tích và chọn lọc. Một bản mẫu được xây dựng nhằm đánh giá khả năng đáp ứng các yêu cầu thiết kế cũng như giải quyết rủi ro. Khi bản mẫu thiết kế được xác nhận thỏa mãn các yêu cầu, nó sẽ được thông qua cho giai đoạn cài đặt. Trong giai đoạn cài đặt, áp dụng tương tự bốn bước như hai giai đoạn trên, các yêu cầu và rủi ro được đề ra. Giải pháp để đạt yêu cầu được áp dụng trong quá trình triển khai bản mẫu. Sau khi được xác

minh đáp ứng tốt các mục tiêu đề ra, bản mẫu này sẽ được chính thức triển khai cài đặt.

❖ Ưu điểm

- Người quản lý dự án và các đối tác dễ dàng giám sát và tương tác trong suốt quy trình phát triển
- Nếu giải quyết tốt các rủi ro thì khả năng thành công của dự án tăng cao
- Mô hình này có thể hỗ trợ giải quyết những phát sinh hoặc thay đổi
- Khả năng đúng hạn của dự án sẽ cao khi rủi ro được giải quyết tốt
- Phù hợp các dự án lớn, phức tạp

❖ Nhược điểm

- Việc đề ra, phân tích và đánh giá rủi ro khá phức tạp, cần nguồn nhân lực, thời gian và chi phí cao
- Người quản lý dự án và đối tác cần có trình độ phù hợp để giám sát cũng như tham gia thúc đẩy tiến độ
- Không phù hợp cho những dự án nhỏ, đơn giản

3.5 Mô hình quy trình hợp nhất (Rational unified process model)

Mô hình quy trình hợp nhất là một khung quy trình phát triển phần mềm đề cao tính linh hoạt trong việc lặp lại các công đoạn (workflow) thông qua bốn giai đoạn gồm Khởi đầu (Inception), Phân tích (Elaboration), Xây dựng (Construction), Chuyển tiếp (Transition) [4] (Hình 3.6).

Khởi đầu là giai đoạn nhằm xác định chủ đề và phạm vi dự án. Những tác nhân bên ngoài hệ thống cần được xác định cụ thể thông qua tất cả các trường hợp sử dụng. Việc diễn giải chi tiết các trường hợp sử dụng then chốt là cần thiết. Chủ đề của giai đoạn này bao gồm vấn đề cần giải quyết, mục tiêu cần đạt, tiêu chí đánh giá thành công của giai đoạn, phân tích rủi ro, xác định nguồn lực cần thiết và kế hoạch thực hiện. Kết quả của giai đoạn khởi đầu có thể bao gồm: đề cương của dự án, mô hình trường hợp sử dụng mức độ tổng quát, kết quả đánh giá tổng quát rủi ro, bản kế hoạch dự án và các bản mẫu.

Các tiêu chí đánh giá cho giai đoạn này có thể bao gồm:

- Sự đồng thuận của các bên chủ chốt về phạm vi, chi phí và thời gian trong kế hoạch
- Sự thấu hiểu các yêu cầu được minh chứng qua các trường hợp sử dụng
- Tính khả thi của quy trình phân mềm được lựa chọn, kết quả đánh giá rủi ro, dự tính chi phí, thời gian
- Chất lượng của bản mẫu kiến trúc
- Sự tương quan giữa thực tế và kế hoạch.

Phân tích là giai đoạn nhằm phân tích phạm vi vấn đề, phát triển chi tiết kế hoạch dự án và loại trừ các thành phần rủi ro lớn nhất đối với dự án. Điều kiện cần thiết để hoàn thành giai đoạn này trước hết là sự am hiểu đầy đủ toàn diện hệ thống bao gồm phạm vi hệ thống, yêu cầu chức năng và phi chức năng. Trong giai đoạn này, một bản mẫu kiến trúc khả thi được xây dựng thông qua một số vòng lặp công đoạn, với số vòng lặp tùy thuộc vào độ lớn, tính rủi ro và tính tiên phong của dự án. Chẳng hạn như, một dự án sử dụng công nghệ, kỹ thuật càng mới mẻ, thì mức độ rủi ro do thiếu kinh nghiệm sẽ càng cao.

Kết quả của giai đoạn phân tích có thể là:

- Một mô hình trường hợp sử dụng gần như hoàn thành (khoảng 80%)
 - Các yêu cầu bổ sung đối với yêu cầu phi chức năng và các yêu cầu chưa tương ứng với các trường hợp sử dụng
 - Bản mô tả, diễn giải cấu trúc của phần mềm
 - Một bản mẫu kiến trúc khả thi
 - Danh sách các rủi ro đã được cải thiện
 - Kế hoạch cho các giai đoạn của dự án và các tiêu chí đánh giá cho từng giai đoạn
 - Các tài liệu hướng dẫn người sử dụng (nếu cần thiết)
- Các tiêu chí đánh giá cho giai đoạn này có thể quan tâm đến:
- Mức độ bền vững của triển vọng và kiến trúc của sản phẩm
 - Mức độ hoặc khả năng giải quyết rủi ro của bản mẫu

- Mức độ chính xác và đầy đủ của kế hoạch cho giai đoạn tiếp theo (giai đoạn xây dựng), sự sẵn sàng của kế hoạch dự phòng

- Tất cả các bên chủ chốt của dự án đều đồng thuận với kết quả và tiến độ đạt được

- Sự tương quan giữa thực tế và kế hoạch được chấp nhận.

Xây dựng là giai đoạn nhằm phát triển, tích hợp và kiểm thử toàn bộ thành phần và chức năng vào sản phẩm. Việc quản lý nguồn lực và kiểm soát các hoạt động trong giai đoạn này rất được chú trọng nhằm tối ưu chi phí, tiến độ và chất lượng. Kết quả của giai đoạn này có thể bao gồm: một sản phẩm phần mềm được tích hợp trên các nền tảng hoàn chỉnh, tài liệu hướng dẫn sử dụng và thông tin mô tả các phiên bản.

Các tiêu chí đánh giá giai đoạn này có thể gồm:

- Mức độ bền vững và hoàn chỉnh của sản phẩm với phiên bản triển khai cho người dùng

- Tất cả các bên chủ chốt của dự án đều sẵn sàng chuyển giao sản phẩm cho người dùng

- Sự tương quan giữa thực tế và kế hoạch vẫn còn chấp nhận được.

Chuyển tiếp là giai đoạn nhằm chuyển giao sản phẩm cho người sử dụng. Sau khi sản phẩm đến tay người sử dụng, một số công việc có thể cần thực hiện như nâng cấp phiên bản mới, sửa chữa các lỗi kỹ thuật, hoàn chỉnh các chức năng đã được trì hoãn. Giai đoạn này bao gồm các công việc sau:

- Kiểm thử bê-ta nhằm xác nhận mức độ đáp ứng của hệ thống mới đối với yêu cầu người dùng

- Vận hành song song với hệ thống đang được thay thế

- Chuyển đổi cơ sở dữ liệu hiện hành

- Tập huấn người sử dụng và người bảo trì

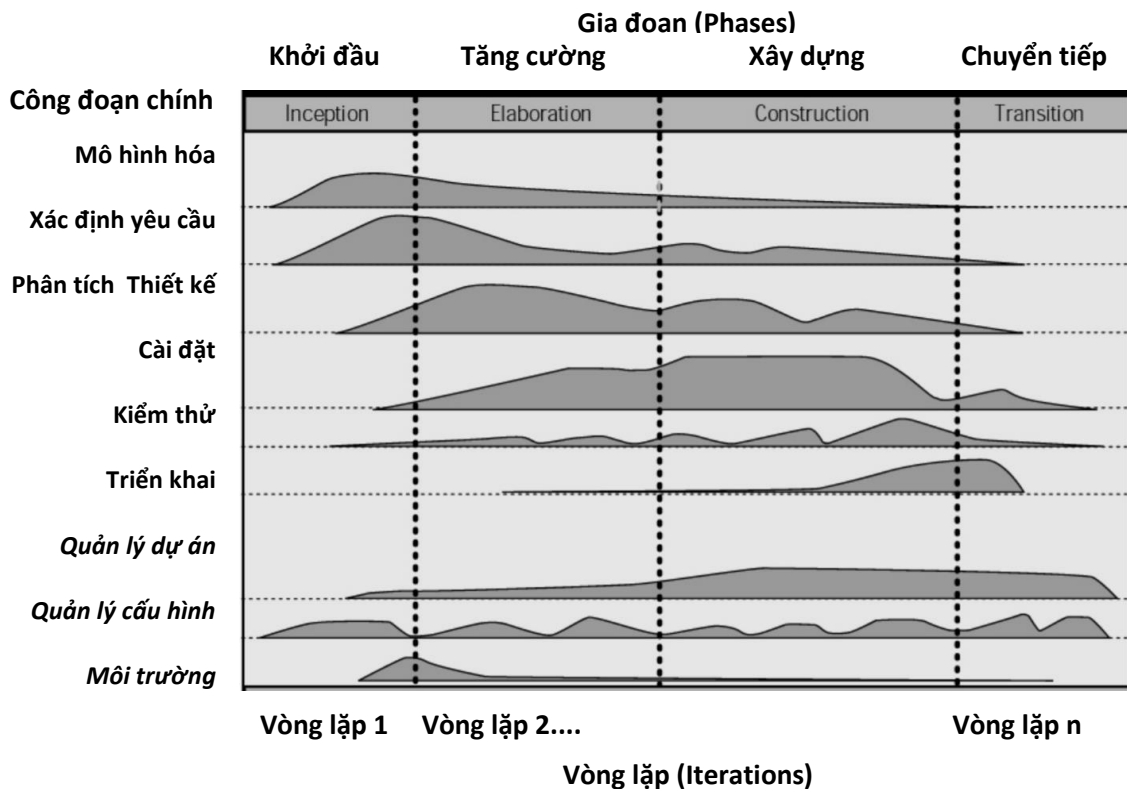
- “Xuất xưởng” sản phẩm đến bộ phận marketing, phân phối và bán hàng.

Mục tiêu chính của giai đoạn này bao gồm:

- Nhận được sự ủng hộ của người sử dụng

- Nhận được sự đồng thuận của tất cả các bên chủ chốt của dự án về việc các nội dung cốt lõi đã hoàn thành và thống nhất với các tiêu chí đánh giá ban đầu
- Đạt hiệu quả về chi phí thực tế trong sản phẩm cuối cùng.

Tiêu chí đánh giá của giai đoạn này tập trung vào sự hài lòng của người sử dụng và sự tương quan giữa thực tế cuối cùng và kế hoạch ban đầu.



Hình 3.6 Mô hình quy trình hợp nhất (RUP model) (@IBM)

Vòng lặp là một đặc trưng của RUP vì mỗi một giai đoạn đều có thể trải qua một số lượng các vòng lặp nhất định. Mỗi vòng lặp là hoàn chỉnh và đều có kết quả là một sản phẩm khả thi. Sản phẩm này thường là sản phẩm thành phần tăng trưởng (incremental element), sẽ tiếp tục được phát triển và hoàn thiện thông qua các vòng lặp đến phiên bản cuối cùng, hoàn chỉnh.

❖ Ưu điểm

- Việc sử dụng vòng lặp trong các giai đoạn giúp việc xác định các yêu cầu liên quan thực hiện thường xuyên và cho phép cập nhật chúng vào dự án

- Công đoạn phân tích trong các vòng lặp tập trung xác định và giải quyết các rủi ro giúp nâng cao khả năng thành công của dự án

- Cho phép áp dụng linh hoạt các mô hình phát triển phần mềm khác vào giai đoạn xây dựng nhằm phù hợp với quy mô và kiểu dự án khác nhau

- Trong giai đoạn khởi đầu và tăng cường, việc lập ra những tài liệu hữu ích có thể giúp giải quyết vấn đề sự thay đổi nhân sự trong dự án

- Các phiên bản phần mềm kiểu tăng trưởng có thể được hình thành khi cần thiết.

❖ **Nhược điểm**

- Mức độ phức tạp của mô hình này (có thể dễ nhầm lẫn với mô hình xoắn ốc)

- Do sự phức tạp, mô hình này thường yêu cầu nhiều nguồn lực hơn các mô hình đơn giản

- Việc phân tích rủi ro luôn luôn không đơn giản

- Không phù hợp với những dự án có độ rủi ro thấp

- Không phù hợp với những dự án nhỏ.

3.6 Phát triển hướng đối tượng (Object-oriented development)

Phương pháp này đề cao việc phát triển một sản phẩm phần mềm từ những thành phần đã có sẵn, được tích hợp, hiệu chỉnh và phát triển lên thành một sản phẩm hoàn chỉnh. Đây giống như việc tạo ra một chiếc xe hơi từ việc sử dụng các thành phần có sẵn như khung xe, bánh xe, động cơ...mà không phải bắt đầu từ việc đúc ra khung xe, làm ra các bánh xe và chế tạo động cơ.

Ngày nay, trong các dự án phần mềm, việc sử dụng những phần mềm thành phần, mô-đun sản phẩm có sẵn (từ các dự án khác hoặc từ ngân hàng phần mềm) là khá phổ biến. Điều này giúp tiết kiệm chi phí, thời gian và tăng mức độ an toàn khi sử dụng các sản phẩm có sẵn. Quy trình phát triển phần mềm áp dụng phương pháp này có thể được chia làm các giai đoạn như sau:

- Xác định yêu cầu

- Phân tích các thành phần hướng đối tượng đáp ứng yêu cầu

- Đánh giá phù hợp yêu cầu – thành phần hướng đối tượng

- Thiết kế phần mềm hướng đối tượng với các thành phần được thông qua

- Phát triển và tích hợp
- Kiểm thử, bàn giao
- Bảo trì

Phương pháp này được đánh giá khá phù hợp với các phần mềm dạng ứng dụng dịch vụ Web (sử dụng ngân hàng dịch vụ Web), ứng dụng dựa trên khung mẫu (Framework hỗ trợ các thành phần tích hợp).

Vấn đề cần lưu ý đối với phương pháp này là khả năng làm phức tạp công tác quản lý việc tích hợp các thành phần, mô-đun, đặc biệt trong quá trình kiểm thử cũng như giai đoạn bảo trì.

Bài tập

1. Hãy trình bày đặc trưng, ưu điểm và hạn chế của các mô hình phát triển phần mềm thác nước, bản mẫu, xoắn ốc, tăng trưởng và hợp nhất.

2. Hãy so sánh những điểm giống và khác nhau giữa các mô hình nêu trên. (Gợi ý: lập bảng hoặc ma trận so sánh)

3. Tìm hiểu và trình bày đặc trưng, ưu điểm và hạn chế của các phương pháp, mô hình phát triển phần mềm sau: Đồng bộ và ổn định (Synchronize and Stabilize), Phát triển ứng dụng tức thời (Rapid-Application Development) và Phát triển phần mềm linh hoạt (Agile Software Development).

Chủ đề thảo luận

1. Trò chơi điện tử (computer game) có phải là một sản phẩm phần mềm không? Hãy lựa chọn một vài mô hình phát triển nào là phù hợp để sản xuất trò chơi điện tử. (Gợi ý: không giới hạn các mô hình trong bài này)

2. Giả sử một dự án phần mềm đã đang được thực hiện, nhân lực của dự án đột ngột bị sụt giảm, khi này phương pháp nào nên được áp dụng để giảm thiểu rủi ro và có thể tiếp tục duy trì dự án một cách hiệu quả?

3. Hãy thử đề xuất một mô hình phát triển phần mềm mới, không thuộc nội dung được đề cập ở trên.

CHƯƠNG 4. NHÂN LỰC & CÔNG CỤ CASE

Chương này trình bày khái niệm và đặc điểm cơ bản của người lao động tham gia trực tiếp hoặc hỗ trợ quy trình sản xuất các sản phẩm phần mềm và giới thiệu khái quát công cụ CASE (Computer Aided Software Engineering tool). Sau khi học xong chương này, người học có thể hiểu và nhận biết được những người lao động theo vị trí việc làm, có thể hiểu và trình bày được các đặc điểm chung các công cụ CASE.

4.1 Giới thiệu

Đối với một sản phẩm phần mềm được phát triển chuyên nghiệp, một cá nhân không thể tự thực hiện và hoàn thành tất cả giai đoạn trong quy trình phát triển. Thành quả lao động cuối cùng là từ sự chung sức của một tập thể, một nhóm. Nhóm này thường được gọi là nhóm phát triển phần mềm.

Như vậy, một nhóm phát triển phần mềm như thế nào là hiệu quả?

Sau đây là các yếu tố chủ yếu cần xem xét nhằm quyết định chất lượng của nhóm phát triển

- Số lượng thành viên
- Đặc điểm của các thành viên
- Quy cách tổ chức
- Môi trường, điều kiện làm việc

4.2 Tiếp cận các nhóm làm việc

Sự phát triển rộng khắp và việc sử dụng phổ thông công nghệ thông tin (CNTT) ngày nay đòi hỏi trong tất cả các ngành, lĩnh vực cần có nhân lực liên quan trực tiếp hoặc gián tiếp đến mảng công nghệ này. Trong thị trường lao động thường có những lớp nhân sự như sau:

- Những người có thể sử dụng CNTT trong nghiệp vụ chuyên môn.
- Những người quản lý và vận hành các hệ thống CNTT
- Những người trực tiếp phát triển và tạo ra các sản phẩm CNTT
- Những người đầu tư vào lĩnh vực CNTT

Ở khía cạnh tham gia vào quy trình phát triển phần mềm, các loại hình công việc được nhiều người quan tâm, bao gồm phát triển ứng dụng, hỗ trợ ứng dụng và tiếp cận thị trường.

4.2.1 Phát triển ứng dụng

Lập trình viên: Những người phụ trách công việc chính là lập trình được gọi là lập trình viên. Công việc chủ yếu là chuyển đổi những tài liệu kỹ thuật (đặc tả, thiết kế) sang các mã lệnh (ngôn ngữ lập trình, ngôn ngữ mã hóa) và có khả năng tự kiểm tra chúng. Các lập trình viên có thể luân phiên chịu trách nhiệm giữa phát triển ứng dụng và công việc bảo trì. Ngoài ra, những chuyên gia trong nghề lập trình có thể làm rất tốt việc kiểm thử.

Kỹ sư phần mềm: Một kỹ sư phần mềm thông thường có khả năng thực hiện hầu hết các công việc liên quan trong quy trình phát triển phần mềm, chứ không đơn thuần việc lập trình. Hơn nữa, những người này hoàn toàn có khả năng lãnh đạo hoặc quản lý các dự án phần mềm. Khi này, những kinh nghiệm trong quá trình làm việc kỹ thuật sẽ giúp ích trong công tác lập kế hoạch, quản lý, điều hành. Một người là kỹ sư phần mềm khi họ có bằng cấp tương ứng được thừa nhận.

4.2.2 Hỗ trợ ứng dụng

Chuyên gia: Chuyên gia là những người có chuyên môn sâu, nhiều kinh nghiệm và có uy tín trong một lĩnh vực cụ thể. Những trường hợp nguồn nội lực sẵn có của nhóm phát triển không đáp ứng yêu cầu về một lĩnh vực nào đó, thì việc hợp tác với các chuyên gia là cần thiết. Có thể họ tham gia trực tiếp vào quy trình sản xuất phần mềm, hoặc họ chỉ ở vị trí là một nhà tư vấn. Ví dụ: một chuyên gia về nghiệp vụ chuyển tiền thời gian thực có thể xác định chính xác thời gian giữa các tác vụ, thủ tục chuyển tiền trong nước và quốc tế, hiểu rõ những quy định, luật lệ phải tuân theo của ngân hàng, chính phủ cũng như của các tổ chức chuyển tiền có liên quan.

Quản trị cơ sở dữ liệu: Những người quản lý việc lưu trữ và sử dụng dữ liệu cả về điều kiện vật lý (cơ sở vật chất, máy móc, thiết bị...) và điều kiện luận lý (các ràng buộc toàn vẹn, quy chuẩn) sao cho tính đúng đắn và sự an toàn được đảm bảo. Những người phụ trách công việc này thường có khả năng phân tích, thiết kế, xây dựng và bảo trì cơ sở dữ liệu cũng như các công cụ, ứng dụng phần mềm tương tác và khai thác cơ

sở dữ liệu có liên quan. Ví dụ: Người quản trị hệ MySQL server của một công ty phải chịu trách nhiệm đảm bảo an toàn khi các dịch vụ Web truy xuất dữ liệu phục vụ vận hành cho trang web của công ty. Như vậy, những người phụ trách mảng công việc này có thể hỗ trợ nhóm phát triển ứng dụng trong việc cung cấp phương án truy cập và khai thác dữ liệu sao cho phần mềm hoạt động hiệu quả.

Thiết kế đồ họa và xử lý ảnh: Những người này có khả năng tạo ra đối tượng đồ họa và hình ảnh phù hợp với từng thành phần của phần mềm hoặc loại phần mềm. Bằng cách sử dụng thành thạo và sáng tạo các công cụ chuyên ngành đồ họa và xử lý ảnh, những người phụ trách công việc này luôn cần thiết trong các dự án phát triển ứng dụng web, ứng dụng di động và ứng dụng giải trí.

4.2.3 Một số chuyên ngành kỹ thuật

Kỹ sư truyền thông: kỹ sư truyền thông có khả năng phân tích, thiết kế và cài đặt các thiết bị và phần mềm truyền thông. Họ có kiến thức và kỹ năng làm việc với các hệ thống truyền thông và mạng máy tính, đặc biệt là các hệ thống, thiết bị thu, phát, truyền dẫn tín hiệu. Tay nghề của họ được đào tạo liên quan đến chuyên ngành điện tử, khoa học máy tính và truyền thông.

Kỹ sư mạng máy tính: kỹ sư này có khả năng thu thập thông tin, xây dựng đặt kế hoạch, thiết kế, lắp đặt, quản trị và duy trì hoạt động của các loại mạng máy tính, phổ biến là mạng cục bộ. Khi phụ trách quản lý hệ thống mạng, người đó được gọi là quản trị mạng.

Người quản trị mạng luôn giữ một vị trí then chốt trong những công ty có ứng dụng CNTT trong hoạt động và sản xuất. Các công việc chủ yếu bao gồm quản lý tài khoản người sử dụng, quản lý các tài nguyên chia sẻ, duy trì liên thông mạng, bảo mật thông tin, đảm bảo an toàn dữ liệu và an ninh mạng.

4.2.4 Một số lĩnh vực khác

Chuyên gia về bảo mật: Một chuyên gia bảo mật chịu trách nhiệm bảo mật và sẵn sàng phục hồi thảm họa. Để bảo mật, một chuyên gia phải thiết lập các chuẩn cho bảo mật dữ liệu, giúp đỡ các đội dự án trong việc quyết định các yêu cầu bảo mật và thiết lập các chuẩn cho trung tâm bảo mật dữ liệu. Tương tự để phục hồi thảm họa, chuyên gia bảo mật giúp đỡ những người quản lý và các đội dự án trong việc xác định các dữ

liệu nguy cấp cần thiết cho tổ chức. Sau đó chuyên gia giúp trung tâm dữ liệu và các đội dự án trong việc phát triển và thử nghiệm các kế hoạch phục hồi thảm họa.

Đào tạo và huấn luyện: Công việc này thường được triển khai đối với nhân sự mới hoặc đối với một dự án có yêu cầu kỹ thuật mới. Người phụ trách công việc này bên cạnh chuyên môn, cần phải có khả năng truyền đạt và diễn giải tốt. Đào tạo và huấn luyện có thể thực hiện trong nội bộ tổ chức, hoặc cũng có thể do một công ty chuyên nghiệp bên ngoài đảm nhận.

Đảm bảo chất lượng: Những người phụ trách đảm bảo chất lượng phải có kiến thức về quy chuẩn được áp dụng, kỹ năng giao tiếp, kỹ năng giải quyết vấn đề nhằm thực hiện công việc kiểm tra chất lượng một cách hiệu quả. Họ cần phải có kinh nghiệm trong tất cả các khía cạnh phát triển của dự án để biết nên làm cái gì và vấn đề nảy sinh từ đâu. Đồng thời sự tinh tế và khả năng phát hiện ra những vấn đề cần phê bình cũng rất quan trọng. Không ai muốn bị nói trước công chúng là mình có lỗi mặc dù về mặt lý trí họ đều biết rằng công việc dự án sẽ có lợi từ những phê bình đó. Nhân viên đảm bảo chất lượng cần phải tinh tế với cả những chính sách và quy chuẩn được áp dụng.

Lập kế hoạch công nghệ: Các chuyên gia giám sát sự phát triển công nghệ xác định các xu hướng, lựa chọn các công nghệ thích hợp để thử nghiệm và tiến đến việc chạy đua trong thực hiện các kỹ thuật mới để tung ra thị trường. Công ty nào sở hữu nhiều kỹ thuật mới sẽ nhiều khả năng chiếm ưu thế trong môi trường cạnh tranh.

Ngoài ra, đối với những sản phẩm phần mềm thương mại, việc tiếp cận thị trường cũng cần một số nghiệp vụ hỗ trợ, chẳng hạn như:

Hỗ trợ sản phẩm: Nhân viên hỗ trợ sản phẩm làm việc cho nhóm người dùng cuối hoặc bán hàng để cung cấp những chuyên môn kỹ thuật liên quan đến sản phẩm hoặc những hỗ trợ trên đường dây nóng.

Tiếp thị sản phẩm: Nhân viên hỗ trợ tiếp thị làm việc cho nhà bán hàng để cung cấp những thông tin kỹ thuật cho đại diện bán hàng trong các tình huống tiếp thị. Loại công việc này đòi hỏi kỹ năng giao tiếp tốt với một vốn cơ bản kiến thức về tiếp thị.

Chuyên gia người sử dụng cuối: Chuyên gia người dùng cuối là người chuyển những yêu cầu sử dụng thành những ngôn ngữ kỹ thuật cho nhóm phát triển sử dụng.

Trong một vài tổ chức, đây là chức năng của người phân tích hệ thống hoặc giao cho kỹ sư phần mềm. Tóm lại, mọi công ty đều phải có sự kết hợp của những đặc điểm công việc khác nhau ở tất cả các bộ phận sao cho đồng bộ và hiệu quả.

4.3 Công cụ CASE

CASE (Computer Aided Software Engineering) là tên gọi những công cụ nào nhằm để hỗ trợ hoặc phục vụ các hoạt động trong quy trình phát triển phần mềm. Các công cụ CASE có thể hỗ trợ trong từng giai đoạn như đặc tả, phân tích, thiết kế, cài đặt, kiểm thử...và cũng có thể hỗ trợ các nghiệp vụ liên quan xuyên suốt như quản lý dự án. Do đó, việc sử dụng các công cụ này sẽ giúp quy trình sản xuất phần mềm nâng cao tính chuyên nghiệp và sản phẩm cuối cùng có chất lượng cao.

Các công cụ CASE có thể chia làm hai nhóm, bao gồm các công cụ riêng rẽ và các công cụ tích hợp.

Các công cụ riêng rẽ thường được sử dụng độc lập có thể phục vụ một hoặc nhiều nghiệp vụ tại một hoặc nhiều giai đoạn. Chẳng hạn, trong các giai đoạn phân tích yêu cầu, đặc tả, thiết kế, trên thị trường hiện nay có thể kể đến như Rational Rose (IBM), PowerDesigner (SAP). Đây là các công cụ mô hình hóa hệ thống dạng quan hệ (Relational model) hoặc hướng đối tượng (Object-oriented model). Đơn giản hơn, có thể là các trình hỗ trợ soạn thảo, biên dịch mã lệnh, như Dev-C++, Code::Blocks.

Các công cụ tích hợp (Integrated Development Environment) thường được sử dụng dưới dạng khung mẫu (Framework), hệ thống quản trị nội dung (Content Management System) hoặc dạng gói (Studio). Các công cụ dạng này chủ yếu hỗ trợ các giai đoạn cài đặt, kiểm thử và bảo trì. Một số công cụ tích hợp phổ biến trên thị trường hiện nay như Eclipse IDE, Laravel Framework, Wordpress CMS và MS Visual Studio.

Như vậy, tại sao trong các dự án chuyên nghiệp lĩnh vực phát triển phần mềm lại không thể thiếu CASE?

Như đã đề cập, quy trình phát triển phần mềm khá phức tạp và đòi hỏi nhiều chuyên môn, kỹ năng khác nhau trong từng giai đoạn. Các giai đoạn luôn phải chuyển đổi và trao đổi thông tin, dữ liệu lẫn nhau với các đặc thù của chúng. Các công cụ CASE giúp chuẩn hóa đầu vào và đầu ra ở từng giai đoạn, giúp tạo ra các bộ tài liệu

đồng nhất về quy cách, hỗ trợ vô số các tác vụ một cách tự động, trực quan và đồng bộ. Từ đó, CASE vừa giúp quy trình phát triển nhanh hơn về thời gian vừa tăng cường độ chuẩn xác của các yếu tố kỹ thuật. Một ví dụ đơn giản là, các công cụ hỗ trợ kiểm thử sẽ chỉ ra nhanh chóng hàng loạt các lỗi trong mã lệnh, thay vì lập trình viên phải tự rà soát lỗi bằng dò tìm thủ công.

Ba đặc điểm nổi bật của các công cụ CASE:

- Giao diện đồ họa (Graphical User Interface): cung cấp sẵn các thành phần điều khiển, tương tác trực quan, sinh động
- Khả năng tích hợp: hỗ trợ tích hợp trong cùng hệ thống hoặc từ hệ thống khác thỏa mãn quy chuẩn
- Tính tự động cao

Bài tập

1. Hãy phân tích điểm giống và khác nhau giữa lập trình viên và kỹ sư phần mềm, giữa quản trị mạng viên và kỹ sư mạng máy tính.

2. Hãy tìm hiểu thêm ít nhất 3 chuyên ngành kỹ thuật (ngoài nội dung bài học) xuất hiện phổ biến trong lĩnh vực công nghệ phần mềm, nêu đặc điểm của các chuyên ngành đó.

3. Hãy kể ra một số chuyên ngành không liên qua kỹ thuật phần mềm hoặc khoa học máy tính (không liên quan đến máy tính), nhưng lại có vai trò quan trọng trong vòng đời phần mềm. Phân tích nội dung câu trả lời.

4. Trình bày vai trò của công cụ CASE trong công nghệ phần mềm.

5. Hãy chỉ ra sự hạn chế của việc sử dụng công cụ CASE trong các dự án phần mềm.

6. Hãy kể tên tất cả công cụ CASE có trong khóa học này và phân loại chúng.

7. Cài đặt và thử sử dụng những chức năng cơ bản của hai công cụ Rational Rose (IBM), PowerDesigner (SAP), đánh giá điểm mạnh và so sánh chúng.

Chủ đề thảo luận

1. Giả sử một người vừa giỏi về kỹ thuật vừa giỏi về quản lý, thì trong một dự án phát triển phần mềm, người này nên được bố trí làm kỹ thuật hay phụ trách quản lý? Phân tích lý do sự chọn lựa?
2. Nên hay không nên sử dụng nhân sự giỏi tay nghề nhưng không có bằng cấp vào các dự án phần mềm? Giải thích lý do.
3. Trong những trường hợp nào thì không nên sử dụng công cụ CASE?
4. Sử dụng Framework trong phát triển các ứng dụng Web luôn có lợi thế hơn việc phát triển thủ công (không dùng Framework hoặc CMS). Nhận định này đúng hay sai? Tại sao?
5. Chi phí đầu tư công cụ CASE có ảnh hưởng như thế nào đến chất lượng sản phẩm phần mềm trong một dự án?

CHƯƠNG 5. THU THẬP YÊU CẦU

Nội dung của chương này tập trung trình bày ba kỹ thuật phổ biến phục vụ việc thu thập yêu cầu bao gồm phỏng vấn, bảng câu hỏi và quan sát. Kiến thức của chương này giúp người học hiểu và áp dụng được các kỹ thuật một cách hiệu quả trong công đoạn thu thập, khảo sát yêu cầu của khách hàng hoặc người sử dụng.

5.1 Giới thiệu

Thu thập và xác định các yêu cầu cho sản phẩm phần mềm là những công việc quan trọng đầu tiên trong quy trình phát triển phần mềm. Để những giai đoạn tiếp theo được triển khai đúng đắn và đầy đủ thì công đoạn khảo sát các yêu cầu phải được đảm bảo về tính phù hợp với mục tiêu đặt ra và được sự thống nhất của các bên tham gia phát triển phần mềm. Bất kỳ sự mâu thuẫn hoặc không thống nhất về ý nghĩa của các yêu cầu giữa các bên tham gia sẽ dẫn đến những khó khăn, bất lợi và thậm chí có khả năng gây ra thất bại cho dự án ở mức trầm trọng. Khi quy trình phát triển được thực hiện đến giai đoạn cài đặt thì mọi sai sót thuộc về công đoạn khảo sát sẽ dẫn đến lãng phí về thời gian, chi phí nhằm khắc phục hoặc giảm thiểu những hậu quả. Những vấn đề trong giai đoạn khảo sát yêu cầu được phát hiện càng trễ, thì khả năng thất bại của dự án càng cao.

Do việc ảnh hưởng dây chuyền đến hầu hết các giai đoạn của quy trình, công việc khảo sát bao gồm thu thập và phân tích yêu cầu luôn được quan tâm cẩn trọng trong thời kỳ khởi động của các dự án phần mềm. Trong nhiều trường hợp, đây còn là cơ sở chủ yếu để triển khai đề án nghiên cứu tính khả thi nhằm tạo tính thuyết phục cho dự án được phê duyệt triển khai. Việc thu thập và phân tích yêu cầu được thực hiện tốt sẽ giúp cho việc lập kế hoạch về tài chính, thời gian, nhân sự thuận lợi, góp phần vào nâng cao chất lượng của sản phẩm phần mềm sau cùng.

5.2 Khởi động việc khảo sát

Để công việc thu thập và phân tích được thực hiện hiệu quả thì các bên tham gia dự án phần mềm có trách nhiệm hợp tác ở từng giai đoạn trong quy trình phát triển phần mềm sao cho đạt được sự thống nhất về ý nghĩa các yêu cầu sẽ được đáp ứng sau này. Giai đoạn này cần sự tham gia trực tiếp của nhà đầu tư, người sử dụng và những đối tượng mà phần mềm sẽ phục vụ để đạt tới mục tiêu của dự án.

Tuy nhiên, phần lớn các trường hợp, yêu cầu được đặt ra một cách chủ động từ phía người sử dụng là không nhiều. Bản thân người sử dụng không có khái niệm chuyên môn về chương trình chạy trên máy tính, phần mềm, hoặc ứng dụng công nghệ thông tin. Vì thế, những thông tin mà họ đưa ra có xu hướng không liên quan nhiều đến kỹ thuật và công nghệ. Công việc cần làm là tạo điều kiện, cơ hội thuận lợi để những thông tin của yêu cầu được thể hiện càng rõ ràng càng tốt.

Bên cạnh đó, có thể khó khăn hơn trong quá trình khảo sát yêu cầu, những ngữ cảnh mà thông tin yêu cầu không rõ ràng. Ngay cả bản thân người sử dụng hoặc người đặt hàng cũng không hiểu đầy đủ hoặc không thể mô tả đầy đủ mong muốn của mình. Vì thế, công việc khảo sát yêu cầu cần áp dụng những kỹ thuật nhằm hạn chế những khó khăn vừa nêu và giúp tăng hiệu quả của giai đoạn này. Ngoài ra, kế hoạch cụ thể về thời gian, công việc và đối tượng tham gia cho việc khảo sát là cần thiết được chuẩn bị nhằm giúp những việc liên quan được triển khai tốt hơn.

5.3 Kỹ thuật khảo sát cơ bản

5.3.1 Phỏng vấn

Phỏng vấn là một kỹ thuật phổ biến để thu thập thông tin trong nhiều lĩnh vực. Đơn giản hơn, phỏng vấn là những buổi trò chuyện nhằm khai thác thông tin cần thiết cho mục đích nhất định. Đây là quá trình sử dụng những câu hỏi và thu nhận những câu trả lời xoay quanh những nội dung mong muốn tìm hiểu. Trong quy trình phát triển phần mềm, phỏng vấn thường được dùng cho giai đoạn thu thập và phân tích yêu cầu. Ưu điểm của kỹ thuật này là sự linh hoạt về thời gian, địa điểm, đối tượng tham gia phỏng vấn.

Một buổi phỏng vấn muốn đạt hiệu quả cao cần được chuẩn bị kỹ lưỡng các nội dung trao đổi thông qua các câu hỏi, lựa chọn đối tượng phù hợp, lập lịch hẹn với thời gian, địa điểm dành riêng cho đối tượng được phỏng vấn. Một số trường hợp, người được phỏng vấn còn yêu cầu thêm các điều kiện khác để tiến hành buổi gặp gỡ, như yêu cầu về chi phí thực hiện.

Nội dung câu hỏi cho buổi phỏng vấn cần được xây dựng dựa trên những phương diện của nhà đầu tư (về kinh phí, lợi ích kinh tế,...), phía phát triển phần mềm (về

công nghệ, kỹ thuật,...), và của người sử dụng (tính năng, công dụng,...). Đối với hình thức câu hỏi, thông thường có hai dạng là câu hỏi dạng đóng và câu hỏi dạng mở.

Bảng 5.1 Tóm tắt 2 dạng câu hỏi đóng và mở

Câu hỏi dạng đóng	Câu hỏi dạng mở
<ul style="list-style-type: none"> - Câu trả lời được giới hạn - Giúp định hướng tốt theo ý muốn của người hỏi - Không phù hợp nếu muốn khai thác những điểm phát sinh 	<ul style="list-style-type: none"> - Câu trả lời không giới hạn - Có thể bị lạc hướng mong muốn nếu người phỏng vấn kiểm soát không tốt - Giúp gợi mở những nội dung mở rộng được quan tâm

Một buổi phỏng vấn nên được tổ chức theo ba phần cơ bản như sau: (Bảng 5.2)

Bảng 5.2 Tóm tắt các bước cơ bản của một buổi phỏng vấn

- Phần mở đầu	<ul style="list-style-type: none"> + Nhằm tạo không khí thân thiện, chân tình và bước đầu dẫn nhập vào nội dung quan tâm + Sử dụng những câu hỏi giao thiệp, đơn giản + Lưu ý tính cách, thái độ, hành vi của người được phỏng vấn
- Phần nội dung chính	<ul style="list-style-type: none"> + Nhằm khai thác thông tin chính theo mục tiêu đặt ra + Sử dụng những câu hỏi đã được chuẩn bị + Tập trung điều phối câu hỏi để dẫn dắt câu trả lời phục vụ mục tiêu cần đạt

	+ Lưu lại nội dung trả lời bằng văn bản, âm thanh, phim ảnh
- Phân kết thúc	+ Tổng hợp lại thông tin vừa thu nhận được + Thể hiện sự quan tâm và trân trọng thông tin nhận được + Yêu cầu người cung cấp thông tin xác nhận lại thông tin nếu cần thiết

Một số lưu ý trong quá trình phỏng vấn:

- Luôn tạo sự tin tưởng cho người được phỏng vấn
- Cần tạo sự phản hồi thông qua thái độ của người phỏng vấn (qua gương mặt, ánh mắt, cử chỉ thân thể,...)
- Điều phối thời gian hợp lý khi câu trả lời dài hơn dự kiến
- Linh hoạt phối hợp câu hỏi dạng đóng và mở nếu cần thiết

Một số lưu ý khi xây dựng câu hỏi:

- Thông tin quan tâm theo dạng định tính (niềm tin, thói quen, sở thích, cảm nhận,...) hay định lượng (số lượng, tần suất,...)
- Kích bản của buổi phỏng vấn phù hợp với đối tượng được phỏng vấn (dựa trên nghề nghiệp, giới tính, độ tuổi,...)
- Kế hoạch dự phòng nếu thông tin nhận được chưa đạt yêu cầu

Hạn chế của kỹ thuật này là đòi hỏi người phỏng vấn phải có kỹ năng tốt để đạt được mục tiêu đặt ra. Bên cạnh đó, câu trả lời của người được phỏng vấn có thể chưa được chuẩn xác vì trả lời tức thời không được chuẩn bị trước. Kỹ thuật này không phù hợp nếu số lượng nhiều người lấy thông tin và với người có tính cách trầm lặng, hướng nội, ít giao tiếp. Ngoài ra, tính trung thực của thông tin bị ảnh hưởng nhiều do tính công khai trong giao tiếp trực tiếp với người lấy thông tin.

5.3.2 Bảng câu hỏi

Trên cơ sở xây dựng những dạng câu hỏi tương tự ở kỹ thuật phỏng vấn, những câu hỏi được định dạng, tổ chức theo những cấu trúc nhất định và được tập hợp trong một bảng câu hỏi. Khác với phỏng vấn, khảo sát qua bảng câu hỏi không triển khai nhận câu trả lời tức thời mà thực hiện qua một thời gian hẹn trước theo kế hoạch. Cách khảo sát này có thể áp dụng cho số lượng lớn đối tượng cần lấy thông tin, mà điều này không thực hiện được với phương pháp phỏng vấn.

Ưu điểm của phương pháp khảo sát qua bảng câu hỏi:

- Thông tin trả lời được chuẩn bị kỹ
- Tính trung thực của thông tin cao do thường không xác nhận thông tin cá nhân
- Người trả lời chủ động thời gian, địa điểm để trả lời câu hỏi

Một số hạn chế như sau:

- Không triển khai nhiều lần được nếu thiếu thông tin
- Khó ràng buộc và kiểm soát khi câu hỏi không được trả lời
- Chất lượng thông tin giảm nếu câu hỏi khó hiểu

5.3.3 Quan sát

Không giống như phương pháp phỏng vấn và bảng câu hỏi, quan sát là một phương pháp có thể thu nhận thông tin nhưng không trao đổi trực tiếp với đối tượng cần lấy thông tin. Phương pháp này chủ yếu dựa trên sự chủ động bao quát của người tiến hành lấy thông tin bằng cách tiến hành quan sát các hoạt động, đối tượng, mối quan hệ có liên quan đến chức năng của phần mềm sẽ triển khai và phục vụ.

Phương pháp này thường được sử dụng để lấy thông tin dạng tổng hợp của nhiều đối tượng, tham gia vào trong một chức năng hoặc một kịch bản cụ thể mà phần mềm sẽ phục vụ, đặc biệt là những giải pháp hướng đến các chuỗi hành vi, thao tác của người sử dụng đối với thiết bị hoặc phương tiện mà họ sử dụng. Điển hình như các thông tin mô tả chuỗi hành vi của những người sử dụng các thiết bị với màn hình cảm ứng, các thao tác của bàn tay, ngón tay chặn hạn. Bên cạnh đó, các thông tin phục vụ phát triển các ứng dụng hướng đến giao tiếp giữa người và máy cũng phù hợp với phương pháp quan sát. Các cải tiến về chức năng, cách sử dụng các máy rút tiền tự động sao cho người sử dụng dễ thao tác và an toàn hơn chẳng hạn.

Việc quan sát có thể thực hiện bằng mắt thường hoặc bằng thiết bị quay phim. Quan sát bằng mắt thường chỉ phù hợp đối với các chuỗi hành vi, thao tác đơn giản, ít công đoạn và số lượng người tham gia không nhiều. Khi này, người lấy thông tin kết hợp vừa quan sát vừa ghi chép thông tin nhận được. Khó khăn của cách làm này là cần nhiều thời gian, cần lựa chọn kỹ không gian tiếp cận và sự trung thực của những người được quan sát. Nhiều trường hợp, người được quan sát cảm thấy không tự nhiên và có hành vi không như thường lệ. Điều này làm giảm tính chính xác của chuỗi hành vi được quan sát. Để khắc phục hạn chế khi quan sát bằng mắt thường, quan sát bằng thiết bị quay phim ngày nay được dùng phổ biến hơn. Ngoài sự đầu tư về kinh phí phải được tính toán kỹ, thì quan sát bằng thiết bị quay phim vừa tiết kiệm được thời gian, dễ lựa chọn không gian quan sát, mà còn tận dụng làm tư liệu số để lưu trữ và xử lý về sau.

Bài tập

1. Trình bày vai trò của việc khảo sát yêu cầu trong quy trình phát triển phần mềm.
2. Hãy chỉ ra thêm những khó khăn trong giai đoạn khảo sát yêu cầu (ngoài nội dung bài học). Phân tích những khó khăn và đề xuất giải pháp.
3. Giả sử sản phẩm phần mềm mục tiêu là một ứng dụng quản lý thời gian biểu, hãy lập ra tối thiểu 10 câu hỏi để khảo sát yêu cầu của khách hàng thông qua 2 phương pháp phỏng vấn và bảng câu hỏi.
4. Hãy đưa ra ít nhất 3 trường hợp khảo sát yêu cầu khách hàng mà phương pháp quan sát chiếm ưu thế hơn (phù hợp hơn) so với 2 phương pháp còn lại.
5. Hãy tìm hiểu các kỹ thuật, phương pháp khác (ba kỹ thuật trong bài) có cùng chức năng thu thập yêu cầu.

Chủ đề thảo luận

1. Trong trường hợp các sản phẩm phần mềm thương mại đã sẵn dùng (không phát triển mới), việc khảo sát yêu cầu của khách hàng đối với sản phẩm đó có cần thiết nữa không? Tại sao?

2. Thay vì khảo sát yêu cầu khách hàng cụ thể, việc khảo sát thị trường phần mềm hoặc thị hiếu của người sử dụng để sản xuất ra một sản phẩm mới (chưa có trên thị trường), thì phương pháp khảo sát yêu cầu sẽ là gì và được thực hiện như thế nào?

CHƯƠNG 6. PHÂN TÍCH & ĐẶC TẢ YÊU CẦU

Nội dung của chương này chủ yếu trình bày các phương pháp cơ bản trong việc đặc tả yêu cầu phần mềm, chú trọng các phương pháp sử dụng ngôn ngữ tự nhiên và đồ họa. Kiến thức của chương này giúp người học hiểu và xác định được các yêu cầu chức năng, phi chức năng, có thể sử dụng ngôn ngữ tự nhiên có cấu trúc để xác định và đặc tả yêu cầu. Sau khi thực hành các bài tập, người học có thể lập tài liệu đặc tả yêu cầu ở mức độ cơ bản.

6.1 Giới thiệu

Sau khi đã có đủ thông tin ban đầu cần thiết về yêu cầu cho sản phẩm, công việc tiếp theo là phải phân tích chúng để hiểu rõ, hiểu đúng về các yêu cầu và tiến đến đặc tả toàn bộ yêu cầu dưới dạng tài liệu chính thức. Tài liệu này phải được tất cả các bên tham gia dự án thông qua nhằm đảm bảo tính đúng đắn về ý nghĩa của yêu cầu phù hợp với mục tiêu đặt ra và đảm bảo rằng các bước tiếp theo phải vận dụng đúng các yêu cầu đó. Nói cách khác, sản phẩm sau cùng phải thỏa mãn các yêu cầu trong tài liệu, phải cung cấp đầy đủ các chức năng trong hợp đồng.

6.2 Phân tích yêu cầu

Có nhiều phương cách để phân loại các yêu cầu trong một dự án phần mềm, yêu cầu chức năng và yêu cầu phi chức năng là một kiểu phân loại phổ biến.

6.2.1 Yêu cầu chức năng (Functional Requirement)

Yêu cầu chức năng là những ràng buộc mô tả về những gì mà phần mềm phải phục vụ, dựa trên một nguồn đầu vào, sau khi được xử lý sẽ có kết quả đầu ra cụ thể. Ví dụ, chức năng tìm kiếm, chọn lọc, sắp xếp danh mục hàng; chức năng trả lời email tự động khi có đơn đặt hàng, chức năng in ấn trực tuyến.

Các yêu cầu chức năng thường dễ xác định do chúng hay gắn liền với các nghiệp vụ, thao tác ngoài thực tế được tin học hóa hoặc tự động hóa bằng giải pháp phần mềm.

6.2.2 Yêu cầu phi chức năng (Non-functional Requirement)

Yêu cầu phi chức năng là những ràng buộc mô tả về những gì mà phần mềm phải thỏa mãn (không đề cập một nghiệp vụ, thao tác cụ thể nào), thường dựa trên các đặc

điểm, tính chất liên quan đến cơ chế xử lý, sự tin cậy và sự an toàn. Ví dụ như, hệ thống nhận đăng ký học phần phải đảm bảo số lượng 1000 truy cập tại một thời điểm, giao diện ứng dụng phải rõ ràng, bố cục hợp lý, thân thiện, dễ sử dụng.

Mặc dù yêu cầu phi chức năng khó được nhận biết hơn yêu cầu chức năng, nhưng yêu cầu phi chức năng lại có ảnh hưởng toàn cục rộng hơn trong việc vận hành của sản phẩm phần mềm. Chẳng hạn như, một hệ thống lưu trữ, xử lý dữ liệu lớn với rất nhiều chức năng hiện đại, tuy nhiên hệ thống lại bị lỗi về chế độ an toàn, độ tin cậy thấp, việc này có thể gây tổn hại dữ liệu nghiêm trọng hoặc tê liệt các chức năng.

Trên thực tế, tùy vào cách phát biểu các yêu cầu ở mức độ khái quát hay cụ thể mà chúng có thể bao gồm vừa là chức năng và cả phi chức năng. Ví dụ như, phải đảm bảo an toàn của hệ thống bằng phương pháp chứng thực, bao gồm chức năng quản lý tài khoản người dùng, đăng nhập, và hỗ trợ mật mật khẩu.

Một cách tổng quát, các yêu cầu phi chức năng có thể tham chiếu đến

- Sản phẩm phần mềm,
- Tổ chức, công ty, doanh nghiệp (môi trường làm việc, chính sách...)
- Các yếu tố khác (quy định, luật pháp,...)

6.3 Đặc tả yêu cầu phần mềm

Các yêu cầu sau khi được phân loại sẽ được chính thức được đặc tả vào tài liệu nhằm làm cơ sở trong việc ký kết các thỏa thuận, hợp đồng, làm tư liệu cho quản lý dự án, tài liệu hướng dẫn cho các giai đoạn tiếp theo, cũng như làm cơ sở để đối chứng, xác nhận mục tiêu dự án có được đáp ứng hoặc thỏa mãn hay không.

Đặc tả yêu cầu là hoạt động sử dụng các phương tiện như ngôn ngữ, đồ họa, toán học mô tả yêu cầu một cách chi tiết, khoa học và lưu trong tài liệu.

Ngôn ngữ tự nhiên: thông thường, một câu diễn giải một yêu cầu

Ngôn ngữ tự nhiên có cấu trúc: ngôn ngữ được trình bày trong một định dạng bảng hoặc biểu mẫu

Ngôn ngữ thiết kế: giống như một loại ngôn ngữ lập trình, thường được dùng để diễn giải mô hình hoạt động của hệ thống.

Đồ họa: các mô hình đồ họa kèm theo các diễn giải bằng ngôn ngữ thường được dùng để trình bày các yêu cầu chức năng. Ví dụ như, mô hình trường hợp sử dụng (Use-case diagram) (Hình 6.1).

Ký hiệu toán học: các ký hiệu trong hệ thống toán học như đại số quan hệ hoặc toán tập hợp. Tuy nhiên, việc sử dụng phương cách cân cân nhắc do nó gây khó hiểu cho khách hàng, người sử dụng, hoặc những ai không có kiến thức về nó.

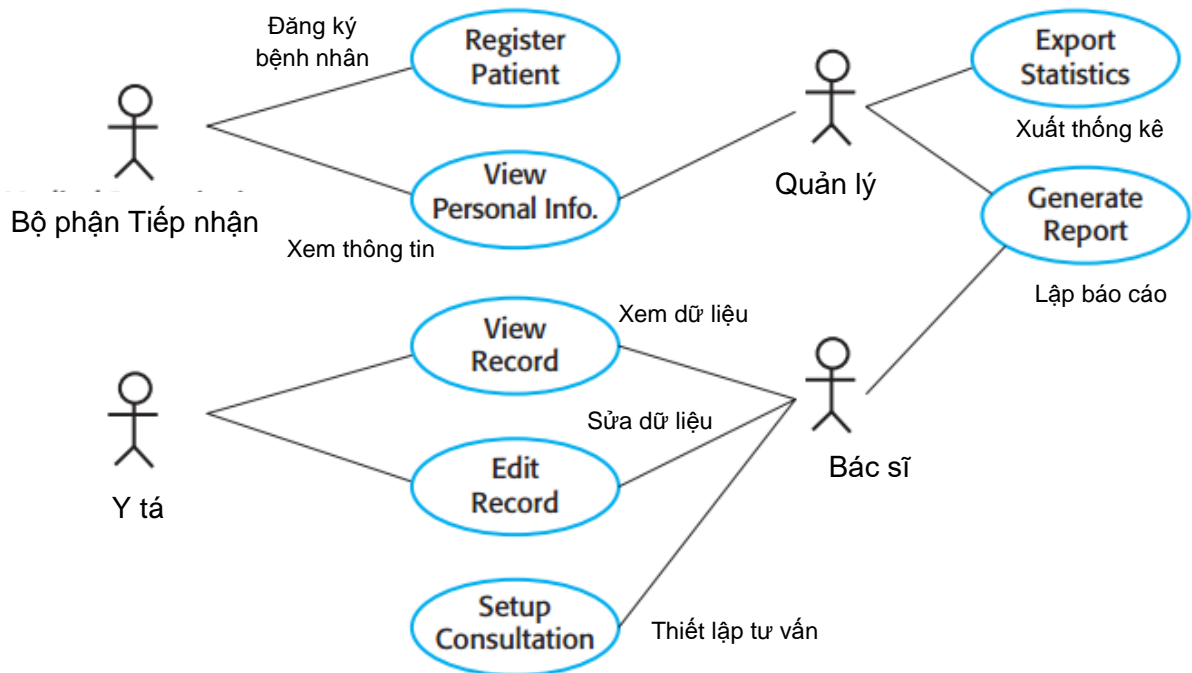
Bảng 6.1 Ví dụ về việc đặc tả yêu cầu Phần mềm điều khiển pha café

Pha chế đồ uống/ Phần mềm điều khiển/ Phân hệ café/ v.1.6	
Chức năng	Tính liều lượng nước pha café: lượng phù hợp
Diễn giải	Tính lượng nước nóng cho mức café phù hợp khi định mức nước trong khoảng an toàn từ 50ml -200ml
Đầu vào	Khối lượng café
Nguồn gốc	Cân cảm biến khối lượng café
Đầu ra	Lượng nước nóng tương ứng
Đích đến	Thiết bị điều khiển lưu lượng nước
Hoạt động/Quy trình	Nếu khối lượng café nằm trong mức an toàn dưới 200 gram, thì trả về mức nước tương ứng từ 50ml đến 200ml. Nếu quá mức trả mức nước về 0, hoặc nếu khối lượng café bằng 0 cũng trả mức nước về 0.
Yêu cầu	Khối lượng café không âm
Điều kiện trước đó	Nước nóng đạt tối thiểu 90 độ C

Điều kiện sau đó	Nhiệt độ phòng không dưới 10 độ C
Tác động	

Bảng 6.2 Ví dụ đặc tả yêu cầu dạng ngôn ngữ tự nhiên có cấu trúc

Điều kiện	Quy trình
Khối lượng café = 0	Lượng nước = 0
Khối lượng café > giới hạn	Lượng nước = 0
Khối lượng café = [1,100gram]	Lượng nước = 50ml
Khối lượng café = [101,200gram]	Lượng nước = 200ml



Hình 6.1 Một trường hợp trình bày bằng đồ họa trong đặc tả yêu cầu của phần mềm quản lý thông tin bệnh viện (Sommerville, I. 2011)

Bên cạnh các kỹ thuật trên, đặc tả có thể được phân nhóm theo dạng có hình thức, phi hình thức và kết hợp.

Bảng 6.3 Tóm tắt hai kiểu đặc tả hình thức và phi hình thức

Đặc tả phi hình thức	Đặc tả hình thức
<ul style="list-style-type: none">- Dễ hiểu, dễ sử dụng- Mềm dẻo, ít ràng buộc- Thiếu sự chính xác- Nhập nhằng, nhiều ngữ nghĩa	<ul style="list-style-type: none">- Khó hiểu, hạn chế đối tượng sử dụng- Chính xác (trên cơ sở toán học)- Công cụ hỗ trợ chủ yếu- Chi phí cao

Những thách thức của công đoạn đặc tả có thể gặp:

- Vừa phải phi hình thức để cho khách hàng hiểu và vừa phải hình thức đủ để có thể phát triển về kỹ thuật
- Khó phân biệt rạch ròi biên giới của đặc tả (cái gì cần xây dựng) và thiết kế (xây dựng như thế nào).

Ví dụ như, tài liệu đặc tả phải liệt kê các ràng buộc nhưng không bao giờ chỉ ra cách làm thế nào để thoả mãn ràng buộc (thuật toán nào phải sử dụng, chẳng hạn).

6.4 Xác nhận đặc tả

Toàn bộ tài liệu đặc tả phải được tất cả các bên có liên quan thống nhất thông qua, khi này tài liệu đặc tả mới chính thức có giá trị sử dụng cho các công việc tiếp theo. Nhằm đảm bảo chất lượng của tài liệu đặc tả, việc lặp lại các bước phân tích, đặc tả, xác nhận là cần thiết.

Nguồn gốc chính của lỗi trong các phần mềm đã phân phối, có khả năng là những lỗi trong tài liệu đặc tả và những lỗi này chỉ được phát hiện khi khách hàng hoặc người sử dụng thao tác đến chức năng đó. Xác nhận hoặc xét duyệt là cách tốt nhất để kiểm tra đặc tả. Mục đích là xác định đặc tả có đúng đắn với mục tiêu của dự án không. Nhóm đảm bảo chất lượng phần mềm có vai trò chủ trì trong công tác này.

Bài tập

1. Trình bày vai trò của công đoạn phân tích và đặc tả yêu cầu trong quy trình phát triển phần mềm.
2. Hãy kể ra ít nhất 3 yêu cầu chức năng và 3 yêu cầu phi chức năng của một sản phẩm phần mềm, ngoài nội dung bài học.
3. Hãy phân nhóm các phương tiện đặc tả trong bài học này thành đặc tả hình thức và phi hình thức hoặc kết hợp.
4. Sử dụng dữ liệu trả lời của Câu 3 ở Chương 5 về khảo sát yêu cầu và các phương tiện đặc tả trong bài này để trình bày thành một tài liệu đặc tả cụ thể.
5. Hãy tìm hiểu các yêu cầu trong tài liệu đặc tả phần mềm theo IEEE, trình bày nội dung các yêu cầu và áp dụng chúng cho một phần mềm cụ thể.

CHƯƠNG 7. THIẾT KẾ

Chương này trình bày vai trò và nội dung cơ bản của ba công đoạn thiết kế quan trọng trong quy trình phát triển phần mềm bao gồm thiết kế kiến trúc, thiết kế hệ thống và thiết kế giao diện. Sau khi học và thực hành bài tập, người học có thể hiểu đặc điểm của từng công đoạn và có thể tổ chức triển khai từng công đoạn thiết kế trong một dự án đơn giản.

7.1 Giới thiệu

Tiếp nối với giai đoạn phân tích-đặc tả trong quy trình phát triển phần mềm, giai đoạn thiết kế đóng vai trò vừa hoàn thiện việc đặc tả vừa làm cơ sở cho giai đoạn cài đặt. Sản phẩm của quá trình thiết kế là bước chuyển trung gian quan trọng giữa những yêu cầu mang tính khái niệm từ phía khách hàng hoặc người sử dụng và yếu tố kỹ thuật liên quan đến cài đặt. Giống như quy trình xây dựng một ngôi nhà, việc lập trình, cài đặt một phần mềm không thể được làm tốt nếu dựa trên một bản thiết kế tồi.

Sau giai đoạn đặc tả, kết quả có được từ quá trình thu nhận các yêu cầu cho phần mềm thường chỉ nằm ở mức quan niệm hoặc trừu tượng. Để tiếp cận dần đến mức cụ thể, những yêu cầu cần được chi tiết hơn thông qua thể hiện của các đối tượng, dữ liệu và các ràng buộc mà hệ thống quản lý hoặc tương tác, tác động với chúng. Chúng được biểu diễn trong giai đoạn thiết kế một cách đồng bộ có tính hệ thống và cho ra các kết quả để sử dụng trong quá trình cài đặt.

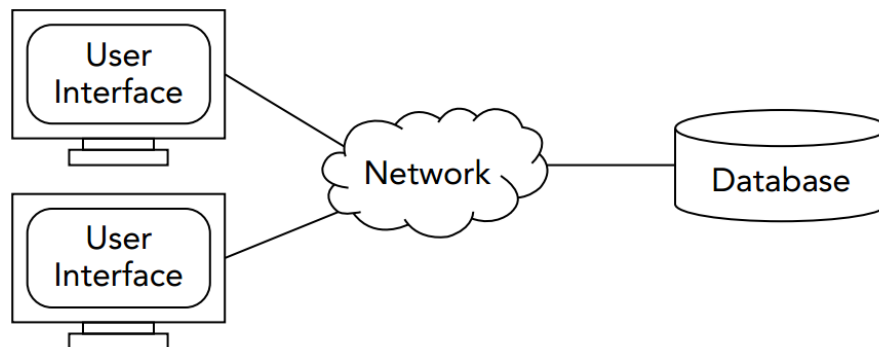
Công đoạn thiết kế thường hướng đến các mục đích gồm thiết kế kiến trúc phần mềm, thiết kế hệ thống và thiết kế giao diện.

7.2 Thiết kế kiến trúc phần mềm

Kiến trúc phần mềm được xem là một bức tranh chung mô tả toàn bộ các thành phần chính cấu thành phần mềm hoặc hệ thống phần mềm, một vài trường hợp có thể mô tả cả môi trường xung quanh phần mềm. Số lượng các thành phần bên trong một phần mềm là rất lớn, do đó, tùy vào cấp độ mô tả của kiến trúc phần mềm mà các thành phần được tổng quát hóa hay được cụ thể hóa. Chẳng hạn như, kiến trúc một phần mềm đơn giản bao gồm thành phần giao diện, thành phần xử lý và thành phần dữ liệu được mô tả với 2 cấp độ tổng quát và chuyên biệt (Bảng 7.1).

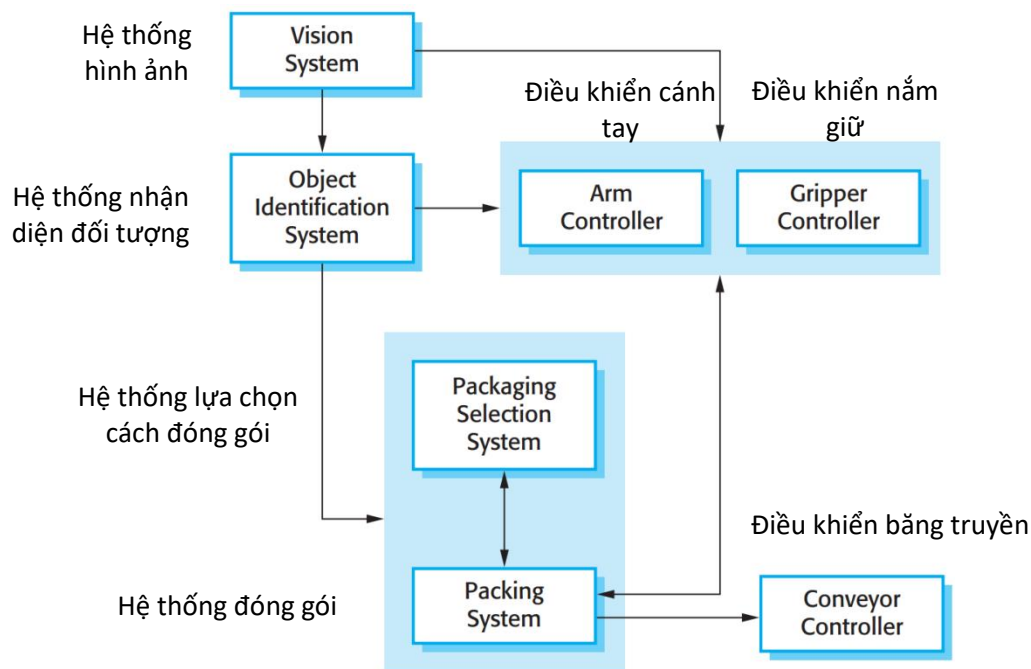
Bảng 7.1 Một trường hợp thiết kế kiến trúc từ mức độ tổng quát đến chuyên biệt cấp 1

Tổng quát	Chuyên biệt cấp 1
Giao diện	Chủ
	Nhánh
Xử lý	Đồ họa
	Số liệu
Dữ liệu	Nội dung
	Ràng buộc



Hình 7.1 Kiến trúc một phần mềm dạng Client – Server

Một ví dụ khác, một kiến trúc phần mềm dạng Client – Server với một Server dữ liệu (database) và hai máy khách (clients) kết nối với nhau qua hệ thống mạng (Network) (Hình 7.1).

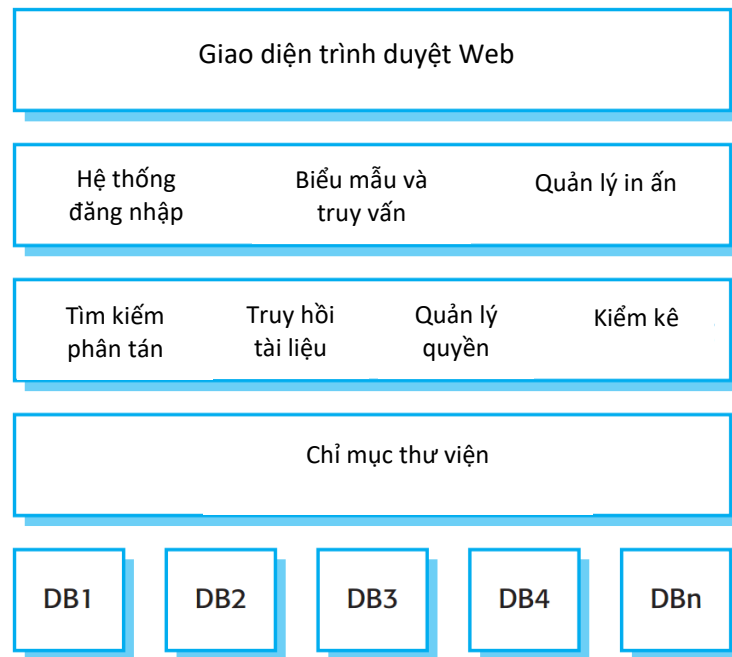


Hình 7.2 Kiến trúc một mô-đun điều khiển người máy đóng gói kiện hàng

(Sommerville, I. 2011)

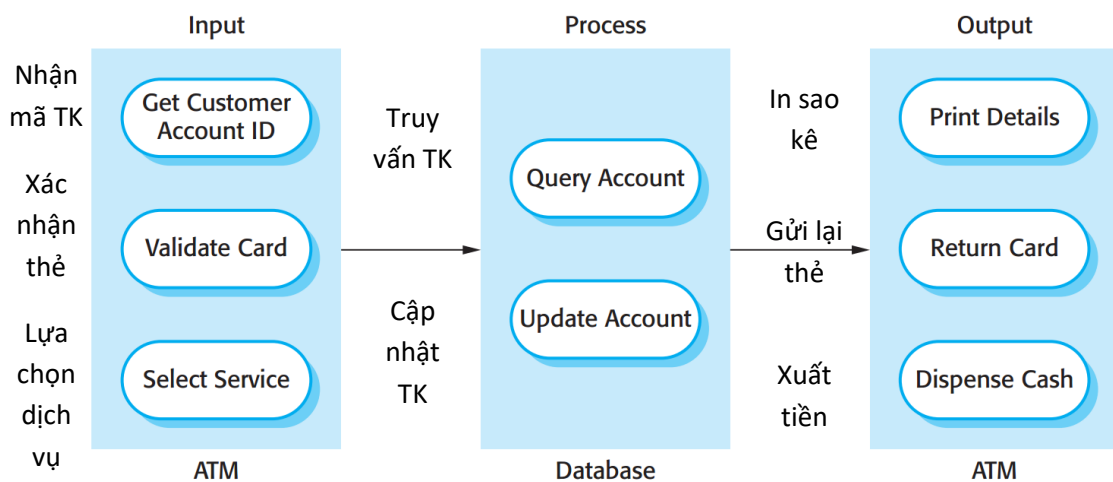
Như vậy, có thể thấy rằng tài liệu thiết kế kiến trúc phần mềm nên được trình bày dạng hình ảnh, bảng biểu và các chú thích bằng ngôn ngữ tự nhiên (Bảng 7.1, Hình 7.1 và 7.2). Đây là các phương tiện trực quan có khả năng biểu diễn, diễn giải các mối quan hệ phức tạp giữa số lượng lớn các thành phần cấu thành nên phần mềm.

Trong số các dạng mô hình, mô hình phân lớp được sử dụng khá phổ biến trong thiết kế kiến trúc phần mềm do tính rõ ràng, dễ áp dụng và dễ hiểu với nhiều bên tham gia dự án phần mềm (Hình 7.3).



Hình 7.3 Kiến trúc phần mềm thông tin thư viện dạng phân lớp (Sommerville, I. 2011)

Bên cạnh việc thể hiện kiến trúc với dạng khái niệm phân lớp, việc mô tả kiến trúc hướng ứng dụng cũng được sử dụng phổ biến, do nó dễ diễn giải mục đích của hệ thống (Hình 7.4).



Hình 7.4 Kiến trúc phần mềm hệ thống máy rút tiền ATM dạng hướng ứng dụng (Sommerville, I. 2011)

7.3 Thiết kế hệ thống

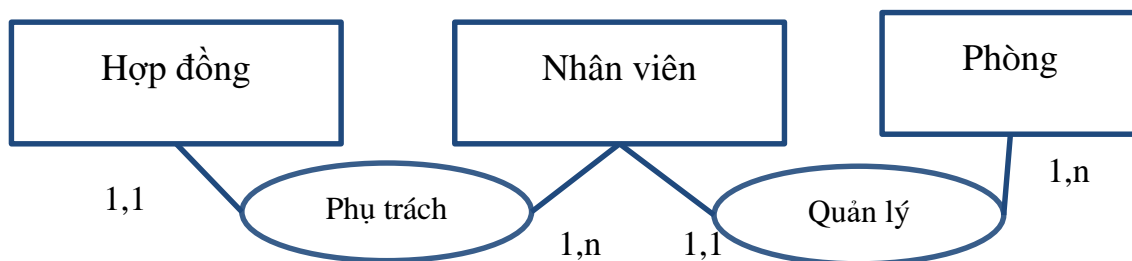
Trong khi kiến trúc phần mềm có vai trò chủ yếu trong việc định hình các thành phần, sắp xếp vị trí, trật tự của chúng nhằm cấu thành nên một thể thống nhất, thiết kế hệ thống sẽ phải chỉ ra được cụ thể các thành phần dữ liệu, thành phần chức năng, thành phần xử lý, cũng như các mối quan hệ, ràng buộc, sự ảnh hưởng lẫn nhau giữa chúng.

Chẳng hạn như, những ví dụ nêu trên có mô tả thành phần dữ liệu, tuy nhiên ở cấp độ kiến trúc phần mềm chưa diễn giải được nội dung và ràng buộc của dữ liệu đó là gì và như thế nào. Trong ví dụ hệ thống thư viện, có đề cập chức năng đăng nhập, tuy nhiên cũng chưa thấy mô tả đầy đủ ai đăng nhập và đăng nhập như thế nào.

Tùy vào mức độ tiếp cận hệ thống cụ thể hóa (top down approach) hoặc trừu tượng hóa (bottom-up approach) mà quy trình thiết kế có thể áp dụng các công cụ hỗ trợ nhằm rút ngắn thời gian lập trình, tạo điều kiện thuận lợi hơn cho quá trình cài đặt. Giáo trình này giới thiệu hai mô hình thiết kế hệ thống nổi tiếng theo hai phương pháp phân tích hệ thống Merise và phương pháp phân tích hệ thống với UML.

7.3.1 Mô hình Thực thể - Quan hệ (Entity – Relationship model)

Phương pháp Merise tiếp cận một hệ thống thông tin với ba cấp độ gồm cấp quan niệm (conceptual level), cấp luận lý (logical level) và cấp vật lý (physical level). Trong đó, mô hình Thực thể - Quan hệ được sử dụng để thiết kế hệ thống ở hai cấp độ đầu tiên. Qua đó, một hệ thống trong thế giới thực được phân tích và chuyển đổi thành các thực thể và các mối quan hệ giữa các thực thể (Hình 7.5).



Hình 7.5 Một trường hợp áp dụng mô hình Thực thể - Quan hệ

Mô hình cơ sở dữ liệu được hướng đến là dạng quan hệ (relational model), được chuyển đổi từ mô hình Thực thể - Quan hệ. Chẳng hạn như (Hình 7.6),

Bảng dữ liệu nhân viên

Mã nhân viên	Họ tên	Ngày sinh	Mã phòng	...
NV0011	Nguyễn Văn A	15/5/1975	PHCTC	...
NV0012	Trần Thị B	25/3/1980	PTCKH	...
...

Bảng dữ liệu phòng

Mã phòng	Tên phòng	Số điện thoại	...
PHCTC	Phòng Hành chính – Tổ chức	02923 895518	...
PTCKH	Phòng Tài chính – Kế hoạch	02923 847131	...
...

Bảng dữ liệu hợp đồng

Mã hợp đồng	Ngày ký	Giá trị	Mã nhân viên	...
HD0071	12/3/2018	75.000.000	NV0012	...
...

Hình 7.6 Tổng hợp các bảng dữ liệu dạng quan hệ

Các thành phần chức năng của hệ thống được thiết kế theo hướng phục vụ lập trình hướng thủ tục (Procedure-Oriented Programming). Đối với các phần mềm có cấu trúc dữ liệu dạng quan hệ và không có nhu cầu sử dụng lại các thành phần đã được xây dựng sẵn, thì việc áp dụng mô hình thiết kế hệ thống này là thích hợp.

7.3.2 Mô hình hướng đối tượng với UML

❖ UML (Unified Modeling Language) – Ngôn ngữ mô hình hóa đồng bộ

UML là một ngôn ngữ tiêu chuẩn nổi tiếng được thế giới công nhận, với chức năng chủ yếu dành cho việc đặc tả, hình tượng hóa, kiến trúc hóa các đối tượng trong hệ thống phần mềm [8]. UML cung cấp một ngân hàng sơ đồ kỹ thuật chuyên dùng cho việc mô hình hóa các hệ thống, đặc biệt các hệ thống hướng được thiết kế và phân tích hướng đối tượng. Một số sơ đồ được sử dụng rộng rãi như:

- Sơ đồ lớp (Class Diagram)
- Sơ đồ trường hợp sử dụng (Use Cases Diagram)
- Sơ đồ trình tự (Sequence Diagram)
- Sơ đồ hoạt động (Activity Diagram)

❖ Thiết kế hướng đối tượng

Hệ thống được thiết kế theo mô hình hướng đối tượng sẽ là một tổ hợp có cấu trúc của các đối tượng có tính tương tác lẫn nhau. Mô hình hướng đối tượng chuyển đổi thế giới thực thành các đối tượng được cấu thành từ thành phần dữ liệu và thành phần chức năng. Ở mức tổng quát, một lớp đối tượng là cơ sở để tạo ra một đối tượng, và lớp đối tượng là một khuôn mẫu quy định sẵn về dữ liệu và chức năng. Vì vậy, các đối tượng được tạo ra cùng một lớp sẽ giống nhau về thành phần dữ liệu và thành phần chức năng. Thành phần chức năng có thể xử lý nội dung của chính thành phần dữ liệu trong cùng một đối tượng hoặc không làm gì ảnh hưởng đến dữ liệu. Chẳng hạn như, trong một trò chơi đua xe hơi, một lớp xe hơi (car class) có thể quy định như sau: (Hình 7.7).

Chính vì thành phần dữ liệu và chức năng được gói gọn trong một lớp, nên mô hình này có ưu điểm là khả năng thích nghi cao với những thay đổi hoặc có lỗi xảy ra liên quan đến thiết kế dữ liệu và chức năng.

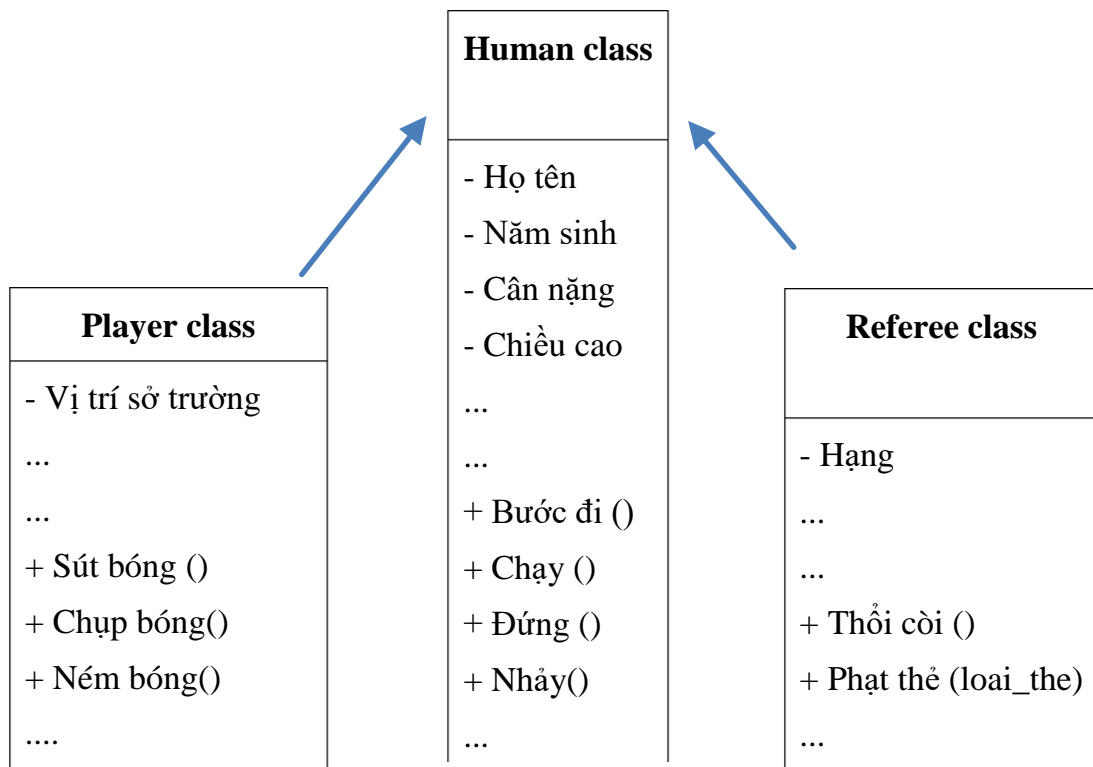
Đặc trưng quan trọng nhất của thiết kế hướng đối tượng là việc sử dụng lại các lớp sẵn có để tạo ra một lớp mới, hay còn gọi là sự kế thừa. Nhờ vậy, các phần mềm sẽ có lợi thế vượt trội trong các dự án gia công phần mềm, từ những sản phẩm hoặc mô-

đun, thành phần riêng biệt sẵn có, được phát triển, tích hợp thành sản phẩm mới hoàn chỉnh.

Car class
<ul style="list-style-type: none">- Tên hãng- Tên kiểu- Số chỗ ngồi- Loại nhiên liệu...+ Nhập tên hãng (ten_hang)+ Nhập tên kiểu (ten_kieu)+ Nhập số chỗ ngồi (so_cho)+ Nhập loại nhiên liệu (nhiem_lieu)...+ Khởi động ()+ Vào số (so)+ Phanh ()+ Tăng ga()+ Giảm ga()...

Hình 7.7 Ví dụ về một lớp đối tượng xe hơi (car class)

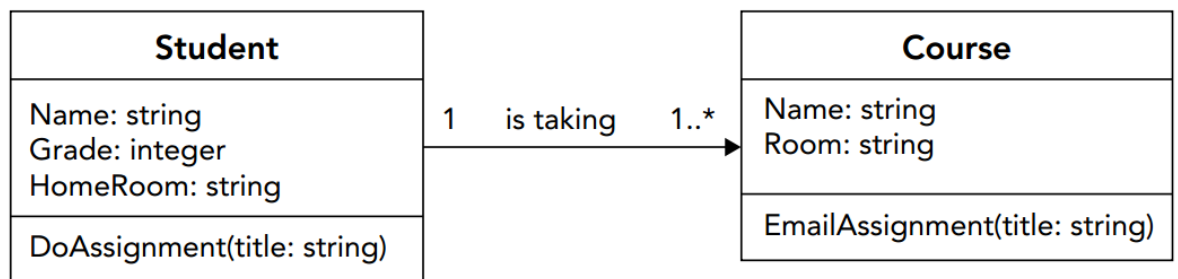
Ví dụ như, trong một trò chơi bóng đá, một số đối tượng như Cầu thủ (player), Trọng tài (referee) có thể được kế thừa từ lớp Người (human) như sau: (Hình 7.8)



Hình 7.8 Một ví dụ về lớp kế thừa

Sơ đồ lớp

Sơ đồ thể hiện mối quan hệ giữa các lớp đối tượng trong hệ thống. Ví dụ như (Hình 7.9),



Hình 7.9 Một mối quan hệ trong sơ đồ lớp (Một sinh viên đang tham gia một hoặc nhiều khóa học)

Sơ đồ trường hợp sử dụng

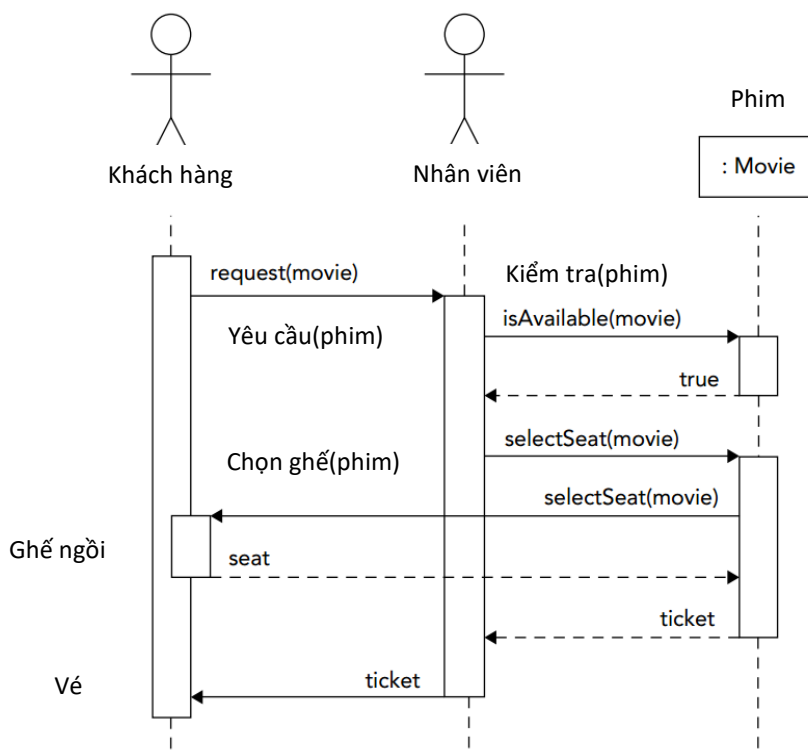
Sơ đồ này được sử dụng để diễn giải các tương tác của người sử dụng với hệ thống. Chẳng hạn như (Hình 7.10),



Hình 7.10 Sơ đồ trường hợp sử dụng thể hiện tương tác giữa khách hàng và hệ thống khi sử dụng chức năng tìm kiếm.

Sơ đồ trình tự

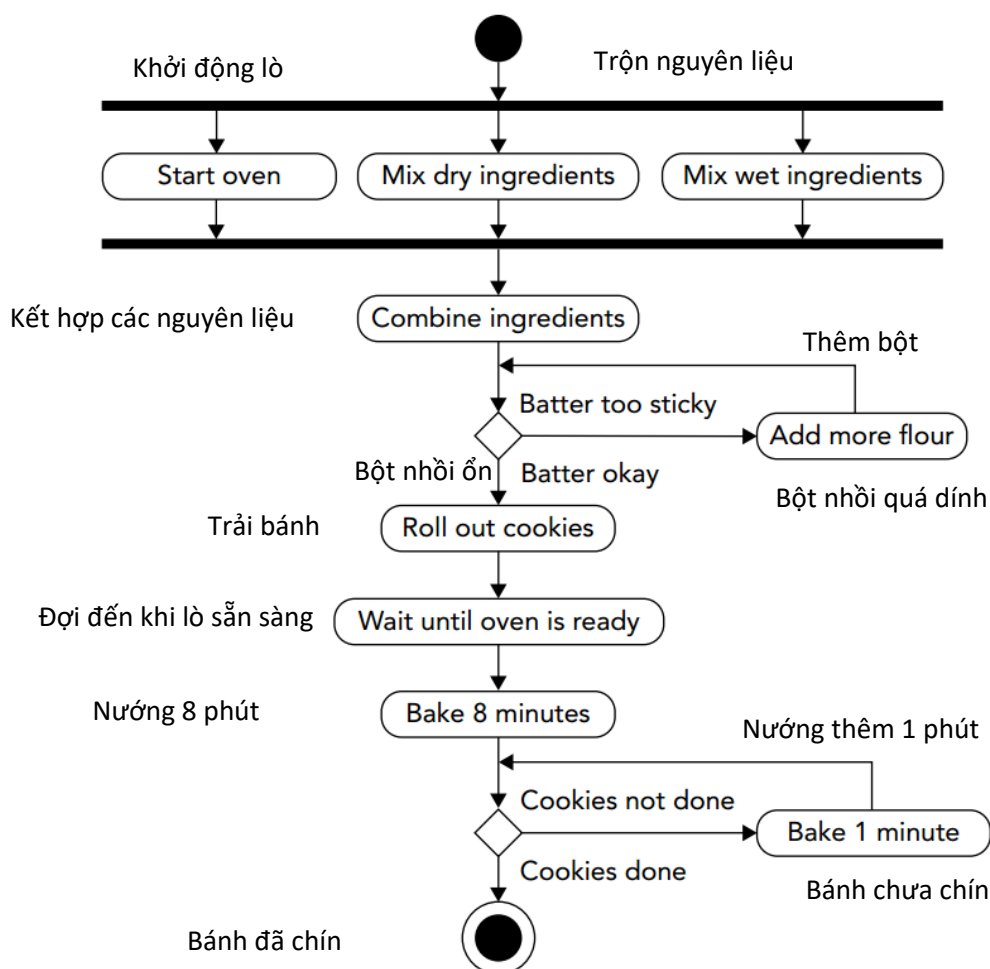
Sơ đồ thể hiện trình tự cộng tác giữa các đối tượng trong một kịch bản, tình huống nhất định thông qua các thông điệp. Ví dụ như (Hình 7.11),



Hình 7.11 Sơ đồ trình tự thể hiện sự cộng tác thông qua chuỗi thông điệp giữa khách hàng, người bán vé và hệ thống.

Sơ đồ hoạt động

Sơ đồ trình bày thứ tự công việc được thực hiện cho một hoạt động cụ thể. Ví dụ như (Hình 7.12),

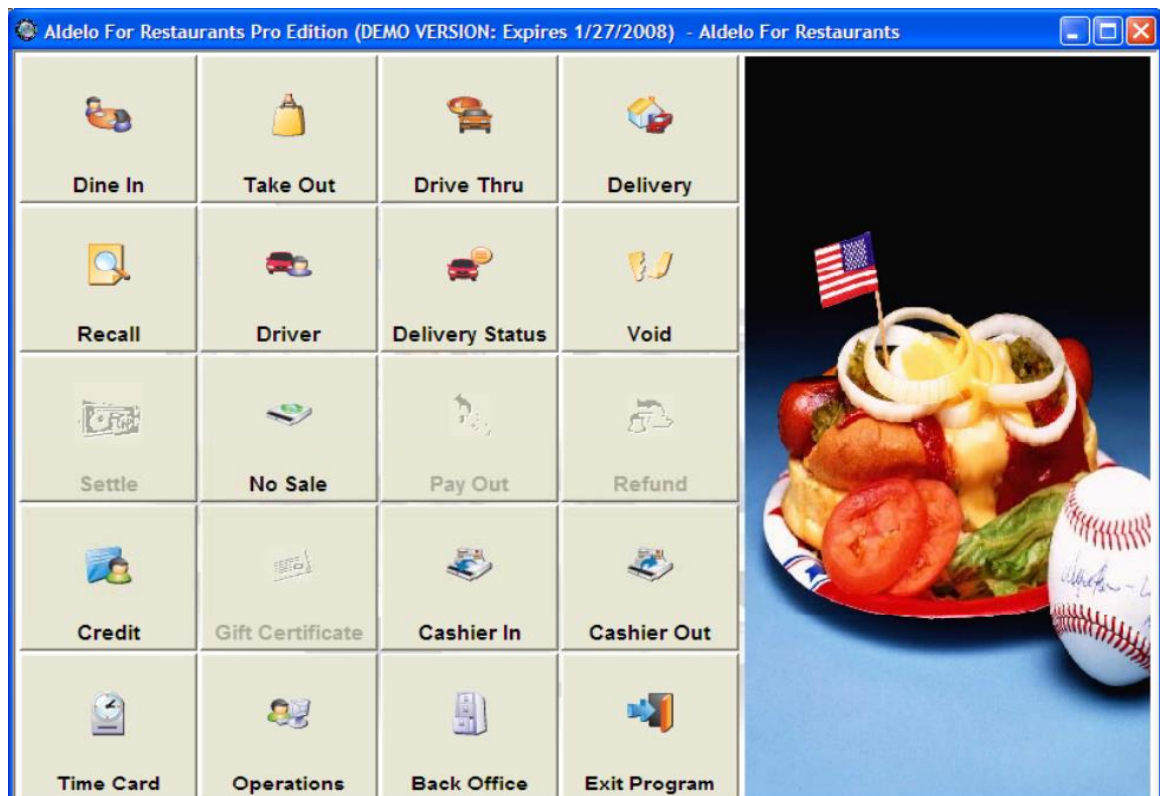


Hình 7.12 Sơ đồ hoạt động diễn giải một quy trình nướng bánh.

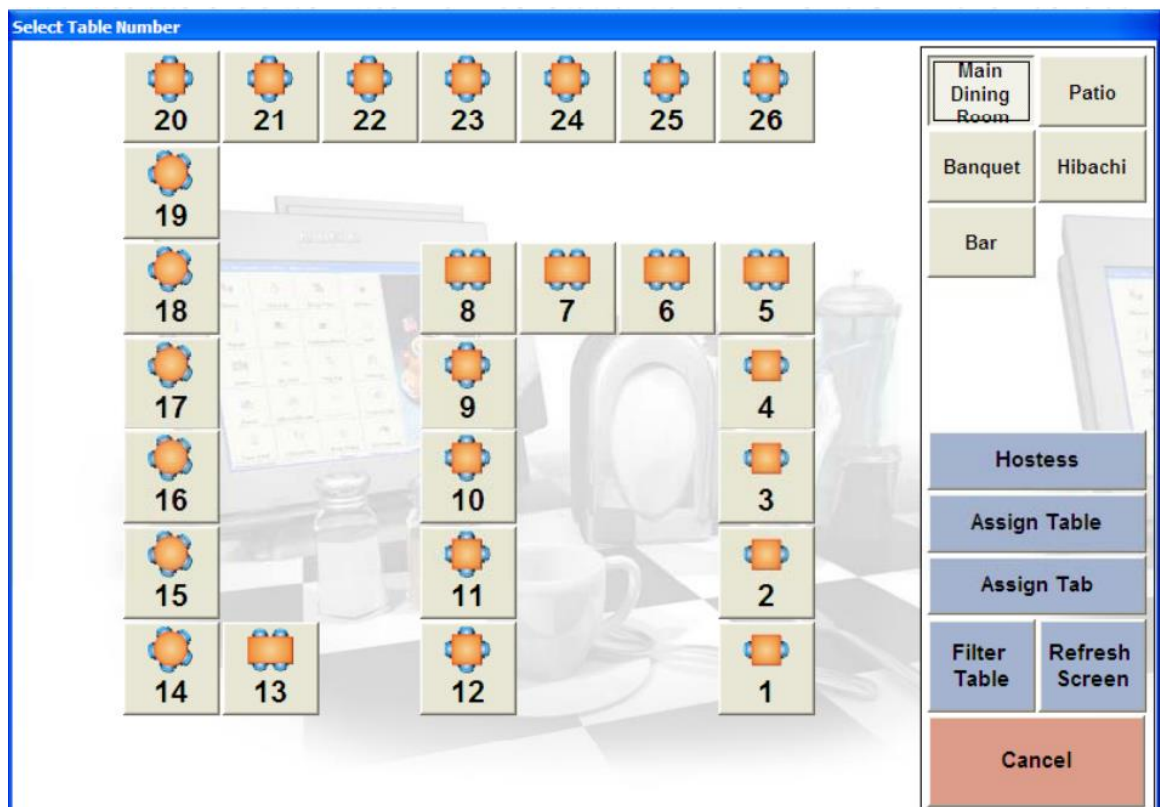
7.4 Thiết kế giao diện

Đóng vai trò làm trung gian giữa phần mềm của máy tính và người sử dụng, giao diện của phần mềm là một sản phẩm quan trọng của quá trình thiết kế nhằm giúp người sử dụng phát huy được tối đa hiệu quả các chức năng của phần mềm. Giao diện về cơ bản được hình thành từ hai yếu tố là trực quan và tương tác. Yếu tố trực quan giúp người sử dụng thấy được các thành phần biểu diễn chức năng. Yếu tố tương tác giúp người sử dụng thao tác được các chức năng đó.


Việc kiểm thử và đánh giá bản thiết kế giao diện có thể áp dụng phương pháp bản mẫu và khảo sát ý kiến của người sử dụng. Các hình sau đây giới thiệu một vài giao diện đặc trưng cho các dạng phần mềm thông dụng.



Hình 7.13 Giao diện chính của một phần mềm quản lý nhà hàng



Hình 7.14 Giao diện chính của một phần mềm quản lý nhà hàng dạng sơ đồ

		Order #: 85 Table #: Table 01 Start time: 05/03/2009 05:30 PM																					
		<table border="1"> <tr> <td>1</td> <td>Lipton Ice tea</td> <td>36.00</td> </tr> <tr> <td>1</td> <td>Nước chanh</td> <td>12.00</td> </tr> <tr> <td>1</td> <td>Bơ Tôm</td> <td>26.00</td> </tr> <tr> <td>1</td> <td>Bơ Value</td> <td>3.00</td> </tr> <tr> <td>1</td> <td>Bơ Zinger</td> <td>50.00</td> </tr> <tr> <td>1</td> <td>Gà Popcorn</td> <td>0.00</td> </tr> <tr> <td>1</td> <td>Cánh Gà Chiên Giòn</td> <td>35.00</td> </tr> </table>	1	Lipton Ice tea	36.00	1	Nước chanh	12.00	1	Bơ Tôm	26.00	1	Bơ Value	3.00	1	Bơ Zinger	50.00	1	Gà Popcorn	0.00	1	Cánh Gà Chiên Giòn	35.00
1	Lipton Ice tea	36.00																					
1	Nước chanh	12.00																					
1	Bơ Tôm	26.00																					
1	Bơ Value	3.00																					
1	Bơ Zinger	50.00																					
1	Gà Popcorn	0.00																					
1	Cánh Gà Chiên Giòn	35.00																					
<div> <div>Settle</div> <div>Void</div> <div>Print Ticket</div> <div>Done</div> <div>Cancel</div> </div>		<table border="1"> <tr> <td>Table charge</td> <td>1,212.35</td> </tr> <tr> <td>Sub Total</td> <td>162.00</td> </tr> <tr> <td>Tax(0.00%)</td> <td>0.00</td> </tr> <tr> <td>Amount Due</td> <td>1,374.35</td> </tr> </table>	Table charge	1,212.35	Sub Total	162.00	Tax(0.00%)	0.00	Amount Due	1,374.35													
Table charge	1,212.35																						
Sub Total	162.00																						
Tax(0.00%)	0.00																						
Amount Due	1,374.35																						

Hình 7.15 Giao diện chính của một phần mềm quản lý dạng đối tượng

Khách Hàng

Lấy Hẹn

Friday, December 05, 2008

4:23:13 PM

English

Xin Chọn Nhân Viên Phục Vụ

T: 0 S: \$0.00



Cindy
(Nguyễn Cao Kỳ Duyên)

T: 0 S: \$0.00



Pam

T: 0 S: \$0.00



Hồng Đào

T: 0 S: \$0.00



Quang Minh



T: 0 S: \$0.00



Harry
(Nguyễn Hưng)

T: 0 S: \$0.00



Khánh Hà

T: 0 S: \$0.00



Jenny
(Ỗ Lan)

T: 0 S: \$0.00



Linda

Chủ Tiệm/
Quản Lý

Sửa Đổi Nhân Viên

Minimize

Hết Ngày

Thối Lại:
\$0.00

www.nailspa123.com

Hình 7.16 Giao diện chính của một phần mềm quản lý dịch vụ Nails

Bài tập

1. Trình bày vai trò của giai đoạn thiết kế trong quy trình phát triển phần mềm.
2. Trình bày đặc trưng của hai phương pháp Merise và hướng đối tượng, chỉ ra những trường hợp nào thuận lợi để áp dụng từng phương pháp. (Gợi ý: dựa vào quy mô dự án và đặc điểm của phần mềm)
3. Sử dụng dữ liệu trả lời của Câu 4 ở Chương 6 về tài liệu đặc tả và áp dụng phương pháp Merise trong bài này để lập một tài liệu thiết kế cụ thể với mô hình Thực thể - Quan hệ.
4. Sử dụng dữ liệu trả lời của Câu 3 nêu trên và áp dụng ngôn ngữ UML để lập một tài liệu thiết kế ít nhất 3 trường hợp sử dụng và 3 quy trình hoạt động.
5. Hãy chọn một hệ thống hướng đối tượng bất kỳ, và thiết kế một trường hợp kế thừa của hai lớp đối tượng trong hệ thống đó.

Chủ đề thảo luận

1. Giả sử thiết kế kiến trúc được bỏ qua, thì những rủi ro nào có khả năng xuất hiện trong những công việc tiếp theo?
2. Phương pháp Merise và phương pháp hướng đối tượng có thể kết hợp với nhau không? Hãy đề xuất cách thức sử dụng kết hợp hai phương pháp này trong cùng một dự án hoặc cùng một quy trình phát triển một sản phẩm.
3. Giao diện phần mềm có vai trò như thế nào trong quá trình giao tiếp giữa con người và máy tính?

CHƯƠNG 8. CÀI ĐẶT

Chương này trình bày các yêu cầu cần thiết trong giai đoạn cài đặt bao gồm công cụ, giải thuật – dữ liệu, kỹ thuật lập trình và nhân lực. Nội dung kiến thức của chương này giúp người học hiểu vai trò của các yếu tố cơ bản giúp việc cài đặt thành công; ngoài ra, người học có thể nhận biết và lựa chọn được các công cụ, phong cách cài đặt sao cho phù hợp với công việc và khả năng của bản thân.

8.1 Giới thiệu

Sau khi mọi thứ đã sẵn sàng bao gồm các tài liệu đặc tả yêu cầu và các tài liệu thiết kế toàn bộ hệ thống, những thành phần đầu tiên của sản phẩm phần mềm sẽ được hình thành theo từng bước, một phiên bản hoàn chỉnh của sản phẩm sẽ ra đời. Giai đoạn cài đặt luôn được đánh giá chiếm nhiều thời gian và kinh phí hơn so với các giai đoạn còn lại trong quy trình phát triển phần mềm. Sở dĩ như vậy, bởi lẽ mức độ phức tạp và tính chuẩn xác trong giai đoạn này là rất lớn. Các yếu tố chính quyết định sự thành công bao gồm việc sử dụng công cụ lập trình, kỹ thuật lập trình, quản lý tài liệu liên quan.

8.2 Công cụ

Máy tính, tất nhiên là loại công cụ đầu tiên mà mọi người trong lĩnh vực này nghĩ đến khi nói về các phương tiện hoặc công cụ cần thiết trong giai đoạn cài đặt. Ngày nay, với sự phát triển nhanh chóng về công nghệ chế tạo máy tính và giá thành tương đối hợp lý và đa dạng với túi tiền của nhiều người, việc lựa chọn và trang bị một cái máy tính phục vụ cho nghề lập trình không phải quá khó khăn. Tuy nhiên, để việc tận dụng máy tính một cách hiệu quả cần lưu ý một số nội dung sau.

Cấu hình của máy tính (kể cả thiết bị để bàn, cầm tay, cố định hoặc di động) dùng để lập trình, phát triển phần mềm và máy tính của người sử dụng nên có cấu hình tương tự. Đối với các dự án phần mềm chuyên nghiệp, điều này được khảo sát kỹ lưỡng để ước tính giá thành đầu tư, tuy nhiên, đối với những dự án nhỏ, nghiệp dư, hầu như chỉ sử dụng những gì đang có. Việc đầu tư các máy tính bên phát triển phần mềm với cấu hình cao hơn máy tính khi triển khai ứng dụng là cần thiết để tiết kiệm thời gian và tăng độ chính xác trong quá trình cài đặt. Tuy nhiên, công đoạn kiểm thử cài đặt lại phải sử dụng máy tính có cấu hình tương tự thậm chí thấp hơn máy tính của

người sử dụng. Điều này giúp ích trong việc xác minh tính bền vững và khả năng thích ứng của sản phẩm.

Hệ thống mạng là yêu cầu thực dụng thứ hai được đề cập. Môi trường làm việc chuyên nghiệp hiện đại không thể thiếu sự kết nối, tuy nhiên, không phải ai cũng biết rằng việc truy cập Internet trong các dự án đặc thù làm điều cấm kỵ. Việc kết nối, chia sẻ trong các nhóm lập trình, giữa các bên liên quan trong dự án là phổ biến, nhưng các kết nối với bên ngoài luôn chứa đựng những rủi ro. Một số nhà quản lý đề phòng tin tặc hoặc đảm bảo an ninh thông tin, một số khác chỉ nhằm quản lý giờ nhân rồi của nhân viên. Về phía các lập trình viên, rất nhiều người thuộc trường phái viết mã lệnh bằng cách chỉ sửa lại và phát triển thêm từ những gì tìm được trên Internet.

Công cụ lập trình là nơi mọi thành phần của sản phẩm phần mềm được sinh ra. Lập trình không đơn giản chỉ là việc soạn thảo những dòng lệnh theo đúng cú pháp, mà nó yêu cầu một loạt các điều kiện sau:

- quản lý mã lệnh
- quản lý lỗi
- quản lý các thành phần, mô-đun
- quản lý dữ liệu
- quản lý phiên bản
- quản lý giao diện...

Do đó, trên thị trường hiện nay cũng đã cung cấp đầy đủ các công cụ phục vụ cũng như hỗ trợ mạnh mẽ công việc lập trình, cài đặt này. Hơn nữa, với tính năng động của cộng đồng lập trình viên, các môi trường cộng tác trực tuyến cũng hỗ trợ đắc lực cho lĩnh vực nghề nghiệp này. Sau đây là một số công cụ phổ biến hiện nay.

Bảng 8.1 Phân nhóm các công cụ hỗ trợ lập trình

Hỗ trợ lập trình với thành phần trực quan	Microsoft Visual Studio Android Studio NetBeans GUI design Adobe DreamWeaver
--	---

	GNUstep
Hỗ trợ kiểm tra mã lệnh	GitLab, Sider, Collaborator
Hỗ trợ tích hợp đa ngôn ngữ	Simplified Wrapper and Interface Generator (SWIG)
Hỗ trợ soạn thảo mã lệnh	Eclipse, Gedit, NetBeans, Notepad++, Visual Studio Code

8.3 Giải thuật & dữ liệu

Giải thuật nào được áp dụng vào mã lệnh cũng là một chủ đề không thể thiếu trong việc quản lý quá trình cài đặt, lập trình. Đây là phần cơ sở quyết định hiệu quả xử lý đầu vào, chất lượng kết quả đầu ra và tốc độ hoàn thành nhiệm vụ. Nếu tiếp cận ở khía cạnh học thuật, thì việc đề ra và lựa chọn giải thuật phù hợp với mục đích sử dụng là công đoạn phức tạp và khó khăn nhất trong quy trình phát triển phần mềm. Giải thuật chỉ đơn giản khi vấn đề đã được nó giải quyết.

Quá trình cài đặt các thành phần hướng đối tượng chịu ảnh hưởng từ hiệu quả của giải thuật ít nghiêm trọng hơn các thành phần hướng thủ tục. Song hành với thiết kế hệ thống hướng đối tượng, thì các giải thuật áp dụng cũng sẽ được cấu trúc theo từng gói khép kín trong từng lớp đối tượng được lập trình. Do đó, việc kiểm soát hiệu quả của giải thuật sẽ dễ dàng hơn.

Một chủ đề vẫn được nhiều người quan tâm là về việc tính phức tạp của giải thuật có tương quan như thế nào với kỹ thuật lập trình khi được cài đặt. Một giải thuật có độ phức tạp càng cao thì quá trình cài đặt sẽ càng khó khăn, cụ thể về số lượng dòng lệnh sẽ lớn hơn và các ràng buộc giữa các dòng lệnh sẽ nhiều hơn. Kỹ thuật lập trình luôn khuyến khích tính rõ ràng, đơn giản trong các dòng lệnh nhằm thuận lợi trong việc theo dõi, kiểm tra và bảo trì. Ở một khía cạnh như vậy, sự phức tạp trong giải thuật không được khuyến khích trong giai đoạn này.

Đối với các sản phẩm phần mềm phục vụ trực tiếp các lĩnh vực nghiên cứu khoa học, thì chất lượng của giải thuật chính là chất lượng của sản phẩm phần mềm. Chẳng hạn như, các phần mềm trong lĩnh vực khai khoáng dữ liệu.

Khi lựa chọn giải thuật áp dụng, thì cấu trúc dữ liệu tương ứng cũng cần được quan tâm. Trước khi công đoạn cài đặt bắt đầu, thì kết quả của giai đoạn thiết kế đã cho ra được mô hình cơ sở dữ liệu của phần mềm. Như vậy, cơ sở dữ liệu và cấu trúc dữ liệu có vai trò như thế nào đối với việc lập trình?

Cơ sở dữ liệu là một nền tảng lưu trữ toàn bộ dữ liệu và ràng buộc ngữ nghĩa, các tiêu chuẩn đảm bảo cho việc vận hành phần mềm cũng như hoạt động ngoại biên của công ty hoặc tổ chức sử dụng phần mềm. Trong khi đó, cấu trúc dữ liệu là quy cách tổ chức dữ liệu phục vụ tức thời cho một chức năng hoặc một yêu cầu nhất định. Do vậy, trong quá trình cài đặt, lập trình có thể nhiều cấu trúc dữ liệu được sử dụng, nhưng nền tảng duy nhất chỉ có một cơ sở dữ liệu.

8.4 Kỹ thuật lập trình

Như đã vừa đề cập ở phần trên, một giải thuật sẽ không phát huy chức năng nếu không được cài đặt tốt. Nói cách khác, kỹ thuật lập trình của lập trình viên sẽ quyết định giá trị thực dụng mà các giải thuật đem lại. Nếu một giải thuật giản đơn, lại được cài đặt một cách rườm rà, thì tốc độ hoàn thành giải thuật sẽ bị ảnh hưởng. Khi một giải thuật phức tạp, nhưng được cài đặt gọn gàng, tinh xảo thì hiệu quả thực thi có thể tốt đẹp hơn. Vậy, việc lập trình như thế nào được gọi là có kỹ thuật?

Một là, trình bày mã lệnh càng dễ hiểu càng tốt.

Máy tính mặc nhiên hiểu rõ những câu lệnh nhờ có trình biên dịch. Tuy nhiên, con người, người lập trình, người bảo trì phần mềm gặp rất nhiều khó khăn khi đọc hiểu những mã lệnh đó. Dễ hiểu, để người lập trình có thể theo dõi, kiểm tra tính đúng đắn của cú pháp, kết quả thực thi. Dễ hiểu, để người bảo trì có thể truy hồi, xác định vấn đề cần được khắc phục trong mã lệnh. Dễ hiểu, để những người không viết ra chương trình đó, vẫn có thể sử dụng để tiếp tục phát triển, mở rộng, cài đặt thêm.

Mã lệnh như thế nào được xem là dễ hiểu?

- Thống nhất trong định dạng: canh lề, thụt đầu dòng, khoảng cách các dòng, đoạn, in đậm, nghiêng...
- Thống nhất trong việc chú thích: vị trí đặt chú thích, quy cách của chú thích, quy cách viết tắt
- Thống nhất trong cách đặt tên biến, hàm...

Hai là, luôn chủ động quản lý lỗi.

Lỗi trong giai đoạn cài đặt được xem như là một rủi ro đương nhiên mà các lập trình viên thường xuyên đối mặt. Có nhiều dạng lỗi như lỗi cú pháp, lỗi chính tả, lỗi cấu trúc, hoặc lỗi do giải thuật,...Lỗi cú pháp và chính tả được các công cụ lập trình hỗ trợ lập trình viên phát hiện và sửa chữa nhanh chóng. Trong khi đó, lỗi do cấu trúc và lỗi do giải thuật thì khó phát hiện và khắc phục hơn. Chẳng hạn như, lỗi cấu trúc vòng lặp, khiến chương trình lặp không dừng. Lỗi do giải thuật, khiến kết quả đầu ra không luôn đúng. Vậy, thế nào là chủ động quản lý lỗi?

- Luôn bắt lỗi ở những nơi có thể
- Luôn quan tâm đến những cảnh báo (không phải lỗi)

Ba là, lập trình thông minh.

- Sử dụng triệt để những lệnh có sẵn
- Không lặp lại các lỗi trước đó
- Không viết lại cùng một đoạn mã nhiều lần
- Hoàn thiện mã lệnh theo thời gian

Bốn là, thường xuyên lập và cập nhật tài liệu cấu trúc chương trình.

Công đoạn cài đặt chuyên nghiệp không thể thiếu tài liệu này. Chúng lưu trữ các thông tin nhằm xác định phiên bản của chương trình, người hoặc đơn vị phụ trách, vị trí của chương trình trong kiến trúc tổng thể hệ thống, chức năng của chương trình và các yêu cầu hoặc điều kiện kết nối với các chương trình hoặc mô-đun khác. Nhờ vậy, tính cộng tác và tính dễ quản lý trong giai đoạn này được phát huy.

8.5 Nhân lực

Dựa theo những yêu cầu như trên, việc chọn lựa nhân sự phù hợp để đảm bảo sự thành công của giai đoạn này có thể dựa trên các tiêu chí sau:

- Đam mê
- Thông minh, sáng tạo
- Khả năng chịu áp lực công việc
- Siêng năng, kiên trì, tỉ mỉ
- Ham học hỏi, trau dồi kiến thức
- Khả năng làm việc theo nhóm

Bài tập

1. Trình bày tầm quan trọng của giai đoạn cài đặt trong quy trình phát triển phần mềm.
2. Kể tên những công cụ hỗ trợ lập trình mà các bạn đang sử dụng, giải thích lý do sử dụng công cụ đó thay vì các công cụ khác cùng chức năng.
3. Hãy tự nhận xét kỹ thuật lập trình của bạn về ưu điểm và hạn chế, và cũng tự nhận xét tính cách bản thân có phù hợp với công việc lập trình không.

Chủ đề thảo luận

1. Công cụ lập trình có tác động như thế nào đến kỹ thuật, phong cách và hiệu quả của việc lập trình?
2. Trong trường hợp lập trình viên gặp khó khăn khi cài đặt giải thuật và cấu trúc dữ liệu, đề xuất những phương án giúp giải quyết trường hợp này.
3. Ngôn ngữ lập trình có vai trò như thế nào đối với chất lượng sản phẩm phần mềm?

CHƯƠNG 9. KIỂM THỬ

Nội dung của chương này trình bày vai trò của giai đoạn kiểm thử và các phương pháp, kỹ thuật thông dụng trong hoạt động kiểm thử. Sau khi học xong, người học có thể hiểu, trình bày được đặc điểm của các kiểu kiểm thử, có thể sử dụng cơ bản các kỹ thuật kiểm thử trong quá trình cài đặt và có thể tổ chức được hoạt động kiểm thử trong các dự án nhỏ.

9.1. Giới thiệu

Kiểm thử là quá trình xác minh một hệ thống phần mềm hoạt động có đúng với đặc tả, thiết kế không và có đáp ứng đầy đủ yêu cầu mà khách hàng mong đợi hay không. Hai mục đích chính của quá trình kiểm tra gồm:

Một là, chứng minh cho các bên tham gia dự án rằng sản phẩm đáp ứng đúng với các yêu cầu đặt ra.

Hai là, tìm ra những trường hợp mà phần mềm hoạt động không đúng đắn, nói cách khác tìm ra lỗi chưa được phát hiện và bảo đảm rằng lỗi sẽ được sửa.

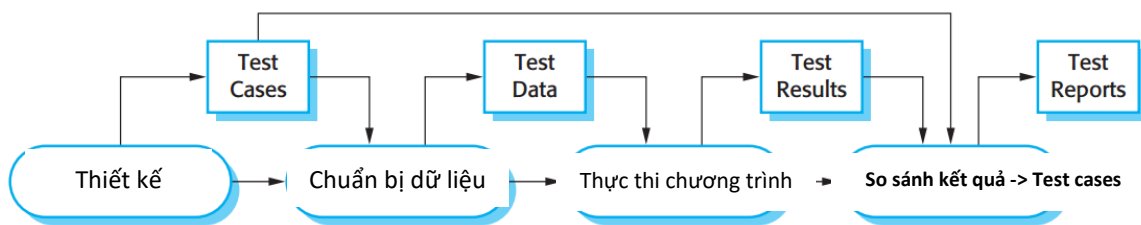
Những khiếm khuyết của phần mềm có thể dẫn đến những hậu quả nghiêm trọng khi sản phẩm đưa vào sử dụng, chẳng hạn như hệ thống ngừng trệ, tương tác ngoài ý muốn với các hệ thống tích hợp khác, việc tính toán sai lệch hoặc dữ liệu bị hư hỏng.

Hoạt động kiểm thử phần mềm có thể được chia làm hai nhóm gồm kiểm thử đáp ứng yêu cầu và kiểm thử lỗi. Bên cạnh đó, nếu tiếp cận theo quy trình phát triển thì hoạt động kiểm tra có ba giai đoạn gồm kiểm tra trong giai đoạn cài đặt, kiểm tra trước khi bàn giao và kiểm tra sau khi bàn giao.

9.2 Kiểm thử đáp ứng yêu cầu

Kiểm thử đáp ứng yêu cầu thường hướng đến khách hàng, vì họ không hoặc ít quan tâm các yếu tố kỹ thuật vận hành bên trong phần mềm, cũng như việc phần mềm có thể xuất hiện lỗi trong quá trình hoạt động. Đối với khách hàng hoặc người sử dụng, điều quan trọng là phần mềm có phục vụ đúng mục đích mà họ mong muốn hay không. Như vậy kiểm thử đáp ứng yêu cầu có vai trò kiểm tra tổng thể, kiểm tra các chức năng đầu cuối so với bản đặc tả yêu cầu ban đầu.

Quy trình cơ bản của một lượt kiểm thử như sau (Hình 9.1):



Hình 9.1 Quy trình cơ bản của một đợt kiểm thử (Sommerville, I. 2011)

Nhằm phục vụ cho quá trình kiểm thử, các trường hợp kiểm thử (Test cases) được thiết kế một cách cẩn thận, đảm bảo các yêu cầu về tính hợp lý của dữ liệu đầu vào so với yêu cầu xử lý của hệ thống, cấu hình vận hành của hệ thống phải phù hợp để xử lý dữ liệu đầu vào, kết quả đầu ra được so sánh với kỳ vọng ban đầu của trường hợp kiểm thử, và công việc quan trọng sau cùng là lập tài liệu báo cáo.

Cơ sở để thiết kế các trường hợp kiểm thử được dựa vào các tài liệu đặc tả và tài liệu thiết kế. Hai bộ tài liệu này vừa thể hiện yêu cầu của khách hàng vừa trình bày chi tiết yêu cầu cho hệ thống một cách đầy đủ nhất. Tùy thuộc vào thời điểm hoặc giai đoạn thực hiện kiểm thử mà người phụ trách thiết kế các trường hợp có thể là lập trình viên, kiểm tra viên, có thể là chính bản thân người sử dụng.

9.3 Kiểm thử lỗi

Không giống hoàn toàn như hình thức kiểm thử đáp ứng yêu cầu, kiểm thử lỗi sẽ tập trung hơn vào các yếu tố kỹ thuật vận hành bên trong phần mềm. Các chức năng thành phần, các mô-đun, các gói tích hợp,... Như vậy, các trường hợp kiểm tra sẽ được thiết kế với kỳ vọng hệ thống sẽ bộc lộ ra các khiếm khuyết, thay vì kỳ vọng đáp ứng yêu cầu khách hàng. Ví dụ một trường hợp đơn giản là kiểm tra thực hiện phép chia giữa số bị chia B và số chia C . Vậy, khi gán $C = 0$, kỳ vọng hệ thống sẽ thông báo lỗi, hệ thống bị treo hoặc kết quả sai lệch được trả về.

Vai trò của hoạt động kiểm thử này không phải nhằm đảm bảo phần mềm không có lỗi, mà nhằm phát hiện ra nhiều nhất những lỗi trong khả năng của hoạt động kiểm thử. Trên thực tế, các sản phẩm phần mềm thương mại trước khi xuất xưởng đều đã vượt qua công đoạn này, và mọi người đều phải thừa nhận rằng hầu như không tồn tại sản phẩm không có khiếm khuyết.

Hai kỹ thuật kiểm thử được sử dụng phổ biến là kiểm thử hộp đen (Black box) và kiểm thử hộp trắng (White box).

9.3.1 Kỹ thuật kiểm thử hộp đen

Kiểm thử hộp đen có tên gọi khác là kiểm tra hướng dữ liệu (Data - driven) hoặc là kiểm tra hướng vào/ra (Input/Output driven). Trong kỹ thuật này, người kiểm tra hoàn toàn không quan tâm đến sự vận hành bên trong của phần mềm, mà tập trung tìm ra các tình huống mà phần mềm không trả về kết quả theo đúng đắn như đặc tả. Do đó, tài liệu đặc tả sẽ là cơ sở chủ yếu giúp người kiểm tra lập ra kế hoạch sử dụng dữ liệu một cách tốt nhất.

9.3.2 Kỹ thuật kiểm thử hộp trắng

Kiểm thử hộp trắng chú trọng kiểm tra cấu trúc và sự vận hành bên trong của phần mềm. Với kỹ thuật này, người kiểm tra cần phải hiểu rõ các thành phần như cấu trúc dữ liệu, giải thuật, các ràng buộc...được cài đặt như thế nào. Từ đó, các trường hợp kiểm tra được thiết kế sao cho sự vận hành bộc lộ ra những sai sót, khiếm khuyết. Tài liệu thiết kế và cài đặt sẽ hữu ích trong kỹ thuật này.

Hai kỹ thuật này có thể hỗ trợ lẫn nhau và luôn xuất hiện xuyên suốt các giai đoạn kiểm thử.

9.4 Kiểm thử giai đoạn

9.4.1 Giai đoạn cài đặt

Song hành trong giai đoạn cài đặt, công đoạn kiểm tra thường xuyên được thực hiện với sự tham gia của những người đã thiết kế hệ thống và những lập trình viên. Không phải bắt buộc cả hai nhóm đều tham gia, nhưng lập trình viên sẽ được ưu tiên kiểm tra trước. Thực tế, trong suốt quá trình viết mã lệnh, những lập trình viên đã thực hiện nhiệm vụ này. Đối với những người thiết kế hệ thống, họ sẽ tiến hành kiểm tra độc lập, kể đến là phối hợp với lập trình viên để tìm ra phương án giải quyết các lỗi hoặc khiếm khuyết được phát hiện.

Trong giai đoạn cài đặt có thể triển khai ba cấp độ kiểm tra, như sau:

- Cấp độ đơn vị (Unit testing): những đơn vị chương trình độc lập hoặc những lớp đối tượng sẽ được kiểm tra chức năng hoạt động.
- Cấp độ thành phần (Component testing): những thành phần tích hợp từ những đơn vị độc lập sẽ được kiểm tra sự kết nối và chức năng tương tác.

- Cấp độ hệ thống (System testing): toàn bộ hệ thống sẽ được kiểm tra chức năng hoạt động và tương tác (trường hợp có các hệ thống liên kết).

9.4.2 Giai đoạn bàn giao

Nhằm đảm bảo mọi thứ đã hoàn hảo và sẵn sàng lên kệ trên thị trường hoặc đến tay khách hàng, người sử dụng. công đoạn kiểm thử xuất xưởng hoặc cho phát hành (release testing) được tiến hành. Đặc trưng của công đoạn này là việc kiểm tra được tiến hành hoàn toàn độc lập với bên phát triển phần mềm và chỉ tập trung vào kiểm tra đáp ứng yêu cầu. Mục đích của kiểm tra xuất xưởng thường nhằm minh chứng và thuyết phục nhà cung cấp sản phẩm về việc đảm bảo sản phẩm đã đáp ứng hoặc thỏa mãn toàn bộ các yêu cầu từ phía khách hàng, người sử dụng, yêu cầu kỹ thuật bên trong sản phẩm và các yêu cầu khác của công ty, tổ chức liên quan. Do đó, các kiểm tra viên sẽ chỉ sử dụng kỹ thuật kiểm tra chứng năng dựa trên kết quả đầu ra của hệ thống thông qua dữ liệu hoặc cấu hình đầu vào. Đây được gọi là kỹ thuật kiểm thử hộp đen (Black box).

Bên cạnh kiểm thử toàn bộ các yêu cầu, có hai dạng kiểm thử khác cũng thường được thực hiện, bao gồm:

- Kiểm thử các tình huống sử dụng: thông thường khi kiểm tra các yêu cầu, các chức năng được thực hiện tuần tự để kiểm tra. Tuy nhiên, việc thực hiện một tổ hợp các chức năng có trật tự nhất định vẫn có thể dẫn đến những phát sinh ngoài dự kiến. Chẳng hạn như, một ứng dụng giải trí có ba chức năng để điều khiển âm thanh A, B và C. Khi A và B được khởi động trước, thì mức cực đại của âm lượng đạt chuẩn. Tuy nhiên nếu C được khởi động trước, thì mức cực đại của âm lượng không đạt yêu cầu.

- Kiểm thử sự ổn định: hiệu suất hoạt động của một hệ thống phần mềm có thể bị ảnh hưởng bởi sự hỗ trợ của các thiết bị phần cứng hoặc môi trường xung quanh. Chẳng hạn như, các phần mềm trực tuyến hoặc các ứng dụng web sẽ dễ bị ảnh hưởng bởi chất lượng đường truyền và lưu lượng kết nối. Các trường hợp kiểm tra được thiết kế phải bao gồm các định mức tiếp cận giới hạn khả năng đáp ứng của hệ thống. Chẳng hạn như, thông số kỹ thuật chỉ ra khả năng cực đại của hệ thống là đáp ứng 1000 kết nối tại một thời điểm. Thực tế, hệ thống đã được xây dựng với mức đáp ứng cao hơn, 1100 chẳng hạn. Với các định mức tăng dần được áp dụng, hệ thống sẽ được

ghi nhận tình trạng hiệu suất hoạt động thay đổi tương ứng theo thời gian. Tính ổn định của hệ thống sẽ được phản ánh dựa vào kết quả đó.

9.4.3 Kiểm thử từ người sử dụng

Đã đến lúc sản phẩm phần mềm đến được tay người sử dụng, tuy nhiên, công việc kiểm tra vẫn chưa kết thúc. Khách hàng hoặc người sử dụng có vai trò sau cùng trong việc hỗ trợ hoàn thiện sản phẩm. Vậy việc này có cần thiết không trong khi sản phẩm đã trải qua hàng loạt cuộc kiểm tra nghiêm ngặt?

Đúng là đã có hàng loạt cuộc kiểm tra nghiêm ngặt, tuy nhiên tất cả các điều kiện, môi trường và tình huống kiểm tra đều có sự chuẩn bị, sắp xếp hay nói cách khác là không tự nhiên và chỉ là giả lập. Ví dụ như, một hệ thống phần mềm phục vụ trong bệnh viện chỉ được kiểm tra trong một mô hình giả lập trước khi bàn giao.

Có ba cấp độ kiểm thử cùng với người sử dụng:

- Phiên bản Alpha (Alpha testing):

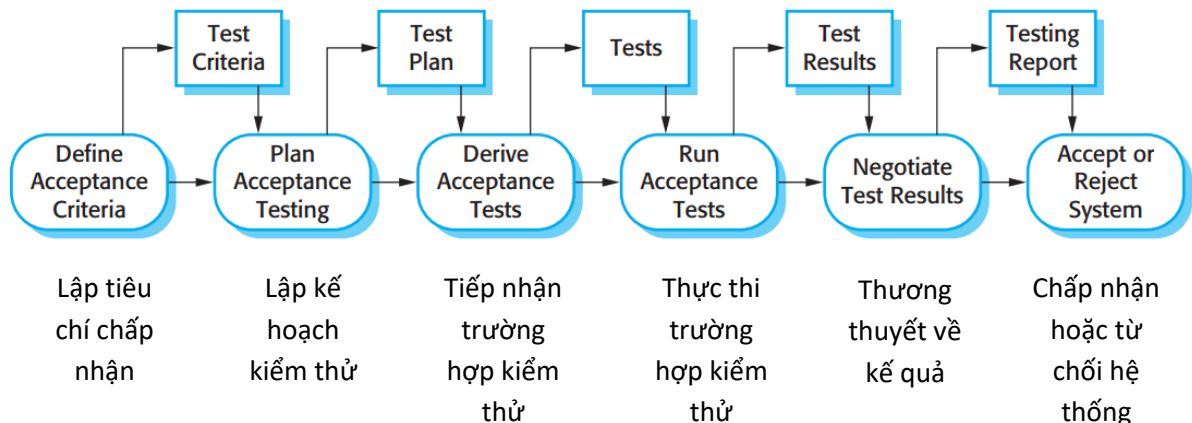
- Người sử dụng cùng với bên phát triển phần mềm sẽ cộng tác kiểm tra sản phẩm trong môi trường giả lập
- Các chức năng và tình huống sử dụng phần mềm được chú trọng kiểm tra
- Kỹ thuật kiểm tra hộp trắng và hộp đen có thể phù hợp
- Những vấn đề và lỗi sẽ được sửa chữa song song
- Giúp nhận biết những trường hợp sử dụng phù hợp với từng nghiệp vụ như thế nào

- Phiên bản Beta (Beta testing):

- Sản phẩm được kiểm tra bởi người sử dụng trong môi trường thực sự
- Tất cả các chức năng, tình huống sử dụng, sự ổn định, an toàn...đều được kiểm tra một cách tự nhiên
- Chỉ có kỹ thuật kiểm tra hộp đen phù hợp
- Đã trải qua phiên bản alpha
- Các lỗi và vấn đề cần khắc phục sẽ được cập nhật trong phiên bản kế tiếp

- Giúp đánh giá mức độ thành công của sản phẩm thông qua sự hài lòng của khách hàng
- Kiểm thử chấp nhận (Acceptance testing)

Đây là công đoạn kiểm thử cuối cùng từ phía người dùng và khách hàng. Kết quả này quyết định việc bàn giao và tiếp nhận hoàn toàn sản phẩm, thanh toán và hoàn thành hợp đồng, chuẩn bị chuyển qua giai đoạn bảo hành, bảo dưỡng. Tổng quát công đoạn này có thể thực hiện như sau



Hình 9.2 Một quy trình cơ bản của kiểm thử chấp nhận (Sommerville, I. 2011)

Bài tập

1. Trình bày vai trò của công đoạn kiểm tra trong quy trình phát triển phần mềm.
2. Trình bày các đặc trưng của hai kỹ thuật kiểm tra hộp trắng và kiểm tra hộp đen, kể ra ít nhất 3 trường hợp cụ thể áp dụng cho từng kỹ thuật, không bao gồm nội dung bài học.
3. Sử dụng dữ liệu các bảng đặc tả yêu cầu của phần mềm điều khiển máy pha cà phê ở Chương 6 và dựa vào quy trình thiết kế trường hợp kiểm tra (Hình 9.1), hãy thiết kế ít nhất 2 trường hợp kiểm tra cụ thể.
4. Hãy tìm hiểu ít nhất 3 phần mềm có áp dụng phương pháp kiểm tra bởi người sử dụng với các phiên bản alpha, bê-ta, hiện nay hoặc trước đây.

Chủ đề thảo luận

1. Công cụ hỗ trợ lập trình các bạn đang sử dụng có khả năng giúp kiểm tra ở cấp độ nào? Hãy chia sẻ kinh nghiệm của bạn trong việc sử dụng các chức năng đó.

2. Kỹ thuật lập trình có tác động như thế nào đối với công việc kiểm tra cài đặt?

3. Hãy tham gia một chương trình sử dụng bản bê-ta của một sản phẩm cụ thể, và chia sẻ trải nghiệm về cách thức tham gia, sử dụng và phản hồi thông tin.

CHƯƠNG 10. BÀN GIAO & BẢO TRÌ

Chương này tập trung trình bày các yêu cầu cần thiết và những vấn đề thường gặp trong giai đoạn bàn giao – bảo trì. Kiến thức chương này giúp người học hiểu vai trò của hoạt động bàn giao – bảo trì, xác định được những rủi ro tiềm năng và có thể tổ chức được hoạt động bàn giao - bảo trì trong các dự án đơn giản.

10.1 Giới thiệu

Trong quy trình phát triển phần mềm, công đoạn bàn giao có thể độc lập sau cùng hoặc có thể song hành với công đoạn kiểm tra bởi người sử dụng. Đến thời điểm này, có vẻ như mọi việc đã thuận lợi khi việc thanh toán hoàn tất và khách hàng tiếp nhận sản phẩm. Tuy nhiên, do đặc thù của sản phẩm phần mềm, trách nhiệm của phía phát triển và cung cấp sản phẩm vẫn còn tiếp tục gắn với khách hàng mặc dù việc thanh toán đã hoàn tất.

Bàn giao là công đoạn giúp khách hàng sở hữu và sử dụng thật sự sản phẩm tại nơi họ mong muốn. Các công việc chính cần thực hiện bao gồm cài đặt và hướng dẫn sử dụng.

Như vậy, mọi việc đã kết thúc chưa?

Vòng đời của một sản phẩm phần mềm chỉ kết thúc khi phần mềm đó không còn được sử dụng hoặc công việc bảo trì chấm dứt. Như vậy, bảo trì sẽ là công đoạn sau cùng. Mục đích của bảo trì không gì khác ngoài việc duy trì sự vận hành của phần mềm một cách tốt nhất. Công việc chủ yếu bao gồm khắc phục lỗi, làm cho phần mềm thích ứng với môi trường, và cải thiện các chức năng.

10.2 Bàn giao

10.2.1 Tiếp cận sản phẩm mới

Khách hàng là thượng đế, bởi vì khách hàng luôn được đáp ứng hoặc thỏa mãn các yêu cầu họ đưa ra. Họ có thể thích thú, thoải mái tiếp nhận phần mềm, hoặc không như vậy. Trong thực tế, nhiều dự án phần mềm gặp nhiều khó khăn khi triển khai các giải pháp mới đến với người sử dụng.

Một giải pháp mới hoàn toàn có thể gây sốc cho khách hàng và người sử dụng. Nguyên nhân có thể nằm trong số sau đây:

- Thay đổi thói quen: suy nghĩ, cách làm việc...
- Thời gian để thích ứng: sự đầu tư cho thói quen mới
- Tính phù hợp với người sử dụng: nhận thức, trình độ...

Điều này lý giải tại sao nhiều giải pháp có chất lượng nhưng khi triển khai lại có hiệu quả ứng dụng ban đầu thấp. Chẳng hạn như, các giải pháp phục vụ trong khu vực chính phủ điện tử (E-Government).

Các phương án sau đây có thể giúp cải thiện vấn đề trên:

- Triển khai thử trên môi trường giả lập
- Triển khai song song một số người dùng giải pháp mới và một số người vẫn dùng hệ thống cũ
- Triển khai cho tất cả người dùng theo từng bước từ mỗi chức năng đơn giản đến sự phức tạp sau cùng

10.2.2 Kế hoạch bàn giao

Để công việc triển khai cài đặt thuận lợi, hạn chế các vấn đề phát sinh, kế hoạch bàn giao cần đảm bảo các bước sau:

- Môi trường: cơ sở vật chất liên quan như không gian phòng, vị trí các thiết bị trong phòng, nguồn điện, nguồn sáng, vị trí các cửa,...
- Phần cứng: máy tính để bàn, laptop, máy chủ, các hệ thống dây cáp, thiết bị mạng, các thiết bị lưu trữ, ...
- Tài liệu chuyển giao: tập huấn, hướng dẫn sử dụng, trợ giúp khẩn cấp, tài liệu an toàn lao động...
- Tập huấn: thời gian, địa điểm, nội dung, nhân sự...
- Cơ sở dữ liệu: tập trung hay phân tán, máy cá nhân hay máy chủ...
- Các ứng dụng hoặc hệ thống có liên quan: các ứng dụng hỗ trợ, các hệ thống giao tiếp khác,...và đừng quên quan tâm đến hệ điều hành.

Những nội dung trong kế hoạch này có thể đã được chuẩn bị trên giấy tờ, trong các điều khoản hợp đồng. Tuy nhiên, nhiều dự án đến khi triển khai cài đặt sản phẩm, mới nhận thấy sự thiếu sót không đáng có.

10.2.3 Một số lưu ý trong công đoạn bàn giao

- Tuyệt đối không giả sử rằng mọi việc rồi sẽ thuận lợi: chỉ tin vào những việc thực làm và đã kiểm tra. Đã có những trường hợp trong quá trình triển khai phần mềm, phát hiện một vài thông số kỹ thuật chưa đạt yêu cầu, nhưng không có kế hoạch khắc phục, với tư duy rằng có thể không ảnh hưởng gì, rồi sẽ tốt thôi. Đây là không phải phong cách làm việc chuyên nghiệp, và không được khuyến khích.

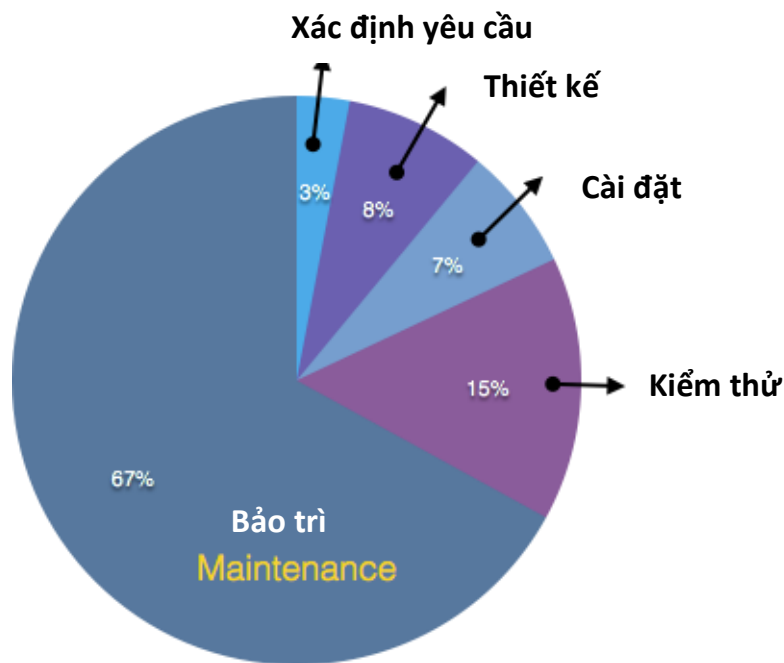
- Luôn có kế hoạch dự phòng: những dự án có liên quan đến triển khai phần cứng, các thiết bị thì càng phải quan tâm đến phương án dự phòng. Chẳng hạn như, các gói thiết bị đã được giao đến nơi cài đặt đúng thời gian. Tuy nhiên, sự cố phát sinh ở nơi triển khai, có thể làm hư hao thiết bị nếu tiếp tục. Trong tình huống này, sự cố chấp có thể dẫn đến thiệt hại, và rắc rối về thủ tục pháp lý. Trong hợp đồng triển khai gói thiết bị hỗ trợ phần mềm cũng không đề cập trách nhiệm cụ thể trong mọi tình huống. Do đó, luôn phải có phương án dự phòng. Thời gian là phương án dự phòng linh hoạt nhất, kể đến nên có nhiều phương án được nêu rõ trong hợp đồng nếu có thể trước khi ký kết.

- Hạn chế gián đoạn công việc của khách hàng: trong nhiều tình huống nghiệp vụ của khách hàng hoặc người sử dụng bị gián đoạn hoặc ảnh hưởng do quá trình bàn giao. Như vậy, để giảm thiểu vấn đề đó, việc lập thời gian biểu phù hợp với công việc bàn giao, cùng phối hợp thông tin đầy đủ với khách hàng, người sử dụng cần được quan tâm cẩn thận. Vậy nên bàn giao nhiều trong một thời gian ngắn hay nên bàn giao ít trong một thời gian dài?

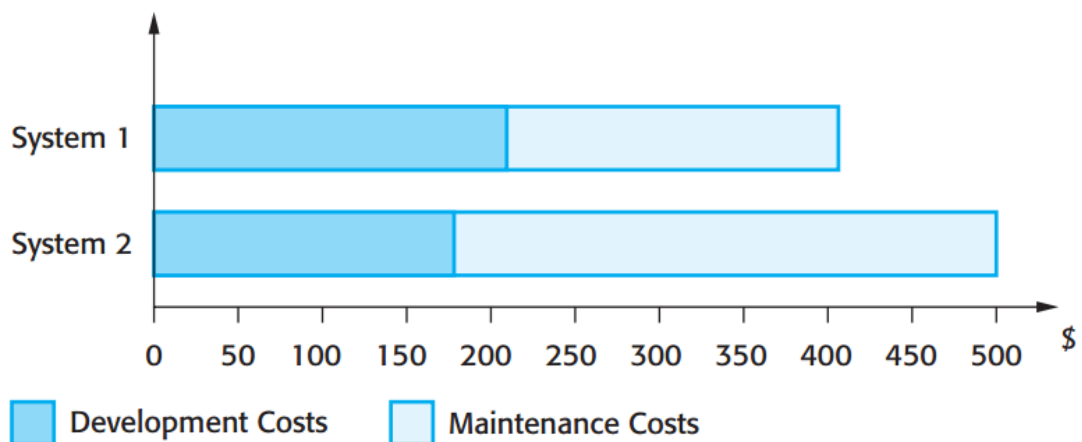
- Nên có điểm phục hồi để tránh cài đặt lại từ đầu: các hệ thống hoặc phần mềm phức tạp thường sẽ được cài đặt nhiều bước, nhiều giai đoạn. Nhằm tránh lãng phí thời gian và ảnh hưởng lớn đến hệ thống khi sự cố xảy ra, trong thiết kế quy trình cài đặt cần có các điểm phục hồi.

10.3 Bảo trì

Tuy đây là giai đoạn sau cùng trong quy trình phát triển, nhưng sự phức tạp có thể mới chỉ bắt đầu. Dựa theo các kết quả nghiên cứu và báo thống kê, kinh phí chi tính trong giai đoạn bảo trì có thể bằng hoặc lớn hơn cả tổng kinh phí phát triển phần mềm ở các giai đoạn trước đó (Hình 10.1 và 10.2).



Hình 10.1 Biểu đồ tỉ lệ chi phí trong quy trình phát triển phần mềm



Hình 10.2 Biểu đồ so sánh chi phí phát triển và bảo trì (Sommerville, I. 2011)

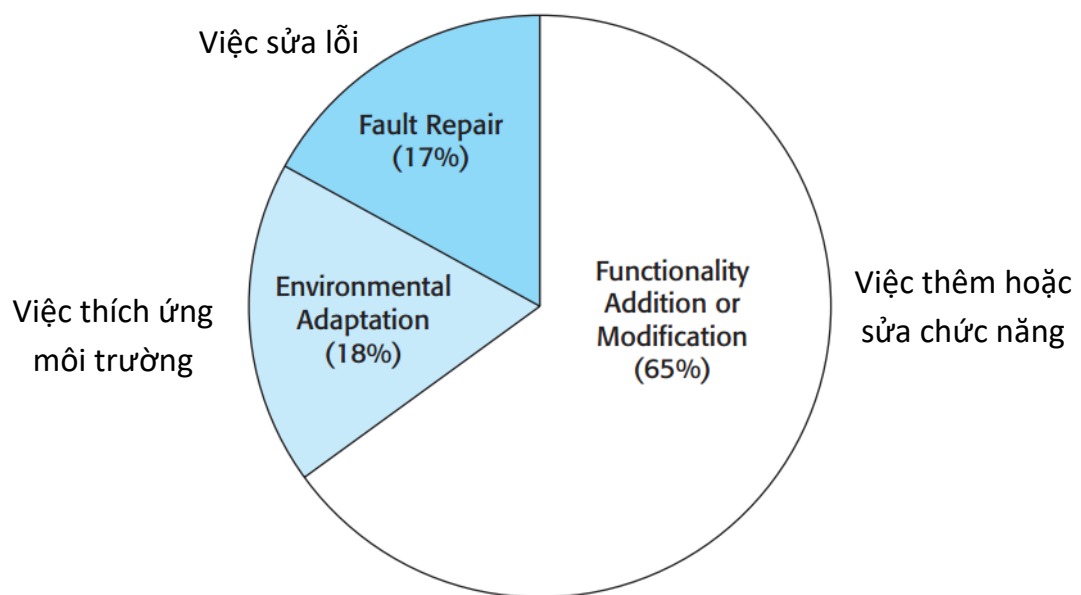
Như đã đề cập ở trên, trong ba công việc chính trong giai đoạn bảo trì thì việc sửa chữa các lỗi, khiếm khuyết của phần mềm cần chi phí không nhiều so với hai việc còn lại.

Việc đảm bảo hệ thống thích ứng với những thay đổi do môi trường chiếm chi phí thứ cao thứ nhì. Các cấu hình về phần cứng, hệ điều hành, các quy chuẩn về dữ liệu hoặc các ràng buộc mới trong quy định nghiệp vụ...sẽ ảnh hưởng trực tiếp đến sự vận hành của phần mềm, và điều đương nhiên phải làm là điều chỉnh để thích nghi. Các chi phí liên quan không hề nhỏ vì sự cần thiết can thiệp vào mã nguồn, cài đặt hệ

thống, các giải thuật hoặc cấu trúc dữ liệu...Đến đây, sự phức tạp đã lộ diện. Mọi người thường nói rằng sửa chữa một căn nhà sẽ khó khăn gấp nhiều lần xây một căn nhà mới. Điều này đúng với cả sản phẩm phần mềm.

Nhân sự dành riêng cho giai đoạn bảo trì thường không được ưu tiên, do nguồn lực nhất định trong các công ty. Ưu tiên hàng đầu vẫn là các dự án mới hoặc đang tiến hành. Mặc dù các công ty luôn cố gắng giữ uy tín, nhưng sự tập trung trong giai đoạn bảo trì là không đáng kể. Đây cũng là một yếu tố đẩy chi phí bảo trì lên cao.

Chưa dừng lại ở mức hao phí này, chi phí cao nhất thuộc về công đoạn cải thiện và bổ sung thêm các chức năng mới vào hệ thống. Một giải pháp được áp dụng thành công sẽ giúp cho tổ chức, công ty phát triển hơn, từ đó, các mối quan hệ mới và các nghiệp vụ mới phát sinh. Vì vậy, nhu cầu thay đổi hoặc bổ sung thêm chức năng vào hệ thống là điều dễ hiểu.



Hình 10.3 Biểu đồ tỉ lệ chi phí trong giai đoạn bảo trì (Sommerville, I. 2011)

Tính mới của chức năng quyết định chi phí tương ứng. Đối với các chức năng đã được cài đặt sẵn thành một mô-đun hoặc một gói lưu trữ trong kho phía phát triển phần mềm, thì chi phí sẽ dễ xác định và thấp hơn những trường hợp còn lại. Do đó, tầm nhìn của nhà đầu tư trong các dự án phần mềm nếu tốt sẽ tiết kiệm được nguồn tài chính đáng kể về sau.

Những vấn đề nêu trên dẫn đến hai ý kiến hiện nay vẫn còn được tranh luận, đó là:

- Chi phí cho giai đoạn bảo trì được tính chung hoặc tách biệt khỏi dự án
- Thời hạn sử dụng của sản phẩm phần mềm có hay không

Một số việc có thể thực hiện nhằm giúp giảm thiểu phát sinh cho phần mềm trong tương lai:

- Dự đoán những thay đổi: những thay đổi được dự đoán sẽ có phương án dự trữ được thiết kế sẵn, và sẽ không có bất ngờ nào xảy ra. Ví dụ như, một hệ thống phần mềm quản lý đào tạo, có thể phải dự đoán về những thay đổi trong quy chế đào tạo trong từng năm, từng thời kỳ. Các mô-đun áp dụng quy chế cần được thiết kế động để cập nhật và bổ sung trong tương lai.

- Các hợp đồng phụ sau phát triển: các hợp đồng phục vụ cho bảo trì được ký kết độc lập với dự án phát triển sẽ giúp bên phát triển phần mềm hoặc quản lý dự án chủ động trong điều phối nhân sự, tập trung nguồn lực tốt hơn, hạn chế sự thụ động trong giai đoạn này.

Bài tập

1. Công đoạn bàn giao sản phẩm phần mềm đến khách hàng và người sử dụng có thật sự cần thiết? Vì sao?
2. Trình bày vai trò của giai đoạn bảo trì trong quy trình phát triển phần mềm.
3. Giải thích nguyên nhân tình trạng chi phí cho giai đoạn bảo trì có thể cao hơn so với chi phí các giai đoạn khác.

Chủ đề thảo luận

1. Nhóm nhân sự nào nên phụ trách công đoạn bảo trì bao gồm khắc phục lỗi và duy trì sự thích ứng môi trường, trong trường hợp một dự án chuyên nghiệp có đầy đủ các tài liệu đặc tả, thiết kế, cài đặt...? Đề xuất các phương án nhân sự phù hợp trong các tình huống tương tự.
2. Chi phí bảo trì không bao gồm phát triển chức năng mới, nên được bao gồm trong giá thành sản phẩm (coi như bảo trì miễn phí) hay được chi trả riêng trong quá trình bảo trì? Phân tích hai trường hợp về thuận lợi và hạn chế, đề xuất phương án phù hợp cho các dự án.

CHƯƠNG 11. QUẢN LÝ DỰ ÁN PHẦN MỀM

Nội dung chương 11 giới thiệu tổng quan về quản lý dự án phần mềm, trình bày các yêu cầu cần thiết trong quản lý rủi ro, quản lý nhân sự và quản lý kế hoạch. Sau khi học xong, người học có thể hiểu và giải thích được vai trò, đặc điểm của công việc quản lý dự án phần mềm, quản lý rủi ro, quản lý nhân sự và có thể sử dụng hai phương pháp PERT, biểu đồ Gantt trong việc lập kế hoạch các dự án đơn giản.

11.1 Giới thiệu

Dự án phần mềm (Software project) là một tập hợp các công việc hoặc hoạt động liên quan với nhau cùng hướng đến các mục tiêu chung nhằm tạo ra sản phẩm phần mềm trong phạm vi nguồn lực và tài nguyên hữu hạn.[]

Quản lý dự án phần mềm là một thuật ngữ chỉ công việc quản trị và điều hành tất cả các hoạt động trong một dự án phần mềm, ưu tiên tập trung vào kế hoạch, lịch biểu, nguồn lực, việc thực thi, giám sát, kiểm soát và phân phối các sản phẩm phần mềm []. Vì thế, quản lý dự án phần mềm là một công việc tất yếu trong lĩnh vực công nghệ phần mềm. Muốn sản xuất ra một sản phẩm có chất lượng cao thì việc quản lý dự án của sản phẩm đó phải có chất lượng trước hết. Việc quản lý dự án tốt có thể không đảm bảo dự án thành công, tuy nhiên, khi việc quản lý tồi thì dễ dàng dẫn đến sự thất bại của dự án. Các ràng buộc về kinh phí, thời gian và nhân lực là những thử thách lớn nhất mà người quản lý dự án phải đối mặt và vượt qua chúng để cho ra đời các sản phẩm phần mềm có chất lượng.

Nhìn chung, việc quản lý dự án thường hướng đến các tiêu chí sau:

- Hoàn thành dự án đúng hạn
- Kiểm soát không phát sinh chi phí ngoài kế hoạch
- Sự hài lòng của khách hàng
- Sự thỏa mãn của đội phát triển phần mềm

Mục tiêu tổng quan của quản lý dự án trong nhiều lĩnh vực thường vẫn hướng đến thời gian, kinh phí và nhân lực, nhưng lại khác nhau về đối tượng sản phẩm. Đối với dự án phát triển phần mềm, đối tượng sản phẩm có hai đặc thù sau:

- Sản phẩm điện tử: sản phẩm chỉ được sử dụng trên máy tính, trong môi trường ảo. Do đó, hầu như các công việc phát triển phần mềm đều phải thông qua máy tính,

không tương tác trực tiếp. Những việc như tạo ra các thành phần, quan sát sản phẩm, kiểm tra chất lượng, hoạt động mua – bán, bàn giao cho khách hàng đều bị chi phối bởi đặc thù sản phẩm.

- Sự thay đổi công nghệ: so với các lĩnh vực kỹ thuật khác, công nghệ thông tin có tốc độ phát triển khá nhanh. Do đó, theo thời gian kinh nghiệm quản lý nhiều dự án phần mềm sẽ cần sự linh hoạt cao đối với các yếu tố kỹ thuật.

Làm thế nào để quản lý tốt một dự án phần mềm chắc chắn sẽ không đơn giản, nhưng trước hết các loại công việc sau đây cần phải được làm tốt,

- Quản lý rủi ro
- Quản lý nhân lực
- Lập kế hoạch dự án
- Lập báo cáo.

11.2 Quản lý rủi ro

Rủi ro là những việc gây ảnh hưởng xấu đến dự án và có thể gây ra những hậu quả nghiêm trọng. Vì vậy, quản lý rủi ro là một công việc quan trọng không thể thiếu trong quản lý dự án. Để nhận thức, ý thức được rủi ro, thì chúng cần được dự đoán, và để kiểm soát khả năng ảnh hưởng thì các phương án phòng và khắc phục cần phải được chuẩn bị. Rủi ro có thể phân loại thành ba nhóm:

- Rủi ro dự án: những rủi ro xuất phát từ hoặc tổn hại đến nguồn lực của dự án, như thời gian, chi phí và nhân sự. Chẳng hạn như, việc mất đi những nhân lực giỏi có thể gây ra chậm tiến độ và tăng chi phí cho việc tìm lực lượng thay thế.

- Rủi ro sản phẩm: những rủi ro xuất phát từ hoặc tổn hại đến các thành phần hoặc toàn bộ sản phẩm phần mềm. Ví dụ như, một thành phần được gia công từ bên ngoài đội phát triển, có chất lượng không đáp ứng kỳ vọng khi tích hợp. Điều này có thể gây giảm sút hiệu quả của toàn hệ thống.

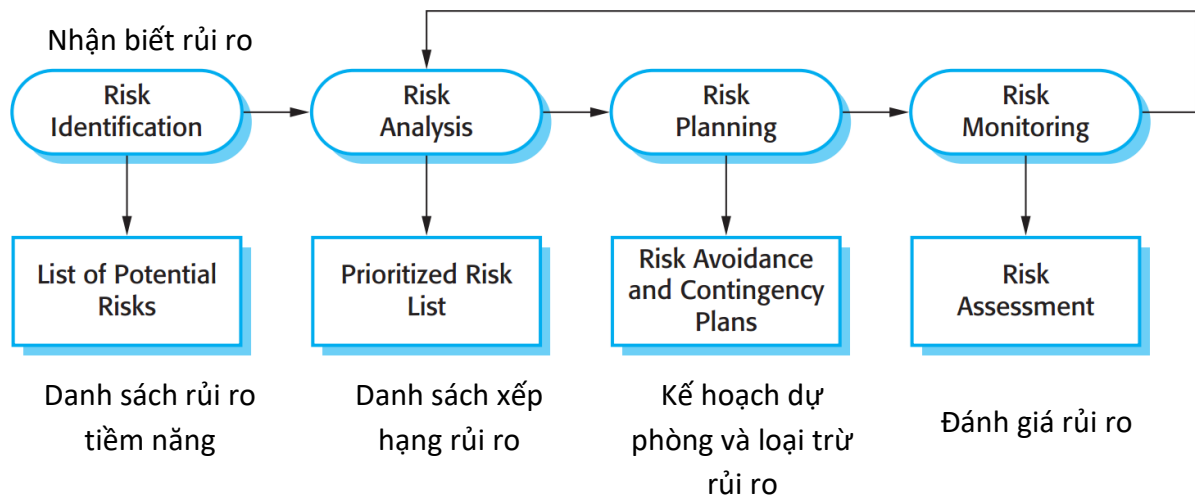
- Rủi ro tổ chức: những rủi ro xuất phát từ hoặc tổn hại đến các hoạt động, sứ mệnh của công ty, tổ chức đầu tư hoặc phát triển phần mềm. Chẳng hạn như, dự án đang thực hiện bị hủy hợp đồng do một biến cố của phía đầu tư. Mặc dù, phía phát triển không bị tổn hại về tài chính (do được bồi thường hợp đồng), nhưng về uy tín và thương hiệu có thể bị ảnh hưởng.

Những rủi ro thường xuất hiện trong nhiều dự án bao gồm: (Bảng 11.1)

Bảng 11.1 Những rủi ro phổ biến

Vấn đề	Nội dung
Nhân lực tổn hao	Sự mất mát hoặc thay đổi nhân lực
Thay đổi chính sách	Tổ chức nội bộ và chiến lược hoạt động thay đổi
Thiết bị không sẵn dùng	Sự chậm trễ trong các gói thầu thiết bị
Các yêu cầu thay đổi	Nếu làm không tốt, việc thay đổi sẽ không bao giờ chấm dứt
Đặc tả thiết sót	Xuất hiện nhiều hơn các vòng lặp xác minh
Đánh giá không đúng quy mô sản phẩm	Các nguồn lực đầu tư không tương xứng
Công cụ CASE không như kỳ vọng	Nhiều nghiệp vụ phát sinh phải xử lý thủ công
Công nghệ thay đổi	Khi sản phẩm hoàn thành, công nghệ đã thay đổi.
Sản phẩm cạnh tranh	Sản phẩm của một thương hiệu khác đã đi trước một bước.

Quy trình quản lý rủi ro có thể được tóm tắt như sau (Hình 11.1):



Hình 11.1 Tóm tắt quy trình quản lý rủi ro (Sommerville, I. 2011)

Quy trình này bắt đầu bằng việc nhận biết và xác định tất cả những vấn đề có khả năng gây ra rủi ro. Kết quả của bước này danh sách các rủi ro tiềm năng có thể trở thành rủi ro thực sự hoặc không. Do đó, bước tiếp theo nhằm phân tích, phân loại và đánh giá khả năng rủi ro xảy ra và mức độ tác động, ảnh hưởng của rủi ro đối với dự án. Kết quả của bước này là danh sách xếp hạng các rủi ro theo mức độ ưu tiên cần phải ứng phó, giải quyết. Việc này sẽ giúp kế hoạch sử dụng nguồn lực của dự án tập trung hiệu quả để ứng phó các rủi ro. Kế hoạch dự phòng và loại trừ rủi ro cần được thiết lập nhằm làm cơ sở cho việc giám sát và đánh giá các rủi ro. Kết quả đánh giá rủi ro sẽ tiếp tục hỗ trợ cho công đoạn phân tích rủi ro trong các giai đoạn tiếp theo của dự án.

11.3 Quản lý nhân lực

Ở bất kỳ dự án và lĩnh vực nào, con người luôn là yếu tố có vai trò quyết định lớn nhất đối với sự thành công của công việc. Vừa là yếu tố vận hành sự hoạt động của công ty, tổ chức vừa là yếu tố mục tiêu mà các công ty, tổ chức hướng đến sự phục vụ. Sự thịnh vượng và phát triển của một thương hiệu thì song song đó là sự hài lòng và thỏa mãn trong nhân viên của họ.

Trong quản lý dự án phần mềm cũng tương tự, cũng luôn cần những nhân sự giỏi, có tài năng. Để nhận được sự hài lòng và thỏa mãn của họ hướng đến những đóng góp to lớn cho công ty, tổ chức, thì vai trò quan trọng nhất thuộc về người quản lý dự án. Để quản lý hiệu quả nhân lực, có rất nhiều việc phải làm, tuy nhiên có ba yếu tố cơ bản sau đây mà người quản lý dự án cần quan tâm:

- Sự công bằng: tương tự như ở ngoài xã hội, sự công bằng luôn là tiêu chí quan trọng trong mối quan hệ, trong hoạt động hành xử, giao tiếp giữa người với người. Bất kỳ sự đóng góp nào của nhân viên, dù to lớn hay nhỏ bé đều phải được ghi nhận một cách công bằng.

- Sự tôn trọng: không ai trên hành tinh này giỏi tất cả mọi việc. Việc quản lý nhân sự hiệu quả như thế nào là phụ thuộc nhiều vào sự tôn trọng cả những điểm mạnh và cả những điểm yếu của nhân viên. Nhiều người thường nói rằng không sợ nhân viên tồi, chỉ lo người quản lý kém.

- Sự chân thật: có một câu châm ngôn như sau, khi sự thật trở nên khó khăn, thì nó là phương án tốt nhất. Tất cả những gì đang diễn ra trong tập thể, dù tốt hay xấu, mọi người đều có quyền biết về nó. Chỉ như vậy, tập thể mới có những hành xử đúng với thực tế. Chẳng hạn như, khả năng lập trình của đội phát triển chưa đáp ứng được yêu cầu mới về công nghệ, người quản lý cần mạnh dạn chỉ ra và có phương án cải thiện.

Các nội dung then chốt này, nếu được đảm bảo sẽ góp phần to lớn hướng đến một môi trường làm việc chuyên nghiệp và khoa học. Bên cạnh đó, nhằm tạo ra những động lực mạnh mẽ nhằm thúc đẩy năng suất làm việc, thì hai nhóm nhu cầu chủ yếu cần phải đáp ứng, bao gồm nhu cầu đảm bảo cuộc sống và nhu cầu khẳng định bản thân.

Đảm bảo cuộc sống là nhu cầu vô cùng nhân văn trong xã hội, nó bao gồm nhu cầu việc làm và thu nhập. Một lập trình viên không thể dành hết thời gian cho dự án, và rồi họ phải làm ngoài giờ để trang trải cuộc sống, thay vì có thời gian rảnh để tận hưởng. Tất nhiên, người quản lý dự án không phải là người quyết định mức chi trả cho từng vị trí, bởi còn các yếu tố khác như chính sách của công ty chẳng hạn, nhưng là người quyết định mức chi trả hợp lý nhất cho nhân viên theo chính sách của công ty.

Khẳng định bản thân là nhóm nhu cầu thứ hai cần được quan tâm. Vị trí, vai trò, sự đóng góp...của bất kỳ thành viên nào trong dự án đều phải được công nhận chính thức và công khai. Nhân viên luôn mong muốn sự hiện diện của họ trong những chiến lược, sứ mệnh của công ty, tổ chức. Từ đó, nhu cầu về việc nâng cao trình độ, năng lực, bằng cấp... sẽ là kết quả tiếp theo nhằm hướng đến sự đóng góp và gắn bó bền lâu.

11.4 Nhóm làm việc

Việc tổ chức, bố trí và sử dụng con người cũng là một chức năng thường xuyên của người quản lý dự án. Cách thức tổ chức phổ biến nhất hiện nay trong các dự án phần mềm là nhóm làm việc. Toàn bộ công việc của dự án được phân bổ trong từng nhóm làm việc theo từng giai đoạn cụ thể. Mỗi nhóm có vị trí, vai trò nhất định và thường số lượng không quá 10 thành viên (nhóm nguyên tử). Vậy, yếu tố nào quyết định sự thành công của nhóm làm việc?

Sự đoàn kết được đánh giá là thử thách và là mục tiêu ảnh hưởng chính đến hiệu quả của nhóm làm việc. Một nhóm đoàn kết thật sự sẽ tự vận động một cách hiệu quả bằng cách tự tìm đến sự đồng thuận, tự học hỏi, chia sẻ và giúp đỡ lẫn nhau, cùng nhau làm việc và sẵn sàng vinh danh người có đóng góp lớn nhất.

Một số đề nghị sau đây có thể khuyến khích gia tăng sự đoàn kết trong từng nhóm:

- Quan tâm đến những mảng khác trong cuộc sống của từng thành viên, bên cạnh công việc
- Thường xuyên có những hoạt động, sự kiện tạo sự thân thiện, gần gũi ngoài giờ làm việc, như thể thao, dã ngoại, tổ chức sinh nhật, các hội games,...
- Chung tay giúp đỡ lẫn nhau trong những tình huống khó khăn
- Luôn nhớ thực hiện những gì đã cam kết, và phải có trách nhiệm khi nó không được thực hiện

Bên cạnh tinh thần đoàn kết, thì việc tổ chức nhóm một cách khoa học và đảm bảo các kênh thông tin trong và ngoài nhóm là các nội dung cần thiết giúp một nhóm làm việc hiệu quả.

11.5 Lập kế hoạch dự án

Lập kế hoạch dự án là một hoạt động chắc chắn không thể thiếu trong việc quản lý dự án. Kế hoạch dự án là tài liệu được lập giai đoạn đầu của dự án nhằm đề ra tất cả công việc sẽ và phải làm để hoàn thành dự án, những nhóm phụ trách công việc và thời gian tương ứng. Người quản lý dự án phải đảm bảo tài liệu kế hoạch được tất cả các bên tham gia dự án thông qua trước khi triển khai.

Kế hoạch của một dự án phần mềm thường bao gồm các nội dung sau:

- Giới thiệu: trình bày tổng quan các mục tiêu của dự án và toàn bộ các ràng buộc (tài chính, thời gian,...) tác động đến việc quản lý dự án.
- Cấu trúc dự án: trình bày cách thức các đội, nhóm trong dự án được tổ chức, thành viên tham gia và vai trò của từng người trong đội, nhóm.
- Rủi ro: trình bày các rủi ro tiềm năng trong suốt dự án, khả năng xuất hiện, mức độ nghiêm trọng, cách thức phòng ngừa và phương án giảm thiểu ảnh hưởng.
- Phần cứng và phần mềm cần thiết: các yêu cầu về phần cứng và phần mềm hỗ trợ phục vụ dự án, chi phí và thời gian ước lượng đầu tư nếu cần thiết.
- Tổ chức công việc: toàn bộ công việc của dự án được phân bổ thành các hoạt động cụ thể, mô tả kết quả hoặc sản phẩm của từng hoạt động và đặc biệt là chỉ ra các thời điểm then chốt nhằm đánh giá tiến độ của dự án.
- Lập lịch biểu: toàn bộ các hoạt động của dự án được đánh giá mức độ ưu tiên, sự liên kết, phân bổ nhân lực, thời gian
- Cơ chế báo cáo và giám sát: một dự án chuyên nghiệp luôn đầy ắp các tài liệu báo cáo ở hầu hết các giai đoạn, và việc giám sát là vô cùng cần thiết để đảm bảo tính khách quan trong đáp ứng yêu cầu công việc hoặc các tiêu chuẩn chất lượng có liên quan.

Việc lập kế hoạch dự án có thể sử dụng các phương pháp đơn giản và phổ biến bao gồm Cấu trúc phân chia công việc (Work Breakdown Structure), biểu đồ Gantt và biểu đồ PERT (Program Evaluation Review Technique). Phương pháp sử dụng biểu đồ PERT được trình bày cụ thể phần tiếp theo.

11.6 PERT

PERT là một kỹ thuật sử dụng biểu đồ mạng kèm theo việc tính toán các khoản thời gian của từng hoạt động nhằm hướng đến việc hoàn thành tất cả các hoạt động đó trong sự trật tự và sự lệ thuộc đã được định trước.

Tại sao sử dụng PERT?

- Giúp người quản lý dự án lập và quản lý thời gian biểu
- Giúp người quản lý xác định các công việc then chốt (những công việc không thể chậm trễ hoặc trì hoãn)

- Giúp giám sát sự phụ thuộc lẫn nhau giữa các công việc để chủ động phân bổ nguồn lực, đề ra các phương án thời gian linh hoạt

Các bước để lập một biểu đồ PERT như sau:

- Xác định các công việc cụ thể và thời gian thực hiện từng công việc độc lập
- Xác định trình tự các công việc và mối quan hệ giữa chúng
- Lập biểu đồ
- Tính các khoản thời gian cần thiết đối với từng công việc
- Xác định chuỗi công việc then chốt

Có bốn khái niệm trong PERT về thời gian cần thiết để hoàn thành một công việc:

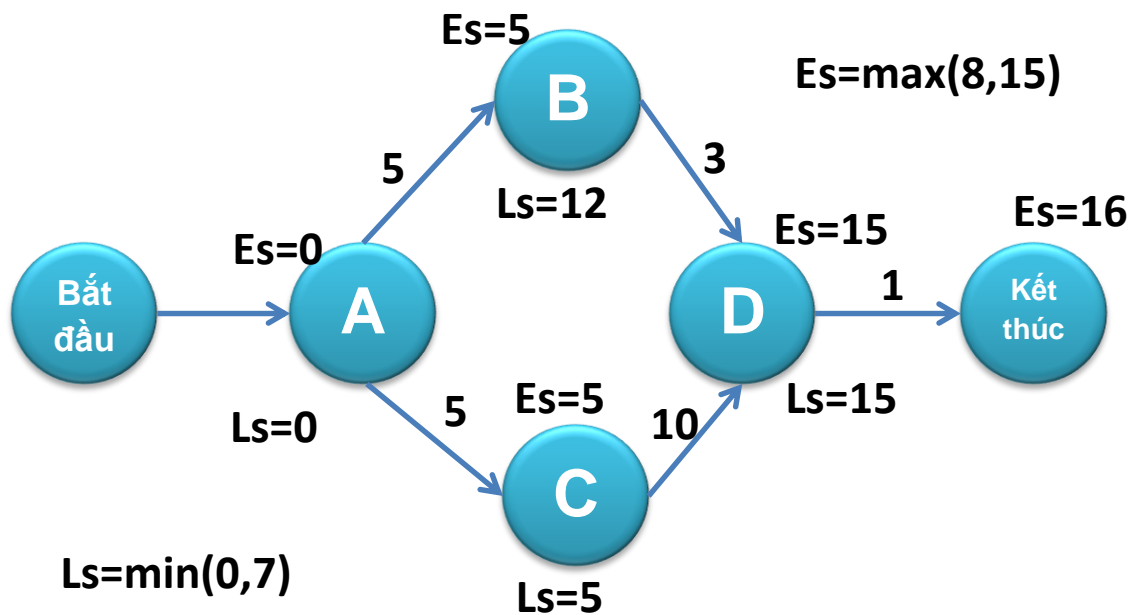
- Optimistic time: thời gian ít nhất để hoàn thành một công việc
- Pessimistic time: thời gian nhiều nhất một công việc có thể hoàn thành
- Most Likely time: thời gian hoàn thành một công việc khi hoàn toàn thuận lợi
- Expected time: thời gian hoàn thành một công việc khi gặp khó khăn

Bên cạnh đó, bốn thông số thể hiện trên biểu đồ bao gồm:

- Thời gian sớm nhất bắt đầu công việc (ES) = Max (các tổng thời gian hoàn thành các công việc trước đó)
- Thời gian sớm nhất hoàn thành công việc (EF) = ES + thời gian thực hiện công việc đó
- Thời gian trễ nhất bắt đầu công việc (LS) = Min (các thời gian trễ nhất để bắt đầu các công việc sau đó) – thời gian thực hiện công việc
- Thời gian trễ nhất hoàn thành công việc (LF) = (LS + thời gian thực hiện công việc) hoặc LF = LS công việc sau đó.
- Xét trường hợp sau:

Mã công việc	Tên công việc	Thời gian thực hiện (tuần)	Công việc trước đó
A	Khởi đầu	5	
B	Phân tích	3	A
C	Cài đặt	10	A
D	Chuyển giao	1	B, C

Áp dụng PERT, kết quả như sau (Hình 11.2):



Hình 11.2 Một trường hợp áp dụng PERT

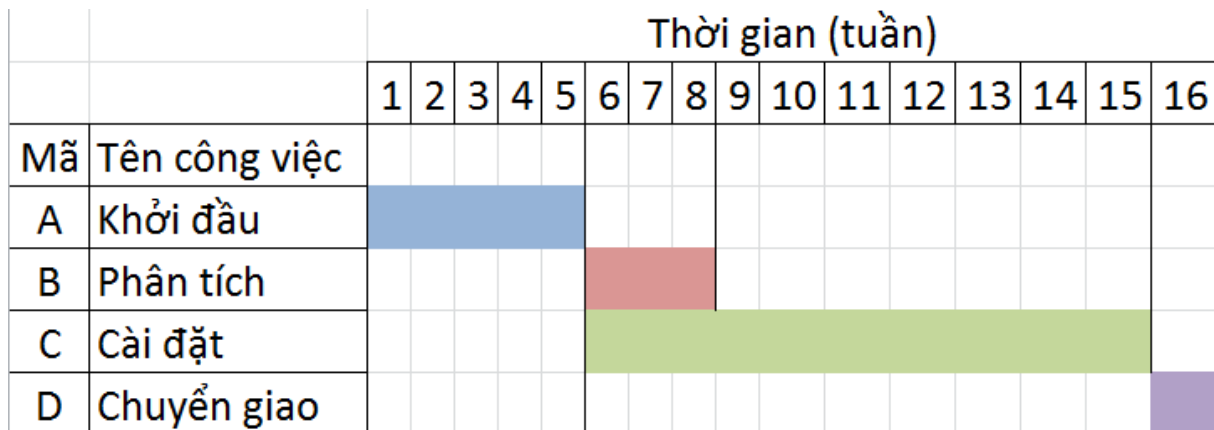
Căn cứ vào kết quả trên, các công việc then chốt bao gồm A, C, D vì chúng không có thời gian linh hoạt, thời điểm sớm nhất và trễ nhất để bắt đầu thực hiện đều tại một thời điểm. Trong khi đó, B có thể linh hoạt do có thể bắt đầu trong khoảng từ tuần 5 đến 12. Tổng thời gian dự án là 16 tuần.

11.7 Gantt

Biểu đồ Gantt là một phương pháp trực quan thể hiện tiến độ của một dự án bằng cách sử dụng các thanh ngang biểu diễn lượng thời gian cho từng công việc. Đây là phương pháp dễ thực hiện và phù hợp các dự án đơn giản với số lượng ít mối quan hệ phụ thuộc giữa các công việc. Các bước cơ bản xây dựng một biểu đồ Gantt như sau:

- Xác định toàn bộ công việc cần thiết để hoàn thành dự án
- Phân tích, xác định sự phụ thuộc và mối quan hệ giữa các công việc
- Ước tính lượng thời gian thực hiện từng công việc và các thời điểm bắt đầu, kết thúc dự án
- Lập biểu đồ Gantt.

Xét một dự án với bảng tóm tắt kế hoạch đề cập trong mục 11.6, Hình 11.3 là biểu đồ Gantt tương ứng.



Hình 11.3 Một trường hợp áp dụng biểu đồ Gantt

Theo biểu đồ Gantt, các công việc then chốt bắt buộc phải hoàn thành đúng thời gian dự kiến bao gồm A, C và D. Chỉ riêng công việc B được linh hoạt thời gian hoàn thành từ tuần 8 đến tuần 15.

Bài tập

1. Trình bày vai trò của việc quản lý dự án trong quy trình phát triển phần mềm.
2. Việc quản lý rủi ro giữ vai trò như thế nào đối với chất lượng của dự án và sản phẩm phần mềm?
3. Tại sao quản lý nhân sự trong các dự án luôn là một công việc nhiều khó khăn?
4. Trình bày vai trò của kế hoạch dự án trong quản lý dự án.
5. Hãy tìm hiểu khung mẫu Scrum (Scrum framework) trong lĩnh vực phát triển phần mềm, trình bày chức năng và cách thức áp dụng.
6. Kỹ thuật PERT được sử dụng với mục đích gì trong quản lý dự án?
7. Sử dụng PERT với dữ liệu như sau:

STT	Hoạt động	Thời gian thực hiện (tuần)	Hoạt động trước đó
1	A	3	-
2	B	5	A
3	C	3	A
4	D	10	B
5	E	7	B
6	F	4	C
7	G	5	E,F
8	H	2	D,G

a. Xác định tổng thời gian cần thiết để hoàn thành dự án

b. Xác định các công việc then chốt và các thời điểm có thể bắt đầu các hoạt động một cách linh hoạt.

8. Sử dụng biểu đồ Gantt quản lý dự án sau:

Mã công việc	Tên công việc	Thời gian thực hiện (tuần)	Công việc trước đó
A	Thu thập yêu cầu	10	
B	Đặc tả và thiết kế	7	A
C	Đánh giá khả thi	5	B
D	Phát triển cơ sở dữ liệu	5	B
E	Cài đặt	22	D
F	Tích hợp và kiểm thử	6	C, D
G	Lập tài liệu	4	B, C, D, E
H	Bàn giao	3	F, G

a. Xác định tổng thời gian cần thiết để hoàn thành dự án

b. Xác định các công việc then chốt và các thời điểm có thể bắt đầu các hoạt động một cách linh hoạt.

9. Hãy so sánh ưu và nhược điểm của hai phương pháp PERT và Gantt.

10. Hãy tìm hiểu các chức năng khác (chưa được đề cập) của phương pháp biểu đồ Gantt.

Chủ đề thảo luận

1. Đề xuất các phương án (khác nội dung bài học) giúp tạo ra một nhóm làm việc hiệu quả trong các dự án phần mềm.

2. Việc lập kế hoạch dự án và việc lựa chọn mô hình phát triển phần mềm có mối liên hệ với nhau không? Giải thích câu trả lời.

TÀI LIỆU THAM KHẢO

- [1] Sommerville, I., (2011). Software engineering 9th Edition. ISBN-10, 137035152.
- [2] Stephens, R., (2015). Beginning software engineering. John Wiley & Sons.
- [3] Marsic, I., (2012). Software engineering. Rutgers.
- [4] IBM (2011). Rational Software White Paper TP026B.
- [5] Tripp, Steven D. & Bichelmeyer, Barbara, (1990). Rapid Prototyping: An Alternative Instructional Design Strategy.
- [6] Bob Hughes & Mike Cotterell, (1999). Software Project Management. (Second Edition). University of Brighton. The McGraw-Hill.
- [7] Lovely Professional University, (2012). Software Project Management. Excel Books Private Limited.
- [8] Grady Booch, James Rumbaugh & Ivar Jacobson, (2005). The Unified Modeling Language User Guide (Second Edition). Addison-Wesley Professional.
- [9] Huỳnh Xuân Hiệp & Phan Phương Lan. (2011). Giáo trình Nhập môn Công nghệ phần mềm. NXB Đại học Cần Thơ.
- [10] Nguyễn Văn Vy & Nguyễn Việt Hà. (2008). Giáo trình Kỹ nghệ phần mềm. NXB Đại học Quốc gia Hà Nội.
- [11] https://www.tutorialspoint.com/software_engineering/. Truy cập 4/2019
- [12] <https://www.wikipedia.org/>. Truy cập 4/2019.

PHỤ LỤC 1

MỘT SỐ TIÊU CHUẨN QUỐC TẾ

1.1 ISO

ISO là tên viết tắt của Tổ chức Quốc tế về tiêu chuẩn hoá (*International Organization for Standardization*), được thành lập vào năm 1946 và chính thức hoạt động vào ngày 23/2/1947, nhằm mục đích xây dựng các tiêu chuẩn về sản xuất, thương mại và thông tin.

Bộ tiêu chuẩn chất lượng ISO 9001-3 của tổ chức ISO, quy định về "Quy trình đảm bảo chất lượng" trong các tổ chức phát triển phần mềm. Chứng chỉ ISO 9001 xác nhận các tổ chức, đơn vị có quy trình đảm bảo chất lượng hợp chuẩn. Bên cạnh đó, một mô hình khác là CMM (Capability Maturity Model) cũng đang rất được quan tâm tại Việt Nam. Công ty nhận được chứng chỉ CMM nghĩa là công ty đó đã đạt được mức độ tương ứng với các cấp độ CMM của chứng chỉ. Một doanh nghiệp phát triển phần mềm, nếu có chứng chỉ CMM hoặc ISO 9001 đều có khả năng sản xuất ra các phần mềm tốt hơn hẳn các công ty chưa có chứng chỉ. Tuy nhiên, chúng ta cần lưu ý đây chỉ là "*khả năng*" chứ không phải là "*chắc chắn*". Vẫn có doanh nghiệp có quy trình tốt nhưng sản xuất ra sản phẩm chất lượng không cao. Điều này chứng tỏ cách tiếp cận theo chất lượng quy trình chưa phải là cách tiếp cận toàn diện mà chỉ giải quyết vấn đề ở mức căn bản.

Những năm cuối thế kỷ 20, tổ chức ISO đã tập trung rất nhiều vào các tiêu chuẩn chất lượng cho phần mềm. Cách tiếp cận về chất lượng của ISO đã thực sự tiến thêm một bậc, toàn diện hơn, phù hợp hơn. Kết quả của sự tập trung này là một loạt các bộ tiêu chuẩn đã ra đời, nhằm hướng tới đánh giá chất lượng toàn diện trong suốt vòng đời của sản phẩm phần mềm, từ khi phôi thai cho tới lúc lạc hậu cần thay thế. Theo cách tiếp cận của ISO, chất lượng toàn diện của phần mềm cần phải được quan tâm từ chất lượng quy trình, tới chất lượng phần mềm nội bộ (chất lượng trong), chất lượng phần mềm đối chiếu với yêu cầu của người dùng (chất lượng ngoài) và chất lượng phần mềm trong sử dụng (chất lượng sử dụng).

1.2 CMM/CMMI

CMMI viết tắt cho Capability Maturity Model Integration - Mô hình trưởng thành năng lực tích hợp - và là khuôn khổ cho cải tiến qui trình phần mềm. Nó dựa trên khái niệm về các thực hành tốt nhất về kỹ nghệ phần mềm và giải thích kỉ luật mà các công ty có thể dùng để cải tiến các qui trình của họ.

CMMI là phiên bản kế tiếp của CMM. Cả CMM và CMMI đều được Viện kỹ nghệ phần mềm Mỹ SEI tại trường Đại học Carnegie Mellon ở Pittsburgh, PA phát triển. CMM đã có mặt từ cuối những năm 80 và một thập kỷ sau nó bị CMMI thay

thể. Năm 2000 CMMI phiên bản 1.02 được đưa ra thị trường. Phiên bản mới nhất hiện nay CMMI 1.2 được trình làng vào tháng 8 năm 2006.

Khác biệt giữa ISO 9001:2000 và CMM/CMMi?

- ISO 9001 là một tiêu chuẩn quốc tế về quản lý, các điều khoản gọi là “yêu cầu” quy định những điểm cần phải làm (what to do), không chỉ ra việc đó nên làm như thế nào (how to do).
- CMM/CMMi là một mô hình, cung cấp các hướng dẫn và kinh nghiệm thực tế dùng để phát triển, cải tiến và đánh giá năng lực của quy trình.
- CMMi không phải là một tiêu chuẩn, tùy vào từng tổ chức, cách thực hiện khác nhau rất nhiều.
- Về nguyên tắc, ISO bao gồm (ở mức cao) hầu hết các quy trình chủ chốt của CMM/CMMi, tuy nhiên ISO được dùng cho hầu hết mọi ngành nghề, do vậy không cụ thể và gần gũi với công việc đặc thù có liên quan đến phần mềm như CMM/CMMi. ISO không cung cấp các ví dụ và kinh nghiệm cụ thể như CMM/CMMi

Các cấp độ của CMM/CMMI

CMM bao gồm 5 cấp độ và 18 KPAs (Vùng quy trình quan trọng - Key Process Area).

Nói cách khác mỗi một cấp độ đều tuân theo một chuẩn ở mức độ cao hơn. Muốn đạt được chuẩn cao hơn thì các chuẩn của các cấp độ trước phải thỏa mãn. Mỗi cấp độ đều có đặc điểm chú ý quan trọng của nó cần các doanh nghiệp phải đáp ứng được.

Cấp độ 1 thì không có KPAs nào cả

Cấp độ 2: có 6 KPAs

Cấp độ 3: có 7 KPAs

Cấp độ 4: có 2 KPAs

Cấp độ 5: có 3 KPAs

18 KPAs của CMM được đều có 5 thuộc tính (chức năng) chung trong đó có các qui định về key practice là những hướng dẫn về các thủ tục (procedure), qui tắc (polities), và hoạt động (activities) của từng KPA.

Mô hình này xác định năm cấp độ của CMM đối với một công ty: Khởi đầu (lộn xộn, không theo chuẩn) - Lập (quản lý dự án, tuân thủ quy trình) - Xác lập (thể chế hóa) - Kiểm soát (định lượng) - Tối ưu (cải tiến quy trình).

1.3 Mô hình chất lượng ISO-9126

ISO-9126 mô tả một mô hình chất lượng sản phẩm phần mềm gồm hai phần:

- Chất lượng trong và chất lượng ngoài
- Chất lượng sử dụng

Phần thứ nhất của mô hình xác định 6 tiêu chí của chất lượng trong, 6 tiêu chí chất lượng ngoài; các tiêu chí này sau đó lại được chia nhỏ thành nhiều tiêu chí con. Những tiêu chí này được bộc lộ ra ngoài khi phần mềm được coi như là một phần của hệ thống máy tính và là kết quả của các thuộc tính phần mềm bên trong.

Phần thứ hai của mô hình mô tả 4 tiêu chí chất lượng sử dụng. Chất lượng sử dụng là hệ quả của 6 tiêu chí chất lượng sản phẩm phần mềm đối với người dùng.

Các tiêu chí sản phẩm phần mềm này có thể áp dụng cho tất cả các loại phần mềm. Những tiêu chí sản phẩm phần mềm tạo ra sự nhất quán đối với chất lượng sản phẩm phần mềm, đồng thời cung cấp một khung cho việc xác định các yêu cầu đối với chất lượng phần mềm.

Trong phần này, chất lượng sản phẩm phần mềm được xác định và đánh giá theo nhiều hướng, gắn với kết quả thu được, các yêu cầu, sự phát triển, sử dụng, đánh giá, hỗ trợ, tính ổn định, đảm bảo chất lượng và kiểm định của phần mềm. Nó có thể được sử dụng bởi nhà phát triển, tổ chức sử dụng, nhân viên đảm bảo chất lượng phần mềm hay người đánh giá độc lập. Đồng thời nó đặc biệt thích hợp cho việc xác định và đánh giá chất lượng sản phẩm phần mềm. Ví dụ, mô hình chất lượng này có thể được dùng để:

- Kiểm tra tính đáp ứng đối với những yêu cầu đã đặt ra.
- Xác định các yêu cầu phần mềm.
- Xác định các đối tượng thiết kế phần mềm.
- Xác định các đối tượng kiểm thử phần mềm.
- Xác định các tiêu chuẩn đảm bảo chất lượng.
- Xác định các tiêu chuẩn chấp nhận cho một sản phẩm phần mềm hoàn chỉnh.

PHỤ LỤC 2
KHUNG ĐẶC TẢ YÊU CẦU PHẦN MỀM
(IEEE)

Đặc tả yêu cầu phần mềm

<Tên dự án>

Biên soạn <Tên tác giả>

<Ngày thông qua>

Mục lục

Mục lục	Error! Bookmark not defined.
Danh mục phiên bản	98
1. Introduction.....	Error! Bookmark not defined.
1.1 Purpose	Error! Bookmark not defined.
1.2 Document Conventions	Error! Bookmark not defined.
1.3 Intended Audience and Reading Suggestions	Error! Bookmark not defined.
1.4 Product Scope	Error! Bookmark not defined.
1.5 References	99
2. Overall Description	99
2.1 Product Perspective	99
2.2 Product Functions.....	99
2.3 User Classes and Characteristics	100
2.4 Operating Environment	100
2.5 Design and Implementation Constraints	100
2.6 User Documentation.....	100
2.7 Assumptions and Dependencies	100
3. External Interface Requirements	100
3.1 User Interfaces.....	100
3.2 Hardware Interfaces.....	100
3.3 Software Interfaces	101
3.4 Communications Interfaces	101
4. System Features.....	101
4.1 System Feature 1	101
4.2 System Feature 2 (and so on)	102
5. Other Nonfunctional Requirements.....	102
5.1 Performance Requirements	102
5.2 Safety Requirements.....	102
5.3 Security Requirements	102
5.4 Software Quality Attributes.....	102
5.5 Business Rules.....	102
6. Other Requirements	102
Appendix A: Glossary	102
Appendix B: Analysis Models.....	102
Appendix C: To Be Determined List	103

Danh mục phiên bản (Revision History)

Tên tài liệu (Name)	Ngày thông qua (Date)	Lý do cập nhật (Reason For Changes)	Phiên bản (Version)

Giới thiệu

Mục đích (Purpose)

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

Cơ sở tài liệu (document convention)

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

Đối tượng độc giả (Target reader)

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

Phạm vi sản phẩm (product scope)

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

Thông tin tham khảo (References)

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Mô tả tổng quan (Overall Description)

Các khái niệm cho sản phẩm (Product Perspective)

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

Các chức năng của sản phẩm (Product Functions)

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

Đặc trưng và Lớp người sử dụng (User Classes and Characteristics)

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

Môi trường vận hành (Operating Environment)

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

Ràng buộc thiết kế và cài đặt (Design and Implementation Constraints)

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

Tài liệu cho người sử dụng (ser Documentation)

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

Các giả định và phụ thuộc (Assumptions and Dependencies)

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

Yêu cầu giao diện ngoại tuyến (External Interface Requirements)

Giao diện người sử dụng (User Interfaces)

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

Giao diện phần cứng (Hardware Interfaces)

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

Giao diện phần mềm (Software Interfaces)

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

Giao diện truyền thông (Communications Interfaces)

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

Chức năng hệ thống (System Features)

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

Tên chức năng 1 (System Feature 1)

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Diễn giải và Độ quan trọng (Description and Priority)

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Luồng công việc người dùng – hệ thống (Stimulus/Response Sequences)

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Yêu cầu chức năng (Functional Requirements)

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

Tên chức năng 2 (System Feature 2)...

Yêu cầu phi chức năng (Other Nonfunctional Requirements)

Yêu cầu vận hành, xử lý (Performance Requirements)

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

Yêu cầu về an toàn (Safety Requirements)

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

Yêu cầu về an ninh (Security Requirements)

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

Các thông số chất lượng phần mềm (Software Quality Attributes)

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

Quy tắc vận hành (Business Rules)

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

Yêu cầu khác (Other Requirements)

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Phụ lục A: Thuật ngữ/Từ khóa (Appendix A: Glossary)

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Phụ lục B: Mô hình phân tích (Appendix B: Analysis Models)

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Tài liệu đặc tả phần mềm <Tên dự án>

Phụ lục C: Danh mục kèm theo (Appendix C: To Be Determined List)

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

PHỤ LỤC 3

TÀI LIỆU ĐẶC TẢ YÊU CẦU

PHẦN MỀM BÁN HÀNG

(Học viện Kỹ thuật Quân sự, (2015). Tài liệu báo cáo môn Công nghệ phần mềm.)

I. GIỚI THIỆU

1. Mục tiêu

Mục đích của tài liệu này là để cung cấp mô tả chi tiết về các yêu cầu cho phần mềm hỗ trợ việc bán hàng. Nó sẽ minh họa mục đích cụ thể và thông tin chi tiết cho việc phát triển hệ thống. Tài liệu này sẽ nói lên đầy đủ về các ràng buộc của hệ thống, giao diện người dùng và tương tác với các ứng dụng bên ngoài(máy in). Nó được đề xuất cho khách hàng phê duyệt và là tài liệu tham khảo đầu vào cho các giai đoạn thiết kế, lập trình, kiểm thử trong quy trình sản xuất phần mềm này.

2. Phạm vi

Phần mềm hỗ trợ bán hàng được cài đặt trên hệ thống máy tính tại quầy thu ngân, giúp nhân viên thu ngân thanh toán hóa đơn của khách hàng nhanh chóng, chính xác và dễ dàng thực hiện việc in hóa đơn cho khách hàng. Đồng thời phục vụ cho người quản lý thống kê, kiểm soát doanh thu của nhà hàng. Là cơ sở giao tiếp của các thành viên, là căn cứ để thiết kế, kiểm thử.

3. Thuật ngữ và các từ viết tắt.

Thuật ngữ	Định nghĩa	Giải thích
CSDL	Cơ sở dữ liệu	Nơi lưu trữ thông tin và cho phép truy cập
DESC	Description	Mô tả
DEP	Dependency	Sự phụ thuộc
RAT	Rational	Quản hệ
MUST		Mức độ tối thiểu cần thiết để vượt qua các đợt kiểm thử

4. Tài liệu tham khảo

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

5. Mô tả tài liệu

Phần	Mô tả	Mục đích
1.Giới thiệu	Mô tả một cách khái quát nhất, phạm vi, mục đích tài liệu cần đạt được	Đưa ra các công việc cần thực hiện của tài liệu
2.Tổng quan về phần mềm	Khái quát phần mềm, đưa ra mô hình của phần mềm	Khái quát phần mềm, đối tượng sử dụng, mô hình ngữ cảnh.
3.Phân tích quy trình nghiệp vụ	Các chức năng của hệ thống	Phân tích qui trình nghiệp vụ một cách cụ thể nhất.
4.Đặc tả yêu cầu	Các yêu cầu về chức năng và hiệu năng của phần mềm	Trình bày các yêu cầu mà phần mềm cần phải thực thi

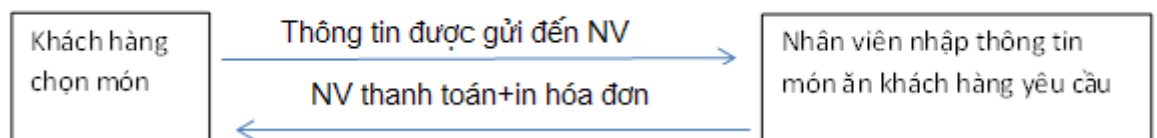
II. TỔNG QUAN VỀ PHẦN MỀM

1. Yêu cầu chung về phần mềm.

Hệ thống phần mềm bao gồm 2 phần: Quản lý việc thanh toán hóa đơn và Quản lý thông tin trong CSDL.

a. Quản lý việc thanh toán hóa đơn:

- Nhập thông tin món ăn của khách hàng yêu cầu(tên món, số lượng, các dịch vụ khác)
- Thanh toán và in hóa đơn cho khách hàng.



b. Quản lý thông tin trong CSDL

- Lưu trữ danh sách các món ăn trong nhà hàng→đối chiếu với món ăn khách hàng yêu cầu để thanh toán hóa đơn chi tiết và chính xác.
- Lưu trữ các hóa đơn đã được thanh toán theo thời gian→ phục vụ cho việc báo cáo thông tin sau này.

2. Chức năng của phần mềm

- a. Thanh toán hóa đơn cho khách hàng
- Khách hàng chọn món ăn trong menu món ăn của nhà hàng.

- Khi khách hàng có yêu cầu thanh toán → danh sách món ăn mà khách hàng chọn sẽ được gửi đến cho NV thu ngân. Tại đây, NV nhập mã món ăn tương ứng với mã món ăn trong danh sách đó.
- In hóa đơn và gửi lại cho khách hàng.
- b. Quản lý việc báo cáo doanh thu.
 - NV thu ngân sẽ thống kê doanh thu của nhà hàng theo thời gian yêu cầu của người quản lý.
 - In báo cáo và gửi lên người quản lý.

3. Đối tượng người dùng.

- Phần mềm được ứng dụng cho 2 đối tượng là: Khách hàng, NV thu ngân và người quản lý:
 - Khách hàng không trực tiếp sử dụng phần mềm, nhưng là người đưa ra các món ăn yêu cầu cho NV thu ngân.
 - NV thu ngân tiếp nhận danh sách món ăn khách hàng yêu cầu và thanh toán hóa đơn cho khách hàng.
 - Người quản lý:
 - Cập nhật món ăn
 - Kiểm soát doanh thu
 - Phân quyền người sử dụng phần mềm.

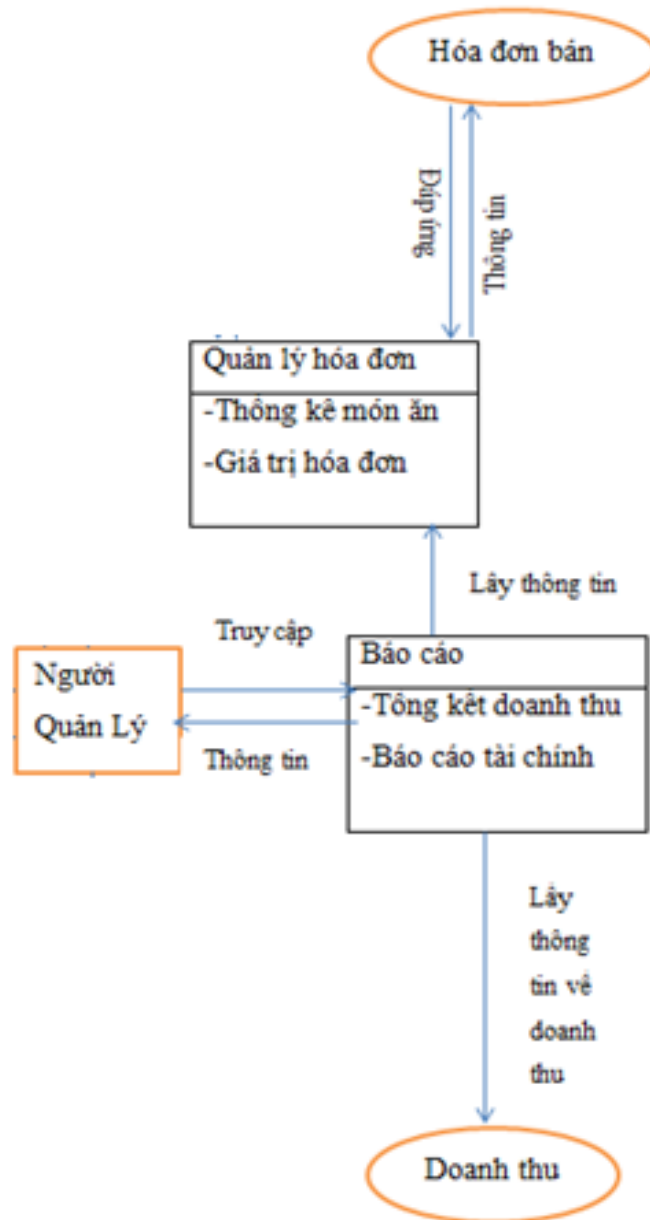
4. Các ràng buộc

- Hệ thống ràng buộc bởi yêu cầu của khách hàng:
 - Khi khách hàng có yêu cầu thanh toán hóa đơn → hệ thống mới cho phép nhập mã món ăn theo yêu cầu của khách hàng
- Ràng buộc về dữ liệu đầu vào so với dữ liệu trong CSDL: món ăn khách hàng chọn, được NV thu ngân nhập mã của các món ăn đó → tất cả thông tin về món ăn được hiển thị lên giao diện.
- Ràng buộc với ứng dụng bên ngoài: để gửi thông tin chi tiết của hóa đơn đến cho khách hàng → yêu cầu phải có máy in kết nối với hệ thống máy tính hiện tại → in hóa đơn cho khách hàng.

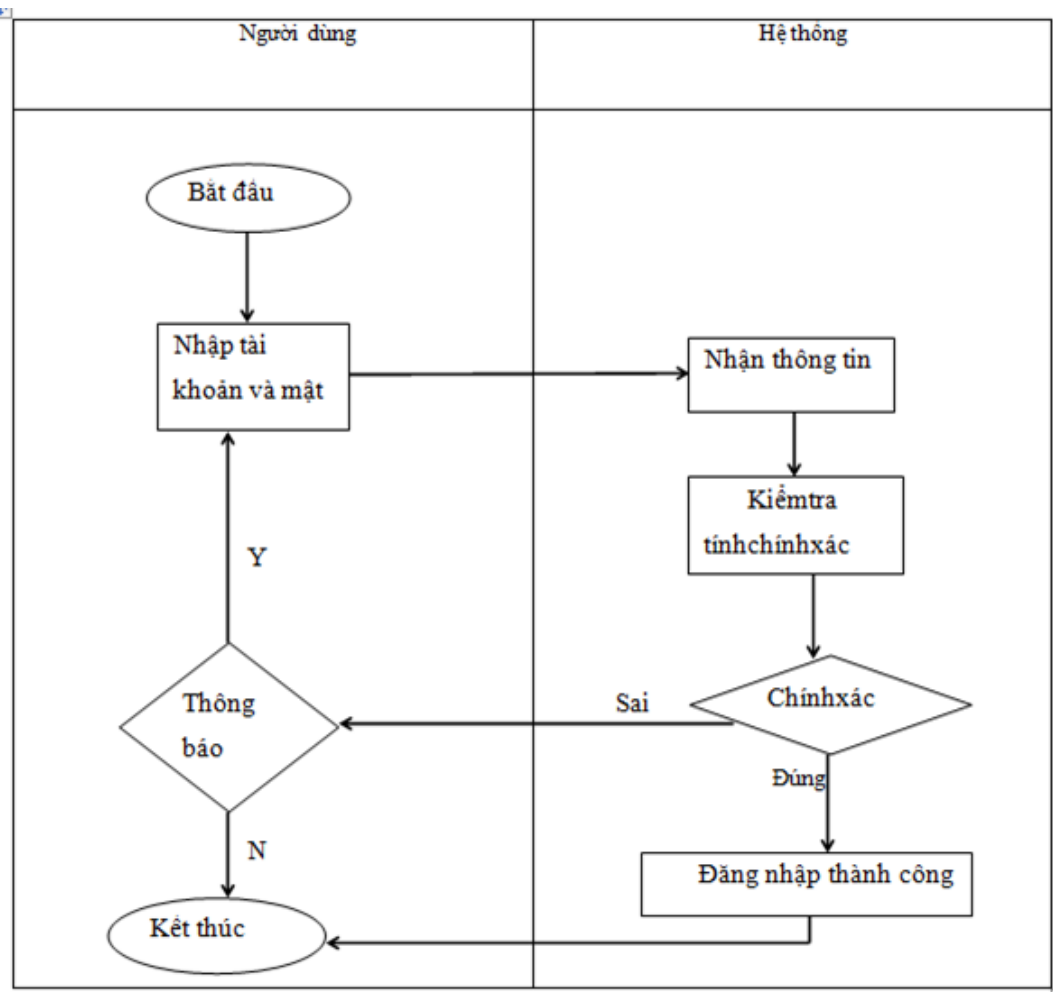
5. Giả định và phụ thuộc

Phần mềm chỉ được cài đặt trên hệ thống máy tính của nhà hàng tại quầy thu ngân, khi đó hệ thống phần mềm mới kết nối được CSDL của nhà hàng. Hoặc nếu không cài đặt trên hệ thống máy tính của nhà hàng mà cài đặt trên máy tính khác → buộc phải có CSDL của nhà hàng thì mới thực hiện đầy đủ được các thao tác nghiệp vụ.

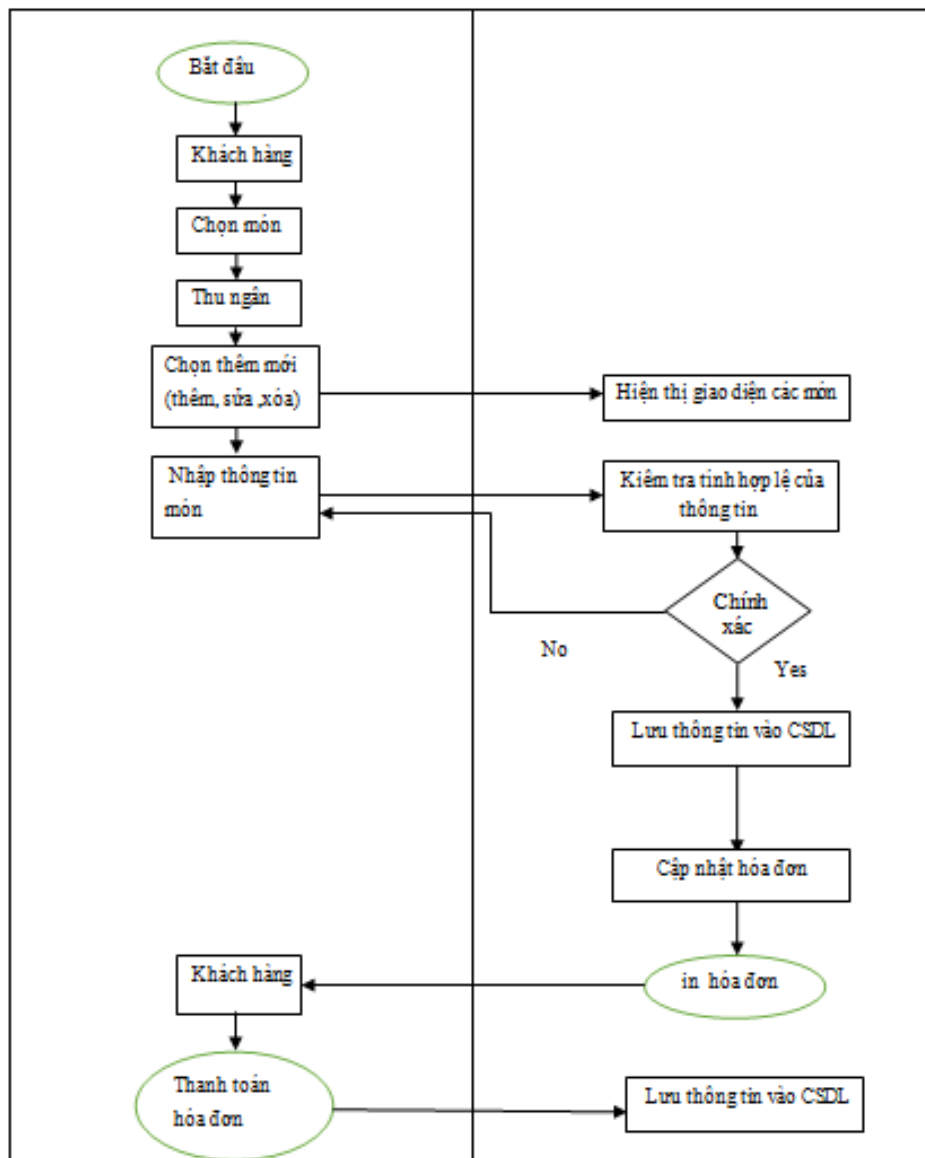
III. QUY TRÌNH NGHIỆP VỤ



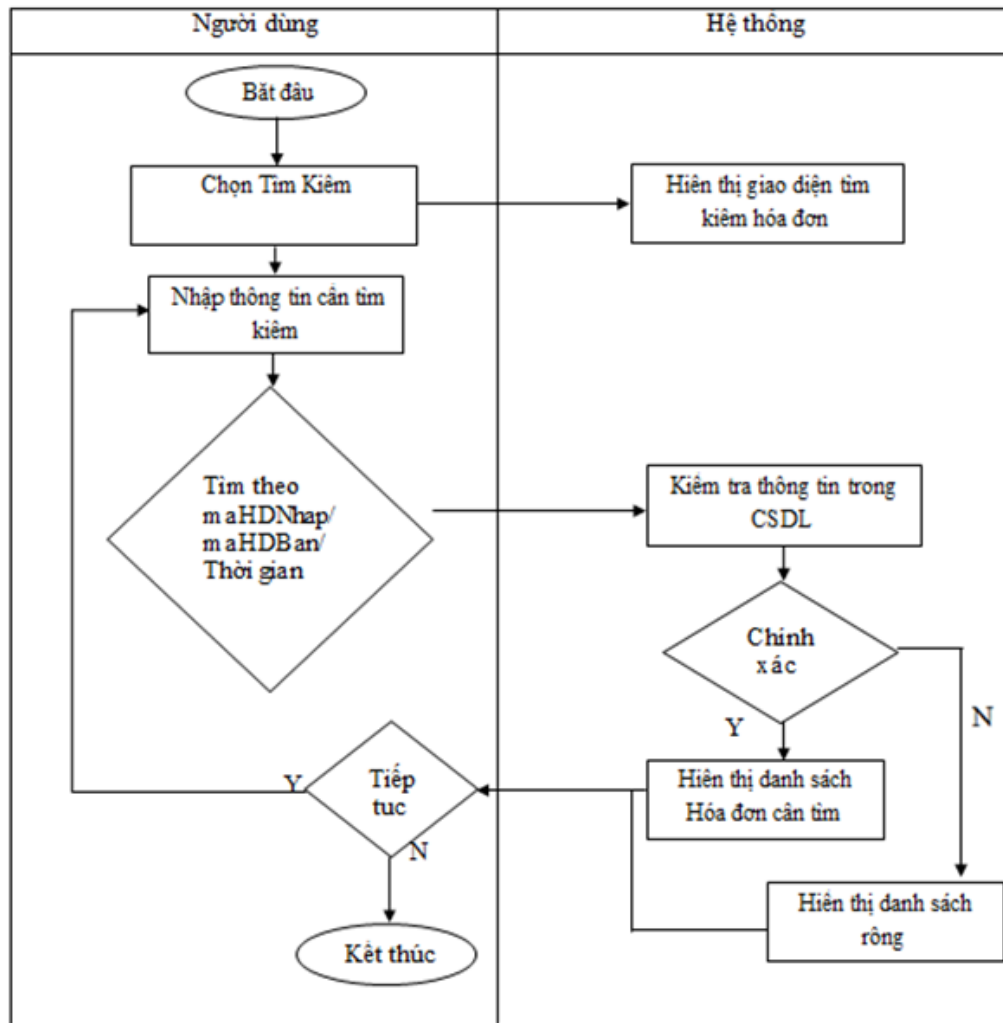
Hình 1. Sơ đồ tổng thể hệ thống



Hình 2. Mô hình nghiệp vụ đăng nhập vào hệ thống



Hình 3. Mô hình nghiệp vụ quản lý hóa đơn bán hàng của phần mềm



Hình 4. Mô hình nghiệp vụ thống kê báo cáo của phần mềm

IV. ĐẶC TẢ YÊU CẦU PHẦN MỀM

Phần này bao gồm tất cả các yêu cầu chức năng và chất lượng của hệ thống. Nó đưa mô tả chi tiết của hệ thống và tất cả các chức năng của nó.

1. Các yêu cầu giao diện bên ngoài

- Phần này Cung cấp một mô tả chi tiết của tất cả các yếu tố đầu vào và đầu ra từ hệ thống. Nó cũng mô tả giao diện phần cứng, phần mềm và cung cấp mẫu cơ bản của giao diện người dùng.
- a. Giao diện người dùng:
 - Khi nhân viên mở hệ thống phần mềm, sẽ hiển thị giao diện như hình 1, từ đây NV sẽ đăng nhập với tài khoản đã được cung cấp trước để đăng nhập vào hệ thống.

Hình 1. Giao diện đăng nhập hệ thống

- Sau khi đăng nhập thành công, sẽ hiển thị giao diện trang chủ, hình 2



Hình 2. Giao diện trang chủ.

- NV thu ngân chọn chức năng Quản lý hóa đơn bán hàng trên giao diện. Khi đó giao diện làm việc được hiển thị, hình 3.

Danh sách hóa đơn trong nhà hàng

Mã hóa đơn	Mã bán	Mã NV	Tên KH	Địa chỉ khách hàng	SĐT	Thời gian	Tổng tiền
HDB0001	B003	NV01	nguyen xuan chinh	thanh ha, hd	1234567	6/19/2015 2:49 ...	95000
HDB0002	B004	NV01	hsdthd	sdgd	24343	6/2/2015 11:36 ...	150000

Thông tin chi tiết cho 1 hóa đơn bán hàng

Mã HD: HDB0001 | Mã nhân viên: NV01 | Mã bán: B003

Tên khách hàng: nguyen xuan chinh | Địa chỉ KH: thanh ha, hd

SĐT: 1234567 | Thời gian đặt: 6/19/2015

Tổng tiền: 95000 vnd | Tình trạng hóa đơn: Đã Thanh Toán

Buttons: Thêm HD mới, Sửa, Xóa

Danh mục món ăn: Quay

Buttons: Thanh toán, Thoát, Tất cả món ăn

Chi tiết Hóa Đơn

Mã món	Số lượng	Đơn giá	Thành tiền
MA001	1	50000	45000
MA002	1	50000	50000

Thông tin hóa đơn

Mã món: | Tên món: | Số lượng: | Đơn giá: vnd | Chiết khấu: %

Buttons: Thêm món, Sửa, Xóa

Danh sách món ăn

Mã món	Tên món	Đơn giá(vnd)	Chiết khấu	Đơn vị tính
MA001	bo ham ngu vi huc...	50000	10	bat
MA003	Vật quay	100000	0	con

Hình 3. Giao diện chức năng #1

- NV nhập đầy đủ thông tin của một hóa đơn, và thực hiện thanh toán khi khách hàng yêu cầu.
- Khi người Quản lý yêu cầu thống kê báo cáo doanh thu nhà hàng, thì NV chọn chức năng báo cáo trên trang chủ sau đó sẽ hiển thị giao diện hình 4

	Mã hóa đơn	Mã bán ăn	Mã NV	Tên KH	Địa chỉ khách hàng	SĐT	Thời gian	Tổng tiền
▶	HDB0001	B003	NV01	nguyen xuan chinh	thanh ha, hd	1234567	6/19/2015 2:49 ...	95000
	HDB0002	B004	NV01	hsdthdf	sdgdg	24343	6/2/2015 11:36 ...	150000

Hình 4. Giao diện chức năng #2

- b. Giao diện phần cứng
 - Phần mềm quản lý và CSDL của nhà hàng đều được cài đặt trên một máy, được quản lý với hệ điều hành hiện tại của máy tính.
- c. Giao diện phần mềm
 - Giao diện thanh toán hóa đơn: cho phép đọc và chỉnh sửa thông tin về khách hàng và món ăn khách hàng yêu cầu.
 - Giao diện báo cáo, thống kê: chỉ cho phép đọc dữ liệu về doanh thu nhà hàng với thời gian chỉ định.

2. Yêu cầu chức năng.

a. Nhân viên.

ID: FR1

TITLE: Đăng nhập

DESC: NV thu ngân đăng nhập vào hệ thống với tài khoản của mình. Không có chức năng tự động đăng nhập cho những lần sau(buộc mỗi lần sử dụng hệ thống phải đăng nhập)

RAT: Nhân viên đăng nhập vào hệ thống

DEP: Không.

ID: FR2

TITLE: Thêm hóa đơn

DESC: Khi khách hàng có yêu cầu thanh toán hóa đơn, NV sẽ thêm mới một hóa đơn và nhập thông tin cá nhân của khách hàng.

RAT: Tạo mới một hóa đơn cho khách hàng.

DEP: **FR1**.

ID: FR3

TITLE: Sửa thông tin cá nhân của khách hàng

DESC: Khi thông tin của khách hàng bị sai, hoặc thiếu → NV sẽ cập nhật lại thông tin theo đúng thông tin cá nhân hiện tại của khách hàng.

RAT: Sửa thông tin của khách hàng.

DEP: **FR2**.

ID: FR4

TITLE: Thêm món ăn cho chi tiết hóa đơn

DESC: Khi khách hàng yêu cầu gọi món → NV cập nhật lại danh sách món ăn trong chi tiết hóa đơn.

RAT: Khách hàng thêm món ăn

DEP: **FR2**.

ID: FR5

TITLE: Sửa món ăn cho chi tiết hóa đơn

DESC: Thông tin món ăn bị sai lệch so với yêu cầu của khách hàng → NV sẽ cập nhật lại thông tin cho đúng.

RAT: Chỉnh sửa thông tin món ăn.

DEP: **FR4**.

ID: FR6

TITLE: Thanh toán hóa đơn.

DESC: Phần mềm tự động tính tổng giá trị của hóa đơn đó.

RAT: Tính tổng tiền hóa đơn.

DEP: **FR2, FR4**.

ID: FR7

TITLE: In hóa đơn

DESC: Máy tính kết nối với thiết bị in → in ra hóa đơn đã thanh toán cho khách hàng.

RAT: In hóa đơn.

DEP: **FR6**.

b. Người quản lý.

ID: FR8

Tính năng: Đăng nhập

DESC: Người quản lý đăng nhập tài khoản bằng tài khoản admin và không có chức năng đăng nhập tự động cho những lần sau.

Kịch bản: Đăng nhập thành công:

- Những người quản lý muốn đăng nhập vào hệ thống, thì phải đăng nhập bằng tài khoản của quản trị viên.

- Sau đó, người quản lý có thể thao tác nghiệp vụ trên phần mềm với vai trò là người quản trị.

ID: FR9

Tính năng: Cập nhật thực đơn trong CSDL

DESC: Người quản lý có thể thêm/sửa/xóa thông tin món ăn trong Menu món ăn của nhà hàng trong CSDL chỉ sau khi đã đăng nhập thành công.

Kịch bản 1: Thêm món ăn mới:

- Khi người quản lý thêm một món ăn mới.
- Món ăn mới đó được thêm vào trong danh sách Menu món ăn trong CSDL

Kịch bản 2: Sửa thông tin món ăn:

- Khi người quản lý sửa xong thông tin của một món ăn.
- Thông tin món ăn đó được cập nhật vào trong CSDL

Kịch bản 3: Xóa thông tin món ăn:

- Khi người quản lý xóa 1 món ăn.
- Món ăn đó bị xóa khỏi CSDL

ID: FR10

Tính năng: Phân quyền người dùng

DESC: Tạo tài khoản và phân quyền cho tài khoản đó để cung cấp cho nhân viên

Kịch bản: NV được cấp quyền tương ứng với tài khoản mà người quản lý cung cấp:

- Lưu thông tin về phân quyền cho từng nhân viên → để phục vụ cho việc xác thực tài khoản mỗi lần NV đăng nhập.

ID: FR11

Tính năng: Tổng hợp doanh thu

DESC: Thống kê doanh thu nhà hàng theo khoảng thời gian xác định

Kịch bản: Tổng hợp doanh thu của nhà hàng theo thời gian:

- Định ra khoảng thời gian để tổng hợp doanh thu
- Hệ thống trả về: doanh thu và danh sách các hóa đơn đã được thanh toán thuộc trong thời gian đó.

3. Yêu cầu về hiệu năng

ID: QR1

TAG: Độ tin cậy

DESC: Khi khách hàng có yêu cầu thanh toán hóa đơn, NV thu ngân tổng hợp lại thông tin hóa đơn → tính giá trị hóa đơn chính xác

RAT: đảm bảo cung cấp thông tin chính xác đến khách hàng.

DEP: Không.

ID: QR2

TAG: Thời gian thực thi

DESC: Khi có yêu cầu thanh toán của khách hàng, kết quả thanh toán đưa ra cho khách hàng là nhanh. Có thể thực hiện tính giá trị hóa đơn cho tối đa 500 lượt khách vào trong 24 giờ

MUST: 100% lần thực hiện thì thời gian không quá 1 giây.

ID: QR3

TAG: Tính dễ sử dụng

DESC: ngôn ngữ, giao diện dễ hiểu, các biểu tượng mang ý nghĩa nhất quán

MUST: 100% lần thực hiện đều đạt.

ID: QR4

TAG: Tính ổn định

DESC: Hệ thống luôn sẵn sàng phục vụ, hoạt động tốt 24/24

MUST: 100% lần thực hiện đều đạt.

ID: QR5

TAG: Tính bảo mật

DESC:

- Người sử dụng được đăng nhập với tài khoản duy nhất, và không thể đăng nhập với tài khoản khác.
- Phân quyền cho người sử dụng đến từng chức năng.
- Đảm bảo khả năng backup dữ liệu và phục hồi hệ thống.

MUST: 100% lần thực hiện đều đạt.

PHỤ LỤC 4
CÁC TIÊU CHUẨN LIÊN QUAN ĐẾN ĐẢM BẢO CHẤT LƯỢNG VÀ
KIỂM THỬ PHẦN MỀM
ISO/IEC/IEEE 29119

Ký hiệu tài liệu	Tên tài liệu
Các tiêu chuẩn liên quan đến kiểm thử phần mềm	
IEEE 829:2008	A standard for the format of documents used in different stages of software testing. <i>(Tiêu chuẩn về mẫu các tài liệu được sử dụng trong các giai đoạn khác nhau của kiểm thử phần mềm)</i>
IEEE 1061:1998	A methodology for establishing quality requirements, identifying, implementing, analyzing, and validating the process and product of software quality metrics is defined. <i>(Phương pháp luận để thiết lập các yêu cầu chất lượng, xác định, thực thi, phân tích, và xác nhận quá trình và sản phẩm có các chỉ số chất lượng phần mềm được xác định.)</i>
IEEE 1059:1993	Guide for Software Verification and Validation Plans <i>(Hướng dẫn kế hoạch kiểm tra và đánh giá phần mềm)</i>
IEEE 1008:1987	A standard for unit testing. <i>(Tiêu chuẩn về kiểm thử đơn vị)</i>
IEEE 1012:2004	A standard for Software Verification and Validation. <i>(Tiêu chuẩn về kiểm tra và đánh giá phần mềm)</i>
IEEE 1028:2008	A standard for software inspections. <i>(Tiêu chuẩn về kiểm tra phần mềm)</i>
IEEE 1044:2009	A standard for the classification of software anomalies. <i>(Tiêu chuẩn về phân loại những bất thường phần mềm)</i>
IEEE 830:1998	A guide for lập trình viêneloping system requirements specifications. <i>(Hướng dẫn phát triển đặc tả các yêu cầu hệ thống)</i>
IEEE 730:2014	A standard for software quality assurance plans. <i>(Tiêu chuẩn về kế hoạch đảm bảo chất lượng phần mềm)</i>
IEEE 1061:1998	A standard for software quality metrics and methodology. <i>(Tiêu chuẩn về hệ phương pháp luận và các chỉ số chất lượng phần mềm)</i>
BS 7925-1: 1998	A vocabulary of terms used in software testing. <i>(Từ vựng về các thuật ngữ được sử dụng trong kiểm thử phần mềm)</i>

Ký hiệu tài liệu	Tên tài liệu
BS 7925-2:1998	A standard for software component testing. (<i>Tiêu chuẩn về kiểm thử thành phần phần mềm</i>)
ISO/IEC/IEEE 29119-1:2013	Software and systems engineering - Software testing - Part 1: Concepts and definitions (<i>Kỹ thuật hệ thống và phần mềm - Kiểm thử phần mềm - Phần 1: Khái niệm và định nghĩa</i>).
ISO/IEC/IEEE 29119-2:2013	Software and systems engineering - Software testing - Part 2: Test processs (<i>Kỹ thuật hệ thống và phần mềm - Kiểm thử phần mềm - Phần 4: Các kỹ thuật kiểm thử</i>).
ISO/IEC/IEEE 29119-3:2013	Software and systems engineering - Software testing - Part 3: Test documentation (<i>Kỹ thuật hệ thống và phần mềm - Kiểm thử phần mềm - Phần 3: Tài liệu kiểm thử</i>).
ISO/IEC/IEEE 29119-4:2015	Software and systems engineering - Software testing - Part 4: Test techniques (<i>Kỹ thuật hệ thống và phần mềm - Kiểm thử phần mềm - Phần 4: Kỹ thuật kiểm thử</i>).
Các tiêu chuẩn liên quan đến chất lượng phần mềm	
Bộ tiêu chuẩn ISO/IEC 14598	Information technology - Software product evaluation (CNTT - <i>Đánh giá sản phẩm phần mềm</i>).
Bộ tiêu chuẩn ISO/IEC 9126	Software engineering - Product quality (<i>Kỹ thuật phần mềm - Chất lượng sản phẩm</i>).
ISO/IEC 25010:2011	Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.
Các tiêu chuẩn vòng đời phần mềm	
ISO/IEC 12207:2013	System and software engineering - Software life cycle processes (<i>Tiêu chuẩn về các quá trình vòng đời phần mềm và dữ liệu vòng đời</i>)