

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

LUẬN VĂN THẠC SĨ KHOA HỌC

NGÀNH: CÔNG NGHỆ THÔNG TIN

**PHÁT TRIỂN PHẦN MỀM
ÁP DỤNG CÁC PHƯƠNG PHÁP SCRUM VÀ
EXTREME PROGRAMMING**

PHẠM QUANG HOÀ

HÀ NỘI 2006

MỤC LỤC

LỜI NÓI ĐẦU	4
CHƯƠNG 1 - TỔNG QUAN	5
1.1. Giới thiệu và đánh giá một số dự án đã triển khai	5
1.1.1. Giới thiệu về các dự án đã triển khai.....	5
1.1.2. Đánh giá các dự án đã triển khai	6
1.1.3. Một số kinh nghiệm được rút ra	8
1.2. Tổng quan về quản lý dự án và phát triển phần mềm	9
1.2.1. Định nghĩa dự án và quản lý dự án.....	10
1.2.2. Các lĩnh vực trong quản lý dự án	13
1.2.3. Vòng đời dự án và quá trình phát triển dự án.....	14
1.3. Các phương pháp phát triển phần mềm.....	17
1.3.1. Các phương pháp truyền thống	18
1.3.2. Các phương pháp phát triển nhanh.....	19
1.4. Kết chương	22
CHƯƠNG 2 - MỘT SỐ PHƯƠNG PHÁP PHÁT TRIỂN NHANH TIÊU	
BIỂU	23
2.1. Extreme Programming	23
2.1.1. Giới thiệu	23
2.1.2. Bốn đại lượng của một dự án	24
2.1.3. Các giá trị của XP	27
2.1.4. Các nguyên tắc.....	29
2.1.5. Quy trình XP.....	32
2.1.6. Hướng dẫn thực hiện	35
2.1.7. Nhận xét.....	39
2.2. Scrum.....	41
2.2.1. Giới thiệu	41
2.2.2. Quy trình.....	42
2.2.3. Nhóm dự án Scrum.....	45
2.2.4. Một số nét đặc trưng của Scrum.....	46
2.2.5. Một số ưu điểm của Scrum.....	47
2.2.6. Nhận xét.....	47
2.3. Phương pháp phát triển phần mềm thích nghi	48
2.3.1. Giới thiệu	48
2.3.2. Quy trình.....	48
2.3.3. Nhận xét.....	52
2.4. Đánh giá và so sánh các phương pháp	52
2.4.1. Những đặc điểm chính.....	53
2.4.2. Khả năng và phạm vi áp dụng	54

CHƯƠNG 3 - PHÁT TRIỂN PHẦN MỀM ÁP DỤNG SCRUM VÀ EXTREME PROGRAMMING	56
3.1. Quy trình phát triển phần mềm	56
3.1.1. Xác định mục tiêu dự án.....	57
3.1.2. Khảo sát và lấy yêu cầu khách hàng.....	57
3.1.3. Phân tích yêu cầu.....	59
3.1.4. Cài đặt các chức năng.....	60
3.1.5. Trình bày kết quả.....	60
3.1.6. Đưa ra các sản phẩm thử nghiệm	61
3.1.7. Kết thúc.....	61
3.2. Một số biện pháp tăng cường trong quản lý.....	62
3.2.1. Làm việc tập trung.....	62
3.2.2. Giảm chu kỳ phát hành.....	63
3.2.3. Thảo luận hàng ngày	64
3.2.4. Khách hàng cùng tham gia phát triển.....	65
3.3. Một số biện pháp tăng cường trong phát triển phần mềm	66
3.3.1. Lập trình theo cặp.....	66
3.3.2. Áp dụng các phương pháp kiểm thử	68
3.3.3. Thiết kế đơn giản.....	72
3.3.4. Tích hợp liên tục.....	73
3.3.5. Đưa ra các chuẩn trong lập trình	73
3.4. Kết chương	74
CHƯƠNG 4 - ÁP DỤNG THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ NGHIÊN CỨU	76
4.1. Môi trường áp dụng.....	76
4.1.1. Về tổ chức.....	76
4.1.2. Về nhân lực.....	77
4.1.3. Về công nghệ.....	77
4.1.4. Đánh giá.....	78
4.2. Giới thiệu một số dự án thử nghiệm.....	78
4.2.1. Dự án phần mềm lập thời khoá biểu	78
4.2.2. Dự án Phần mềm quản lý bán hàng.....	81
4.2.3. Dự án Phần mềm quản lý nhà hàng phiên bản 2	84
4.3. Đánh giá chung.....	85
KẾT LUẬN	87
TÀI LIỆU THAM KHẢO.....	89

DANH MỤC CÁC BẢNG

Bảng 4.1 – Đánh giá kết quả dự án 1	81
Bảng 4.2 – Đánh giá kết quả dự án 2	83

DANH MỤC CÁC HÌNH VẼ

Hình 1.1 - Quá trình thực hiện dự án	15
Hình 2.1 - Quy trình XP	33
Hình 2.2 - Tỷ lệ thành công tăng khi đáp ứng tốt những thay đổi.....	42
Hình 2.3 - Quy trình Scrum.....	42
Hình 2.4 - Quy trình của ASD	49
Hình 3.1 – Quy trình phát triển phần mềm đề xuất	62
Hình 3.2 – Quy trình kiểm thử TDD.....	70
Hình 4.1 – Cơ cấu tổ chức công ty.....	77

LỜI NÓI ĐẦU

Trong quá trình làm việc, tôi đã từng tham gia vào nhiều dự án tin học ở các công ty. Một trong những điều tôi thấy rõ nhất ở các dự án, đó là tỉ lệ thành công thường không cao. Rất nhiều dự án bị chậm tiến độ, không thoả mãn yêu cầu người sử dụng và trầm trọng hơn là không đúng nghiệp vụ.

Có thể kể ra đây một số nguyên nhân khiến cho dự án không được thành công là: Quy trình quản lý dự án không tốt, công nghệ áp dụng lỗi thời, khả năng của người phát triển có giới hạn và sự cộng tác với khách hàng không được đảm bảo.

Xuất phát từ lý do đó nên tôi đã chọn nghiên cứu lĩnh vực quản lý dự án và các phương pháp phát triển phần mềm, với mục đích là làm sao giảm được rủi ro khi thực hiện dự án, đưa ra được sản phẩm có chất lượng cao nhất mà vẫn đảm bảo thực hiện đúng tiến độ.

Trong luận văn này, tôi tập trung nghiên cứu một số phương pháp phát triển phần mềm tiên tiến hiện đang được chú ý của các nhà phát triển phần mềm trên thế giới, và lựa chọn cách áp dụng phù hợp với điều kiện thực tế của công ty.

Tôi xin được gửi lời cảm ơn chân thành đến thầy giáo TS. Huỳnh Quyết Thắng đã tận tình hướng dẫn, cảm ơn công ty Giải pháp kỹ thuật quốc tế đã tạo điều kiện để tôi có thể áp dụng thử nghiệm những kiến thức được nghiên cứu.

CHƯƠNG 1 - TỔNG QUAN

1.1. Giới thiệu và đánh giá một số dự án đã triển khai

Phần này giới thiệu một số dự án đã triển khai và đánh giá mức độ thành công của từng dự án, đồng thời phân tích nguyên nhân hạn chế sự thành công của dự án.

1.1.1. Giới thiệu về các dự án đã triển khai

Trong quá trình làm việc tại công ty Giải pháp kỹ thuật quốc tế (ITS) tôi đã tham gia phát triển một số dự án phần mềm với quy mô từ nhỏ tới trung bình với vai trò là người phát triển.

Dự án đầu tiên mà tôi tham gia là dự án Hệ thống quản lý công ty xe đạp ViHa. Khách hàng là công ty xe đạp ViHa. Đây là dự án đã được triển khai, nhưng không được áp dụng trong thực tế do sự thay đổi cơ cấu tổ chức của đơn vị khách hàng. Nhiều quy trình quản lý và quy trình nghiệp vụ của các phòng ban thay đổi, do đó các chức năng của phần mềm không còn phù hợp nữa.

Dự án thứ hai là Hệ thống quản lý đường sắt Thanh Hoá. Khách hàng là Xí nghiệp quản lý đường sắt Thanh Hoá. Dự án này có quy mô trung bình, với mục tiêu là xây dựng hệ thống phần mềm quản lý nghiệp vụ và các phần mềm hỗ trợ kỹ thuật cho các phòng ban. Dự án bắt đầu từ năm 2001 và kết thúc năm 2004.

Dự án thứ ba là Hệ thống quản lý nâng cao năng lực điều hành Trung tâm điều độ hệ thống điện quốc gia. Khách hàng là Trung tâm điều độ hệ thống điện quốc gia. Đây là một dự án mức độ trung bình, với mục tiêu là xây dựng các phân hệ phần mềm phục vụ cho từng phòng ban trong trung tâm, và

các phân hệ này có liên hệ chặt chẽ với nhau tuân thủ quy trình làm việc hiện thời của đơn vị khách hàng. Dự án bắt đầu từ năm 2003 và kết thúc vào năm 2006.

Dự án thứ tư là dự án phần mềm Quản lý nhà hàng thông minh, được xây dựng với mục đích quản lý toàn bộ hoạt động của một nhà hàng. Phần mềm được xây dựng sao cho có thể tùy biến một cách nhanh chóng theo yêu cầu của từng khách hàng, với đầy đủ các mảng chức năng liên quan như: Bán hàng, quản lý kho hàng, quản lý khách hàng... Dự án bắt đầu năm 2004 và kết thúc phiên bản 1.0 vào năm 2006, và đã áp dụng ở một số nhà hàng. Phiên bản tiếp theo đang trong quá trình phát triển.

1.1.2. Đánh giá các dự án đã triển khai

Qua một số dự án đã triển khai, theo tôi thì các dự án này chưa hẳn đã là thành công. Còn có rất nhiều vấn đề tồn tại trong việc phát triển phần mềm cũng như việc phân phối phần mềm tới người sử dụng.

Các dự án được đánh giá là không thành công như mong đợi là các dự án Hệ thống quản lý đường sắt Thanh Hoá và dự án Hệ thống quản lý nâng cao năng lực điều hành trung tâm điều độ hệ thống điện Quốc gia.

Dự án Hệ thống quản lý đường sắt Thanh Hoá đã được triển khai và áp dụng. Tuy nhiên do đặc thù của đơn vị khách hàng là các quy trình nghiệp vụ mang tính kỹ thuật cao, có rất nhiều phần mềm chuyên dụng cho từng công việc cụ thể nên việc áp dụng các phần mềm thuộc dự án còn rất hạn chế.

Đối với dự án Hệ thống quản lý nâng cao năng lực điều hành trung tâm điều độ hệ thống điện quốc gia, có thể nói đây là một dự án chỉ thành công ở mức vừa phải. Thứ nhất, thời gian thực hiện dự án kéo dài tới trên ba năm nên

chi phí nhân công và chi phí thiết bị cho dự án này là rất lớn. Thứ hai, do thời gian kéo dài nên rất nhiều quy trình nghiệp vụ và các văn bản pháp quy đã thay đổi, điều này làm cho một số phân hệ phần mềm không phục vụ tốt cho công việc của khách hàng. Thứ ba, do quy trình phát triển phần mềm này còn yếu kém, tài liệu không đầy đủ nên việc bảo hành bảo trì rất khó khăn, gây nhiều phiền hà cho khách hàng.

Có thể đưa ra đây một số nguyên nhân dẫn đến việc không thành công của các dự án này như sau:

Trước tiên, đó là việc trao đổi với khách hàng không được tiến hành thường xuyên. Việc tìm hiểu quy trình chủ yếu thông qua một số buổi lấy yêu cầu khách hàng, với thời gian có hạn. Chính vì lý do đó nên nhiều quy trình nghiệp vụ người phát triển không nắm được đầy đủ.

Tiếp đến, đó là các thủ tục hành chính liên quan đến dự án khiến dự án phải kéo dài và khó kết thúc.

Và những nguyên nhân chính dẫn đến dự án không thành công nằm về phía những người quản lý và phát triển dự án. Người quản lý không đưa ra được một quy trình hợp lý nên dẫn đến việc phát triển các phân hệ của hệ thống hoàn toàn phụ thuộc vào người phát triển phân hệ đó. Điều này gây rất nhiều khó khăn khi đội ngũ phát triển thay đổi nhân sự, người tiếp quản một công việc nào đó thiếu nhiều tài liệu nên phải mất một khoảng thời gian để hiểu được công việc của người trước đó. Thêm vào đó, trình độ của những người phát triển không đồng đều, nên việc xảy ra lỗi trong các phần mềm là thường xuyên. Các lỗi này làm giảm đáng kể chất lượng của phần mềm đưa ra.

Dự án được đánh giá là tương đối thành công, đó là các dự án Phần mềm quản lý nhà hàng. Tuy không thực sự đáp ứng được đầy đủ các yêu cầu của khách hàng nhưng nói chung phần mềm đáp ứng được những công việc quản lý chính mà một nhà hàng cần, và được khách hàng đánh giá tốt.

Có thể đưa ra một số nguyên nhân thành công của dự án này, như sau: Thứ nhất, khi triển khai dự án những người phát triển nhận được sự hợp tác đầy đủ từ phía khách hàng. Thứ hai, quá trình phát triển các chức năng được tiến hành song song với quá trình khai thác phần mềm, do đó các lỗi phần mềm nhanh chóng được cập nhật và xử lý.

1.1.3. Một số kinh nghiệm được rút ra

Qua việc phân tích và đánh giá các phần mềm đã triển khai, có thể rút ra một số kinh nghiệm như sau:

Thứ nhất, việc liên hệ thường xuyên với khách hàng là điều rất quan trọng, bởi khách hàng là những người am hiểu nhất về nghiệp vụ, đồng thời họ biết những gì mà phần mềm phải đáp ứng. Ngoài ra, khách hàng đóng vai trò quan trọng trong việc kiểm thử phần mềm, phát hiện lỗi cũng như các chức năng không phù hợp.

Thứ hai, việc quản lý dự án cần phải được chú trọng. Để làm được điều này, cần người quản lý có kinh nghiệm, khả năng lập kế hoạch tốt và nhanh nhạy trong việc xử lý tình huống.

Thứ ba, cần phải có một quy trình phát triển phần mềm hiệu quả. Quy trình tốt sẽ làm tăng khả năng làm việc của từng thành viên, chuẩn hoá các tài liệu, từ đó giảm bớt các tác động tiêu cực khi đội ngũ phát triển thay đổi.

Trong các chương tiếp theo của luận văn, tôi sẽ trình bày một số phương pháp phát triển phần mềm đang được chú ý hiện nay. Các phương pháp này áp dụng tốt cho các dự án có phạm vi vừa và nhỏ, phù hợp với thực tế của nhiều công ty phần mềm hiện nay.

1.2. Tổng quan về quản lý dự án và phát triển phần mềm

Việc phát triển bất cứ sản phẩm nào đều cần phải giải quyết rất nhiều các vấn đề nảy sinh. Đặc biệt với dự án công nghệ thông tin, có thể liệt kê ra đây một số vấn đề sau:

Khi bắt đầu dự án, người quản lý phải xác định được chi phí nhân lực, vật tư và các chi phí khác cần thiết để tiến hành dự án. Việc xác định này tương đối khó khăn, do đặc thù sản phẩm phần mềm là sản phẩm trí tuệ, mang nhiều yếu tố ngẫu nhiên và khó định hình trước.

Trong quá trình phát triển phần mềm, yêu cầu khách hàng thường xuyên thay đổi. Các thay đổi này có thể là do chủ quan khách hàng, cũng có thể do khách quan. Khi đó vấn đề đáp ứng sự thay đổi này là cần thiết.

Thêm vào đó, đội ngũ phát triển phần mềm cũng có thể bị thay đổi. Đây là một vấn đề tất yếu không thể tránh khỏi, vì thế cần phải có các biện pháp nhằm giảm thiểu rủi ro khi gặp phải vấn đề này.

Ngoài ra, khi sản phẩm hoàn thành khâu phát triển, thì khâu phát hành và bảo trì cũng rất quan trọng. Với một số dự án phần mềm, khâu phát hành là yếu tố quyết định sự thành công của toàn bộ dự án. Khi phát hành, cần phải chú ý đến các yếu tố như thời điểm phát hành, mạng lưới phân phối, các chính sách bảo hành bảo trì phần mềm và vấn đề nâng cấp phiên bản.

Từ những lý do trên, nên cần phải quản lý dự án và áp dụng các kỹ thuật lập trình trong phát triển phần mềm. Tuy rằng việc áp dụng các phương pháp đó không thể giải quyết được toàn bộ các vấn đề nảy sinh, nhưng sẽ góp phần hạn chế rủi ro, nâng cao chất lượng phần mềm và giảm chi phí.

1.2.1. Định nghĩa dự án và quản lý dự án

Theo như các định nghĩa được chấp nhận rộng rãi và được cung cấp bởi tổ chức Project Management Institute (PMI) – một tổ chức thành lập vào năm 1969 chuyên về lĩnh vực quản lý dự án – thì dự án và quản lý dự án được định nghĩa như sau [3]:

Dự án là một sự nỗ lực tạm thời, đảm bảo hoàn thành một mục đích duy nhất. Quản lý dự án là việc áp dụng những kiến thức, kỹ năng, công cụ và các kỹ thuật vào các hoạt động của dự án với mục đích đạt được hoặc vượt những yêu cầu và sự mong đợi của nhà đầu tư.

Dự án còn được xem xét dưới góc độ những thuộc tính của dự án, các thuộc tính này bao gồm:

Khung thời gian: Bởi vì dự án mang tính chất tạm thời nên thời gian bắt đầu và kết thúc dự án phải được định nghĩa. Thông thường, các dự án được bắt đầu vào một ngày được định trước và ngày kết thúc được ước lượng, lên kế hoạch. Đôi khi, một dự án mà ngày kết thúc không thể thay đổi được, thì khi đó người ta thực hiện ngược lại, tức là phải tính toán thời gian bắt đầu dự án sao cho có thể đảm bảo kết thúc đúng thời hạn.

Mục tiêu: Một dự án được đảm bảo phải hoàn thành một mục đích nào đó. Trong một dự án công nghệ thông tin, thì kết quả có thể là một hệ thống, một sản phẩm phần mềm hoặc các kết quả mang tính nghiên cứu. Do đó, mục

tiêu của dự án phải được định nghĩa một cách rõ ràng để có thể lên kế hoạch những công việc phải làm, đồng thời giúp cho nhóm phát triển thực hiện công việc đúng hướng và có hiệu quả. Thông thường, dự án là là kết quả hợp đồng giữa khách hàng và đơn vị phát triển, nên các mục tiêu dự án cần được sự đồng ý của hai bên. Mục tiêu này phải đáp ứng những điều mà khách hàng cần và mong đợi, do đó để đạt được sự thoả mãn của khách hàng thì cần phải đạt được các mục tiêu đã đề ra.

Quyền sở hữu: kết quả dự án là phải đem lại lợi ích cho một người hoặc một tổ chức nào đó, do đó cần phải xác định một cách rõ ràng người sở hữu sản phẩm khi dự án kết thúc. Ngoài ra, cần xác định rõ những người phải trả những khoản chi phí phát triển cũng như bảo trì hệ thống sau khi đưa vào sử dụng.

Tài nguyên: Để thực hiện các dự án CNTT, cần phải có thời gian, tiền bạc, nhân lực và công nghệ. Những tài nguyên này không thể thiếu để có thể đạt được mục tiêu. Mục tiêu của dự án xác định trực tiếp phạm vi dự án, là những gì cần phải đạt được, và chúng ta có thể dựa vào đó để có thể tính toán được những tài nguyên cần thiết cho dự án.

Vai trò của những người tham gia dự án: Một dự án công nghệ thông tin, các thành viên có thể có vai trò khác nhau và chịu trách nhiệm trong các lĩnh vực khác nhau. Mặc dù mỗi dự án một khác, nhưng trong một dự án tiêu biểu thường có:

- **Quản lý dự án:** là người đứng đầu nhóm phát triển, chịu trách nhiệm chính về quản lý dự án, cũng như việc thực hiện dự án theo các quy trình kỹ thuật theo các yêu cầu và các chuẩn đã được đưa ra.

- **Chủ đầu tư:** chủ đầu tư có thể là khách hàng, hoặc là người quản lý trong trường hợp công ty sản xuất sản phẩm bán rộng rãi, là người cung cấp các tài nguyên để thực hiện dự án.
- **Các nhà chuyên môn:** những nhà chuyên môn có thể là khách hàng, hoặc những người dùng cuối, là những người có kiến thức chuyên môn trong lĩnh vực của mình, có thể cung cấp kinh nghiệm, kiến thức phục vụ cho việc phát triển hệ thống. Ví dụ, khi thực hiện một hệ thống phục vụ công việc kế toán, nếu trong đội phát triển có thêm những người am hiểu nghiệp vụ kế toán chia sẻ những kiến thức của họ trong việc phát triển hệ thống thì sẽ hiệu quả hơn là việc những người phát triển phải học toàn bộ những kiến thức về kế toán.
- **Chuyên gia kỹ thuật:** cần phải có những chuyên gia kỹ thuật trong việc cung cấp các giải pháp kỹ thuật hiệu quả để giải quyết vấn đề. Có nhiều chuyên gia kỹ thuật trong các lĩnh vực khác nhau, như phân tích hệ thống, giải pháp mạng, thiết kế đồ họa, lập trình viên... Các chuyên gia này chịu trách nhiệm thiết kế, cài đặt và giải quyết các vấn đề trong lĩnh vực của mình.

Rủi ro: mọi dự án đều tiềm ẩn những rủi ro. Rủi ro có thể nảy sinh từ những nguyên nhân bên trong nhóm thực hiện dự án, ví dụ như việc một thành viên quan trọng rời khỏi nhóm phát triển khi dự án đang trong quá trình thực hiện. Những nguyên nhân rủi ro từ bên ngoài có thể do việc phải lệ thuộc vào những nhà cung cấp, khách hàng hoặc thị trường. Như vậy cần phải coi rủi ro như là một yếu tố hoàn toàn có thể xảy ra và phải chuẩn bị để đáp ứng được với các rủi ro này.

Sự phụ thuộc lẫn nhau của các công việc: trong dự án, có nhiều công việc phụ thuộc vào các công việc khác. Nếu một công việc không được hoàn thành đúng hạn, hoặc thất bại thì có thể kéo theo nhiều công việc khác bị trễ hoặc không thực hiện được, làm ảnh hưởng chung đến toàn bộ dự án.

Việc xem xét các thuộc tính của dự án cho ta một cái nhìn toàn diện hơn về dự án, để từ đó có thể đưa ra được những quyết định đúng đắn khi thực hiện dự án.

1.2.2. Các lĩnh vực trong quản lý dự án

Nội dung quản lý dự án bao gồm những lĩnh vực chính sau:

- ***Quản lý tính thống nhất*** – Tập trung vào việc thực hiện kế hoạch và xử lý thay đổi trong quá trình thực hiện.
- ***Quản lý phạm vi*** – Là việc đảm bảo các công việc của dự án được định nghĩa một cách chính xác và hoàn thành theo kế hoạch.
- ***Quản lý thời gian*** – Cần phải xác định các giai đoạn và các công việc của dự án, sau đó sắp lịch, ước lượng thời gian, gán tài nguyên cho từng công việc và quản lý quá trình thực hiện sao cho dự án được thực hiện đúng tiến độ.
- ***Quản lý chi phí*** – Đảm bảo ngân sách cho việc thực hiện và hoàn thành dự án.
- ***Quản lý chất lượng*** – Tập trung vào việc quản lý việc lập kế hoạch, thực hiện kế hoạch sao cho kết quả đạt được hoặc vượt yêu cầu và sự mong đợi của nhà đầu tư.
- ***Quản lý nhân lực*** – Con người là tài nguyên quan trọng nhất của một dự án. Quản lý nhân lực tập trung trong việc tạo lập và phát

triển đội ngũ phát triển cũng như việc sử dụng hiệu quả nguồn nhân lực hiện có.

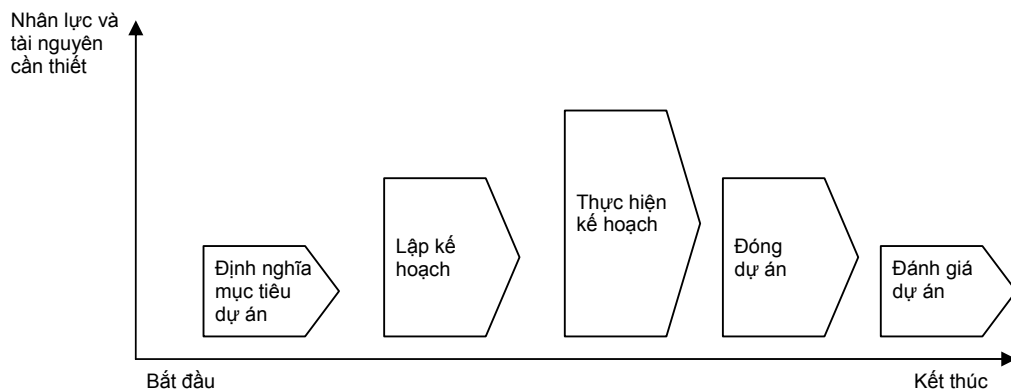
- ***Quản lý việc liên lạc*** – Giữ liên lạc thường xuyên giữa những người phát triển dự án và nhà đầu tư.
- ***Quản lý rủi ro*** – Các dự án luôn phải đối mặt với các rủi ro. Quản lý rủi ro là việc dự tính và đưa ra các biện pháp xử lý những rủi ro có thể xảy đến với dự án.
- ***Quản lý nguồn cung cấp*** – Rất nhiều tài nguyên cần thiết cho dự án phải đưa vào từ bên ngoài. Quản lý nguồn cung cấp đảm bảo cho việc cung cấp các tài nguyên được ổn định.

1.2.3. Vòng đời dự án và quá trình phát triển dự án

Vòng đời dự án là một tập các giai đoạn trong toàn bộ khoảng thời gian từ khi dự án bắt đầu đến khi kết thúc để định nghĩa, xây dựng và đưa ra sản phẩm của dự án [3]. Mỗi một giai đoạn cần phải đưa ra các kết quả công việc trong giai đoạn đó, như kế hoạch dự án, các tài liệu đặc tả, sản phẩm cuối...

Một dự án cần phải được chia thành các giai đoạn để có thể quản lý được dễ hơn và giảm rủi ro. Ở cuối mỗi giai đoạn cần đưa ra những kết quả đã thực hiện được trong giai đoạn đó. Việc xem xét các kết quả này cho phép xác định được hiệu quả của dự án và nhanh chóng đưa ra các biện pháp điều chỉnh hay giải quyết các vấn đề xuất hiện trong từng giai đoạn.

Mặc dù mỗi phương pháp phát triển phần mềm khác nhau có thể định nghĩa các giai đoạn của dự án khác nhau, phần sau đây đưa ra các giai đoạn chung nhất của một dự án.



Hình 1.1 - Quá trình thực hiện dự án

1.2.3.1. Định nghĩa mục tiêu dự án

Định nghĩa mục tiêu của toàn bộ dự án là bước đầu tiên của một dự án. Các mục tiêu được định nghĩa tốt sẽ giúp cho nhóm phát triển tập trung thực hiện công việc theo các mục đích rõ ràng. Ngoài ra, hầu hết các dự án đều có những đặc điểm sau khi bắt đầu dự án:

- Tại thời điểm bắt đầu dự án, nhân lực và vật lực cần thiết thường thấp, nhưng sẽ tăng dần trong quá trình thực hiện dự án, và sau đó lại giảm dần tới khi dự án kết thúc.
- Rủi ro tại thời điểm bắt đầu là cao nhất, một khi đã xác định được mục tiêu dự án và dự án được đưa vào thực hiện thì khả năng thành công sẽ tăng dần.
- Việc thay đổi phạm vi dự án dễ thực hiện nhất khi dự án mới bắt đầu. Càng về sau, chi phí để thay đổi phạm vi dự án và khắc phục lỗi càng cao.

1.2.3.2. Lập kế hoạch dự án

Một khi mục tiêu dự án đã được xác định, cần phải đưa ra kế hoạch thực hiện dự án. Kế hoạch dự án cần phải chỉ ra được [3]:

- Những công việc gì cần thực hiện? Đây chính là việc chi tiết hoá mục tiêu dự án thành các công việc cụ thể cần phải thực hiện.
- Thực hiện những công việc đó như thế nào? Cần phải đưa ra các giải pháp khả thi để thực hiện các công việc đã đề ra.
- Ai sẽ thực hiện những công việc đó? Cần phải xác định nhân lực phù hợp để thực hiện các công việc.
- Thực hiện trong thời gian bao lâu? Việc ước lượng thời gian thực hiện dự án cần phải được tiến hành.
- Chi phí thực hiện là bao nhiêu? Cần phải dự toán chi phí thực hiện dự án.
- Có gì không ổn không, và nếu có thì phải xử lý như thế nào? Đây chính là dự đoán các rủi ro có thể xảy đến và dự phòng phương án giải quyết trong trường hợp có rủi ro.
- Kế hoạch thực hiện như thế nào và phải dự trù ngân sách ra sao? Các công việc cần phải được lập lịch, và ngân sách để cho dự án hoạt động cũng phải được tính đến.

Ngoài ra, cũng cần chỉ ra các công việc phải làm, những tài nguyên cần thiết, thời gian thực hiện cũng như những kết quả cần đạt được cho mỗi giai đoạn của dự án.

1.2.3.3. Thực hiện dự án

Sau khi kế hoạch dự án được lập, thì bắt đầu tiến hành thực hiện các kế hoạch của dự án. Trong quá trình thực hiện dự án, phạm vi, nhân lực, lịch trình thực hiện cần phải được quản lý một cách tích cực để có thể đạt được

mục tiêu dự án. Cuối giai đoạn này, nhóm phát triển cần đưa ra được kết quả là sản phẩm của dự án.

1.2.3.4. Đóng dự án

Như đã đề cập ở trên, thời điểm kết thúc một dự án phải có hạn định. Việc đóng một dự án đảm bảo rằng toàn bộ các công việc đã hoàn thành theo kế hoạch và được xác nhận bởi nhóm phát triển và nhà đầu tư. Kết quả của dự án được trình bày với khách hàng để cho thấy toàn bộ những yêu cầu chỉ ra đã được hoàn thành.

1.2.3.5. Đánh giá dự án

Sau khi dự án được đóng, cần đánh giá dự án. Những người quản lý và những người phát triển cần phải đánh giá được độ thành công của dự án, những kinh nghiệm rút ra trong quá trình thực hiện dự án. Thêm vào đó, cần phải đánh giá xem dự án có được quản lý tốt, tuân thủ quy trình, đáp ứng các chuẩn và đưa ra đầy đủ các chức năng yêu cầu.

1.3. Các phương pháp phát triển phần mềm

Trong thời gian gần đây, rất nhiều các phương pháp phát triển phần mềm được đề xuất. Nhiều phương pháp đã được lý thuyết hoá thành các phương pháp luận. Trong dự án công nghệ thông tin, một phương pháp luận có thể được hiểu như là một tập các hoạt động thực tiễn được hệ thống hoá. Tùy theo phạm vi dự án, điều kiện thời gian và nhiều yếu tố khác mà có thể lựa chọn áp dụng các phương pháp khác nhau, hoặc kết hợp các phương pháp sao cho phù hợp.

Các phương pháp phát triển phần mềm có thể được phân chia thành hai lớp chính: các phương pháp truyền thống và các phương pháp phát triển

nhanh. Phần này sẽ cung cấp một cái nhìn tổng quan về hai lớp các phương pháp này. Các phương pháp phát triển nhanh, nội dung chính của luận văn, sẽ được đề cập kỹ hơn ở các phần sau.

1.3.1. Các phương pháp truyền thống

Các phương pháp truyền thống là các phương pháp thiên về kế hoạch, quá trình phát triển phần mềm phải tuân thủ quy trình một cách nghiêm ngặt. Trong quá trình phát triển phần mềm, rất nhiều tài liệu được tạo ra, được xét duyệt và đó là một yếu tố quan trọng trong quản lý rủi ro.

Với các phương pháp này, toàn bộ quá trình phát triển được lên kế hoạch chi tiết và các tài liệu trước cũng như trong khi phát triển được chuẩn bị đầy đủ. Quá trình phát triển được thực hiện theo các quy trình được định trước, và việc tuân thủ quy trình sẽ làm tăng chất lượng phần mềm và giảm rủi ro.

Theo các phương pháp này, thì quá trình phát triển phần mềm giống như sản xuất các mặt hàng công nghiệp khác. Những người phát triển thực hiện công việc một cách nghiêm ngặt theo các chuẩn và quy trình, không yêu cầu sáng tạo nhiều. Những người quản lý chỉ quan tâm đến việc tăng năng lực sản xuất và đạt được một số mục tiêu như [10]:

- Giảm thiểu lỗi và làm sao cho mọi công việc diễn ra trơn tru.
- Cố gắng giữ ổn định (về tổ chức, về sản lượng...)
- Chuẩn hoá mọi thao tác và bắt buộc người thực hiện phải tuân theo một cách nghiêm ngặt.
- Không cho phép sự sai sót

Các phương pháp này thường áp dụng cho các dự án lớn. Một số phương pháp tiêu biểu thuộc lớp này như: Waterfall Model, Capability Maturity Model. Sơ lược về các phương pháp này như sau:

Waterfall Model: Waterfall Model là một mô hình phát triển phần mềm tuần tự trong đó quá trình phát triển được xem như là một quá trình trôi đi đều đặn (giống như một thác nước) thông qua các giai đoạn: phân tích yêu cầu, thiết kế, cài đặt, kiểm thử, tích hợp và bảo trì. Thuật ngữ Waterfall được đề cập trong một bài viết xuất bản vào năm 1970 bởi W. W. Royce [9].

Capability Maturity Model (CMM): CMM thường được hiểu như là một cách tiếp cận nhằm cải tiến một quy trình dựa trên một quy trình đã có. CMM được phát triển bởi Software Engineering Institute (SEI) trường Carnegie Mellon ở Pittsburgh, Pennsylvania, Mỹ [7].

1.3.2. Các phương pháp phát triển nhanh

Các phương pháp phát triển nhanh được gọi với cái tên là Agile, theo nghĩa là nhanh nhẹn, khéo léo trong hành động, là các phương pháp dựa trên các quy trình phát triển nhanh. Điều này đặc biệt cần thiết trong lĩnh vực Internet và truyền thông di động hiện đang phát triển rất nhanh chóng.

Các dự án phát triển theo các phương pháp Agile dựa trên các giá trị thương mại, quá trình thực hiện dự án được điều khiển theo hướng đáp ứng thực tại hơn là theo kế hoạch. Việc quản lý rủi ro đạt được bằng một sự cộng tác chặt chẽ với khách hàng, giảm chu kỳ phát hành và tập trung thực hiện các chức năng quan trọng trước.

Các phương pháp phát triển nhanh ra đời cách đây không lâu. Nó được bắt đầu bởi Tuyên ngôn về các phương pháp phát triển phần mềm Agile được

đưa ra bởi một nhóm những người hoạt động trong lĩnh vực phần mềm vào năm 2001. Những người này đại diện cho các phương pháp như Extreme Programming (XP), Scrum, Crystal và các phương pháp khác cùng thống nhất đưa ra một bản tuyên ngôn. Nội dung bản tuyên ngôn có những điểm chính sau [4]:

Chúng ta dần phát hiện ra những cách phát triển phần mềm tốt hơn bằng cách thực hiện nó và giúp người khác thực hiện nó. Qua công việc này, chúng ta thu được các giá trị:

- Các cá nhân và sự tương tác với nhau quan trọng hơn các quy trình và các công cụ.
- Làm phần mềm quan trọng hơn việc lập tài liệu.
- Việc hợp tác với khách hàng quan trọng hơn việc ký kết hợp đồng.
- Đáp ứng thay đổi quan trọng hơn việc theo một kế hoạch.

Đó là, trong khi những thành phần bên phải là quan trọng, nhưng chúng ta coi trọng những thành phần bên trái hơn.

Chúng ta sẽ phân tích những điều được tuyên bố trong bản tuyên ngôn này.

Câu đầu tiên cho thấy, chúng ta không phải lúc nào cũng có giải pháp cho mọi thứ, mà giải pháp nằm chính bên trong của công việc. Và để tìm được giải pháp, thì không thể chỉ dựa vào các lý thuyết mà phải trực tiếp làm công việc phát triển phần mềm.

Những câu sau gồm hai phần, phần bên phải có giá trị ít hơn phần bên trái mặc dù điều này không có nghĩa là phần bên trái không quan trọng. Chúng ta sẽ xem ý nghĩa của từng câu.

Giá trị đầu tiên cho thấy những quy trình và công cụ là quan trọng. Sẽ không thể phát triển một phần mềm tốt nếu không có quy trình và công cụ tốt, vì lẽ đó nên dùng công cụ tốt nhất hiện có. Nhưng điều mà bản tuyên ngôn muốn nhấn mạnh là vai trò của từng cá nhân và mối quan hệ của các cá nhân với nhau trong đội ngũ phát triển phần mềm.

Đối với một dự án thành công, thì cần phải lập tài liệu một cách đầy đủ. Nhưng bản thân tài liệu sẽ không giúp được gì nếu không có sản phẩm thực sự. Vì thế, việc tạo ra sản phẩm phần mềm quan trọng hơn, và tài liệu chỉ đóng vai trò hỗ trợ phần mềm, mô tả phần mềm một cách chính xác.

Việc ký kết hợp đồng là quan trọng nhưng không đủ. Một môi trường hợp tác tốt sẽ giúp cho những người phát triển đưa ra giải pháp tốt nhất cho khách hàng. Các hợp đồng định nghĩa những điều khoản ràng buộc mà trong đó cả hai phía đối tác đều phải tuân theo, nhưng những người phát triển cần hợp tác với khách hàng để có thể hiểu rõ yêu cầu cũng như những gì cần phải đưa ra.

Và cuối cùng, chúng ta thấy là việc lập kế hoạch là không thể thiếu. Kế hoạch giúp công việc được định hướng tốt hơn. Tuy nhiên thực tế có rất nhiều thay đổi, và cứ nếu nhất nhất tuân theo kế hoạch thì có thể sẽ dẫn đến thất bại. Do đó cần phải đáp ứng những thay đổi để có thể điều chỉnh sao cho phù hợp.

Ở đây không có sự mâu thuẫn giữa các phương pháp truyền thống và các phương pháp phát triển nhanh. Vấn đề là ở chỗ những điều mà các

phương pháp phát triển nhanh và các phương pháp truyền thống chú trọng vào là khác nhau. Điểm chính của các phương pháp phát triển nhanh là việc đáp ứng thay đổi trong khi các phương pháp truyền thống tập trung vào kế hoạch.

Các phương pháp phát triển nhanh được đề cập kỹ hơn trong chương 2.

1.4. Kết chương

Qua chương này, chúng ta đã có một cái nhìn tổng thể về những vấn đề gặp phải trong quá trình thực hiện một dự án. Những vấn đề này thường làm cho việc thực hiện dự án bị chậm, chất lượng sản phẩm phần mềm không cao, chưa thoả mãn được mong muốn của khách hàng.

Từ đó cho thấy, cần phải nghiên cứu, áp dụng các phương pháp phát triển phần mềm phù hợp, có khả năng đáp ứng thay đổi nhanh để có thể đưa ra được sản phẩm phần mềm có chất lượng, thoả mãn mong muốn của khách hàng và đảm bảo tiến độ thực hiện.

Đã có nhiều phương pháp phát triển phần mềm được đề xuất. Gần đây, các phương pháp phát triển nhanh đang được chú ý nghiên cứu và áp dụng do những ưu điểm mà nó mang lại. Trong các chương sau, chúng ta sẽ tìm hiểu chi tiết hơn về các phương pháp phát triển nhanh này.

CHƯƠNG 2 - MỘT SỐ PHƯƠNG PHÁP PHÁT TRIỂN NHANH TIÊU BIỂU

Hiện nay, đã có nhiều phương pháp phát triển nhanh được đề xuất và áp dụng. Mỗi phương pháp có một cách tiếp cận khác nhau, đưa ra những quy trình, các hướng dẫn thực hiện riêng. Nhưng chung nhất, các phương pháp này đều có những tính chất đã được tuyên bố trong bản tuyên ngôn về các phương pháp phát triển nhanh như: tính tương tác cao, coi trọng vai trò khách hàng, khả năng đáp ứng thay đổi nhanh.

Chương này sẽ giới thiệu một số phương pháp phát triển phần mềm tiêu biểu thuộc lớp các phương pháp phát triển nhanh, bao gồm Extreme Programming (XP), Scrum và Adaptive Software Development (ASD).

Trong các phương pháp này, Scrum và ASD là các phương pháp thiên về lĩnh vực quản lý. Scrum đưa ra một quy trình chặt chẽ, trong đó nêu rõ vai trò của những thành viên tham gia dự án cũng như những hoạt động cần phải tiến hành trong quá trình thực hiện dự án. ASD đưa ra một khung quản lý chung hơn, trong đó có nhiều tùy chọn cho phép những người quản lý áp dụng linh hoạt. Trong khi đó, cách tiếp cận của XP thiên về các kỹ thuật áp dụng trong lập trình. Nhiều hướng dẫn thực hiện trong lĩnh vực lập trình được XP đề cập một cách chi tiết.

2.1. Extreme Programming

2.1.1. Giới thiệu

Extreme Programming (XP) là một trong những phương pháp phát triển nhanh tiêu biểu nhất. Phương pháp này được đề xuất và áp dụng từ khi cách tiếp cận nhanh còn chưa phổ biến rộng rãi. XP ra đời từ thực tiễn muốn

khắc phục các vấn đề gặp phải trong các cách tiếp cận truyền thống có chu kỳ phát triển phần mềm dài như phần mềm không đáp ứng yêu cầu khách hàng, khả năng áp dụng của sản phẩm thấp, hay không đảm bảo tiến độ thực hiện.

Dựa trên những kinh nghiệm thực tế đã có từ trước, cộng với những thành công qua quá trình áp dụng thử nghiệm, XP đã được tổng quát lên thành lý thuyết với một loạt các nguyên lý và các bài học thực tiễn. Khi mà cuốn sách *Extreme Programming Explained: Embrace Change* của Kent Beck ra đời, nó đã trở thành một trong những cuốn sách quan trọng đối với nhiều nhà phát triển. Phần này sẽ giới thiệu những điểm chính của XP.

2.1.2. Bốn đại lượng của một dự án

Trong một dự án, bốn đại lượng được XP nhấn mạnh, đó là: phạm vi, chi phí, chất lượng và thời gian. Các đại lượng này có liên hệ chặt chẽ với nhau và ảnh hưởng lẫn nhau, và việc thay đổi một đại lượng sẽ làm thay đổi các đại lượng còn lại. Việc quản lý các đại lượng này sẽ có tác động đến sản phẩm đầu ra.

2.1.2.1. Phạm vi

Phạm vi dự án cần phải được xác định rõ. Nếu phạm vi dự án lớn, hệ thống phức tạp thì cần phải đầu tư nhiều thời gian và tốn nhiều chi phí để có thể hoàn thành. Kích cỡ của một hệ thống phần mềm thường được xác định dựa trên số lượng các chức năng theo yêu cầu của người sử dụng. Tuy nhiên, quan hệ giữa phạm vi với các đại lượng khác không hoàn toàn tuyến tính, thậm chí việc thêm các chức năng đối với một số dự án lại có thể làm hệ thống đơn giản hơn.

2.1.2.2. Chi phí

Một điều dễ nhận thấy là nếu chi phí hạn hẹp thì sẽ dẫn đến chất lượng, phạm vi và thời gian dành cho dự án cũng bị hạn chế. Một dự án muốn có chất lượng tốt, phạm vi lớn thì cần phải có đầu tư tương xứng. Tuy nhiên, việc đầu tư nhiều tiền hơn vào một dự án không phải lúc nào cũng đảm bảo các đại lượng còn lại tốt hơn, mà còn có thể gây ra kết quả không mong muốn, thậm chí là ngược lại. Lấy ví dụ như việc trả lương cao hơn cho một người phát triển không hẳn là người đó sẽ làm được nhiều việc hơn trong một khoảng thời gian, hay hoàn thành công việc nhanh hơn, mà thậm chí còn có thể gây áp lực khiến hiệu quả công việc giảm xuống.

2.1.2.3. Chất lượng

Chất lượng của sản phẩm đầu ra phụ thuộc nhiều vào các yếu tố còn lại. Yêu cầu chất lượng sản phẩm càng cao, càng cần nhiều tiền và thời gian. Ngược lại, việc đầu tư nhiều thời gian và tiền sẽ góp phần nâng cao chất lượng phần mềm.

Tuy nhiên, việc đánh giá chất lượng sản phẩm còn phụ thuộc vào việc người đánh giá nhìn theo góc độ nào. Đối với khách hàng, thì một phần mềm có chất lượng có thể phải là phần mềm có một giao diện đẹp và dễ sử dụng, các chức năng được bố trí tương tự nhau để người sử dụng có thể dễ học hơn và thao tác nhanh hơn; hoặc việc cung cấp một dịch vụ liên tục, ổn định và tin cậy, dữ liệu được đảm bảo đồng nhất 100%; hay hệ thống cung cấp được nhiều nhất các chức năng mà người dùng yêu cầu, thậm chí cung cấp một số chức năng tốt hơn người dùng mong đợi.

Trong khi đó, đối với người phát triển, thì phần mềm có chất lượng là sản phẩm được thiết kế tốt, mã nguồn rõ ràng, các chức năng thực hiện một cách tối ưu nhất.

Từ những nhận xét trên, chúng ta có thể thấy rằng chất lượng của phần mềm phản ánh những yếu tố:

Các yêu cầu của hệ thống mà phần mềm đáp ứng: Đó chính những gì mà sản phẩm đầu ra có được dưới góc nhìn của người dùng cuối. Việc so sánh giữa trạng thái hiện thời của sản phẩm với những yêu cầu ban đầu cung cấp một cách đánh giá chất lượng của phần mềm. Cũng nên chú ý rằng, số lượng yêu cầu hệ thống liên quan tới phạm vi dự án.

Khả năng của nhóm phát triển: Điều này bao gồm kinh nghiệm, việc đào tạo và môi trường làm việc cũng như chính sách động viên, khuyến khích đối với những người làm việc trong nhóm.

Quy trình phát triển phần mềm được áp dụng: Có rất nhiều quy trình phát triển phần mềm, và khó có thể nói rằng quy trình này tốt hơn quy trình kia, nhưng có điều là nếu một sản phẩm đầu ra có chất lượng tốt có nghĩa là nó đã được áp dụng một quy trình tốt.

2.1.2.4. Thời gian

Thời gian thực hiện dự án cũng có ảnh hưởng tới chi phí dành cho dự án và chất lượng của sản phẩm. Thời gian quá nhiều hay quá ít đều không phù hợp. Nếu thời gian ít sẽ làm giảm chất lượng và số lượng các chức năng cung cấp, nhưng nếu thời gian quá nhiều sẽ làm tăng chi phí, đồng thời có thể tạo điều kiện cho người phát triển thêm vào những chức năng không cần thiết.

Quan hệ giữa phạm vi, thời gian, chất lượng và chi phí không những áp dụng cho XP mà còn có thể áp dụng trong nhiều trường hợp. Tuy nhiên, việc xác định các đại lượng này vẫn còn chưa được thống nhất. Có những đại lượng có thể xác định giá trị là những con số cụ thể như thời gian, chi phí.

Nhưng cũng có các đại lượng chỉ có thể xác định một cách định tính, như chất lượng phần mềm, hay chỉ có thể đánh giá một cách tương đối dựa trên số các yêu cầu chức năng như phạm vi dự án.

2.1.3. Các giá trị của XP

Khái niệm giá trị ở đây là muốn nói đến những gì được coi trọng của một ai đó hay một cái gì đó. Beck nêu bật bốn giá trị chung nhất và được coi như là nền tảng của Extreme Programming. Các giá trị đó là sự liên lạc, tính đơn giản, sự phản hồi và sự can đảm.

2.1.3.1. Sự liên lạc

Liên lạc rất quan trọng trong việc tạo ra mối quan hệ giữa người với người. Đó có thể là quan hệ cá nhân hay quan hệ công việc. Trong XP, sự liên lạc giữa các thành viên được chú trọng đặc biệt. Để một dự án thành công, các thành viên trong nhóm phải thường xuyên trao đổi với nhau. Có thể thấy rõ, mặc dù nhóm gồm một số người riêng lẻ nhưng công việc của từng thành viên phụ thuộc và ảnh hưởng lẫn nhau. Và để công việc có thể diễn ra trôi chảy, thì từng thành viên cần phải giữ liên lạc với các thành viên còn lại trong nhóm, để có thể thông báo kịp thời các vấn đề của mình và nắm được thông tin của nhóm. Một số kỹ thuật được sử dụng để tăng cường sự liên lạc là lập trình theo cặp, khách hàng cùng phát triển, tạo điều kiện làm việc tốt và báo cáo thường xuyên.

2.1.3.2. Tính đơn giản

XP đưa ra quan điểm: “Đâu là cái đơn giản nhất có thể mà có thể làm việc được”. Quan điểm trên thể hiện tính đơn giản của một giải pháp nào đó. Theo đó, ta nên chọn giải pháp đơn giản nhất có thể để giải quyết vấn đề hiện thời, hơn là đưa ra các giải pháp phức tạp cho những vấn đề mà *có thể có* sau

này. Trong phát triển phần mềm, điều này có nghĩa là chỉ nên nghĩ đến những vấn đề đang phải giải quyết, chứ không phải là các yêu cầu có thể có sau này của hệ thống. Khẩu hiệu hành động mà XP đưa ra là: “Thiết kế cho hôm nay”.

Tính đơn giản và sự liên lạc có quan hệ qua lại với nhau. Liên lạc càng nhiều thì chúng ta càng rõ hơn về những yêu cầu phải thực hiện hay không cần thực hiện của hệ thống. Ngược lại, giải pháp càng đơn giản thì càng ít phải liên lạc và việc liên lạc sẽ rõ ràng hơn, nhanh hơn.

2.1.3.3. Sự phản hồi

Trạng thái hiện thời của hệ thống cung cấp những thông tin phản hồi tốt nhất cho nhóm phát triển. Việc cung cấp phản hồi trong thời gian ngắn nhất có thể sẽ giúp ích cho việc ra quyết định, để đảm bảo dự án thực hiện đúng hướng. Cần phải phản hồi lại càng sớm càng tốt kết quả của những chức năng được cài đặt, nhất là trong hoàn cảnh mọi thứ đều có thể thay đổi.

Sự phản hồi phụ thuộc trực tiếp vào các giá trị còn lại và đồng thời ảnh hưởng đến các giá trị còn lại. Phản hồi là một phần của việc liên lạc và làm tăng sự can đảm.

2.1.3.4. Sự can đảm

Điều này liên quan tới việc phải quyết định một việc khó khăn để đảm bảo dự án được thực hiện đúng hướng. Cần phải có can đảm bỏ và thực hiện lại các chức năng trong trường hợp không đúng, can đảm nhận xét đối với công việc của mình cũng như của những thành viên trong nhóm. Sự e ngại thường dẫn đến việc các thành viên không bày tỏ quan điểm cũng như liên lạc với các thành viên khác, hoặc cố giấu đi sai sót vì sợ trách nhiệm. Những thứ đó đều có thể gây ra những thiệt hại và làm giảm tính hiệp đồng giữa các thành viên.

2.1.4. Các nguyên tắc

Bốn giá trị của XP đã đưa ra những điều kiện quan trọng để thu được thành công. Tuy nhiên các giá trị này còn chung chung. Các nguyên tắc của XP được đưa ra dựa trên bốn giá trị và cụ thể hoá các giá trị này và đóng vai trò định hướng cho những hướng dẫn thực hiện của XP. Phần này sẽ đề cập tới những nguyên tắc của XP, bao gồm các nguyên tắc cơ bản và các nguyên tắc phụ khác.

2.1.4.1. Các nguyên tắc cơ bản.

Phản hồi nhanh – Nhanh chóng nhận phản hồi, xử lý và áp dụng vào sẽ làm tăng tính thích nghi với thay đổi và sửa lỗi nhanh hơn. Xử lý phản hồi càng chậm thì việc sửa lỗi hoặc đáp ứng thay đổi càng tốn kém và chứa đựng nhiều rủi ro.

Giải pháp đơn giản – Coi mọi vấn đề như thể rằng có thể giải quyết nó một cách đơn giản nhất. Trên thực tế, hầu hết các vấn đề đều có thể giải quyết được một cách đơn giản. Đôi khi người ta hay phức tạp hoá vấn đề, mà không nghĩ rằng có thể giải quyết nó bằng một giải pháp đơn giản hơn.

Thay đổi dần dần – Cố gắng giữ cho mọi thứ thay đổi một cách từ từ, tránh sự thay đổi lớn tại một thời điểm. Một thay đổi lớn có thể thực hiện qua một chuỗi những sự thay đổi nhỏ.

Đối mặt với sự thay đổi – Chấp nhận sự thay đổi là một trong những ưu điểm của XP so với các phương pháp phát triển phần mềm khác. Trong khi các phương pháp truyền thống cố gắng tránh thay đổi các đặc tả đã được thiết kế, thì XP cho phép khách hàng thường xuyên cập nhật những thay đổi yêu cầu hệ thống, với mục đích thoả mãn tối đa yêu cầu của khách hàng.

Chất lượng công việc – Mọi người đều muốn một công việc tốt, và không ai muốn làm việc một cách cẩu thả. Nếu người làm việc yêu thích công việc của mình thì sẽ đạt hiệu quả cao, ngược lại có thể làm giảm chất lượng của sản phẩm làm ra. Điều này có nghĩa là mọi thành viên phải được tạo điều kiện phát huy khả năng của mình một cách tốt nhất.

2.1.4.2. Các nguyên tắc phụ

Ngoài các nguyên tắc cơ bản trên, XP còn đưa ra thêm một số nguyên tắc khác ít quan trọng hơn.

Hướng dẫn cách học - Nguyên lý này cho rằng nên dạy những người cách học thay vì dạy kiến thức theo một khuôn mẫu. Có thể đưa ra các mức học khác nhau, ví dụ như mức đầu là chỉ ra những kỹ thuật cơ bản, những bài mẫu, trong khi mức cao hơn chỉ đưa ra những ý tưởng một cách trừu tượng, còn việc cụ thể hoá ý tưởng là do người học tự thực hiện.

Đầu tư từng bước – Đầu tư một lượng vừa đủ để dự án có thể bắt đầu, và đầu tư từng bước theo tiến độ thực hiện của dự án. Việc đầu tư này áp dụng cho cả tài nguyên là con người và tài chính. Đối với tài nguyên con người, điều này có nghĩa là không nên chuẩn bị nhân lực cho toàn bộ dự án, mà thay vào đó chỉ nên bắt đầu với một số người tốt thiểu và sẽ tăng dần trong quá trình thực hiện dự án. Đối với việc đầu tư tài chính, nhà đầu tư không nên đầu tư toàn bộ số tiền thực hiện dự án tại thời điểm bắt đầu dự án, mà chỉ nên cung cấp một lượng nhỏ vừa đủ để dự án bắt đầu, và sẽ đầu tư từng bước mỗi khi một tính năng mới được hoàn thiện.

Thực hiện để giành chiến thắng – Luôn giữ vững mục tiêu của mọi công việc là để giành thắng lợi. Điều này có nghĩa nhóm phát triển làm mọi

thứ giúp cho họ giành được thắng lợi và không làm bất cứ thứ gì không giúp cho họ thắng lợi.

Kiểm nghiệm cẩn thận – Đối với mỗi quyết định được đưa ra, cần phải được kiểm nghiệm một cách cẩn thận và đầy đủ để có thể chắc chắn rằng quyết định là đúng đắn. Việc ra một quyết định mà không được kiểm nghiệm hoàn toàn có thể là một quyết định sai, và càng ra nhiều quyết định thì độ rủi ro càng cao. XP thực hiện việc kiểm nghiệm bằng cách trao đổi và kiểm thử theo chức năng.

Giao tiếp cởi mở và thành thật – Đối tác cần liên lạc với nhóm phát triển, vì thế mọi người cần có khả năng nói lên suy nghĩ của mình một cách cởi mở, không bị ức chế. Thậm chí, đối với những tin xấu, cũng cần phải thông báo một cách đầy đủ với khách hàng.

Hiểu những mong muốn tự nhiên của con người – Mọi người đều muốn chiến thắng, muốn học, muốn giao tiếp với người khác, muốn làm một công việc tốt, muốn làm được những phần mềm hữu ích. Đây là những mong muốn tự nhiên của mỗi người, và cần được tôn trọng và phát huy, không nên chống lại nó.

Tự đảm nhận trách nhiệm – Không nên giao trách nhiệm cho một ai đó mà phải tự người đó thấy được trách nhiệm của mình. Mỗi thành viên trong nhóm đều cần có trách nhiệm, và khi có một việc cần phải làm thì phải có người đứng lên chấp nhận làm công việc đó, dù cho nó có thể nào đi chăng nữa.

Áp dụng theo hoàn cảnh cụ thể – XP không phải là một quy trình cứng nhắc phải tuân theo, mà nó chỉ mang tính chất chỉ đạo, hướng dẫn. Mỗi

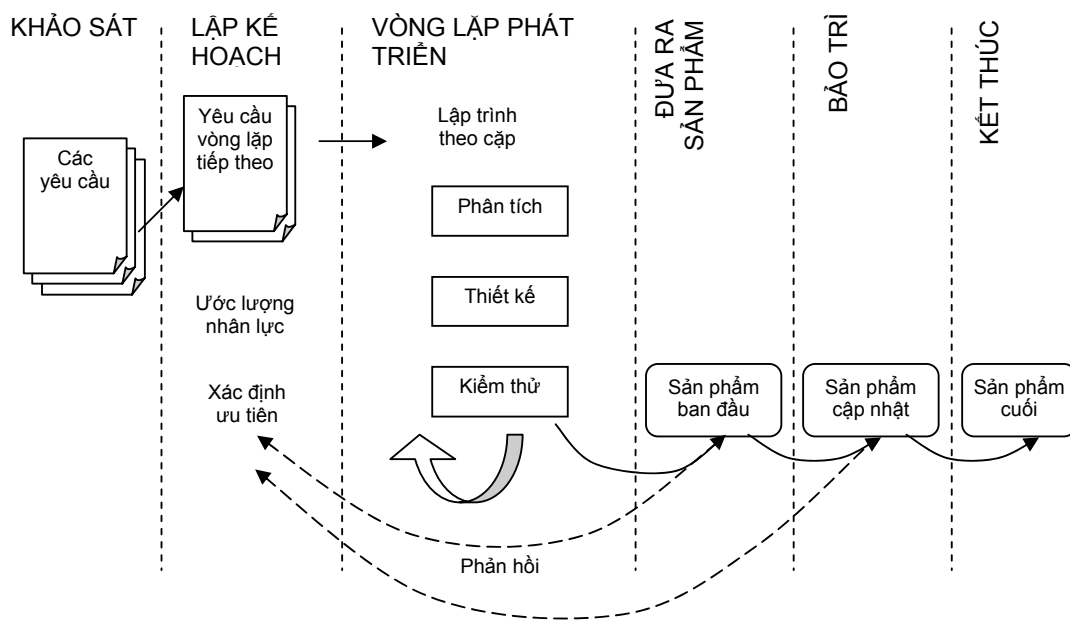
nhóm cần lựa chọn cho mình một cách áp dụng phù hợp tùy theo hoàn cảnh cụ thể. Tuy nhiên việc áp dụng không thể tùy tiện, mà nên thực hiện theo khung mà được chỉ ra bởi XP.

Trang bị gọn nhẹ – Để có thể dễ dàng thay đổi, nhóm cần phải được trang bị gọn nhẹ. Những người này phải có khả năng linh động cao, nhanh chóng chuyển hướng theo yêu cầu của khách hàng hoặc hoàn cảnh.

Lựa chọn tiêu chuẩn đánh giá tốt – Cần phải sử dụng những tiêu chuẩn đánh giá đúng đắn và chi tiết sao cho có lợi. Ví dụ như việc lấy số dòng mã làm tiêu chuẩn đánh giá sản lượng và chất lượng, thì sẽ không hữu dụng đối với các ngôn ngữ lập trình hiện đại, nhất là sau khi quá trình làm mịn và tối ưu được thực hiện.

2.1.5. Quy trình XP

Vòng đời của quy trình XP gồm các giai đoạn: khảo sát, lập kế hoạch, vòng lặp phát triển, đưa ra sản phẩm, bảo trì và kết thúc dự án.



Hình 2.1 - Quy trình XP

2.1.5.1. Giai đoạn khảo sát

Trước khi bắt đầu việc phát triển, nhóm phát triển cần đánh giá và phải đảm bảo năng lực thực hiện dự án, bao gồm những yếu tố như nhân lực, kỹ năng, công nghệ, thời gian.

Trong giai đoạn này, các khách hàng viết ra các yêu cầu mà họ muốn có trong phiên bản đầu tiên. Mỗi yêu cầu này tương ứng với một chức năng của chương trình. Tuy nhiên việc này không phải lúc nào cũng diễn ra suôn sẻ. Việc chậm trễ thường xuyên xảy ra do khách hàng có thể biết những gì mà những người lập trình cần, nhưng khó biết được những gì mà người lập trình không cần.

Song song với đó, các thành viên dự án làm quen với các công cụ, công nghệ và cách họ sẽ làm việc trong dự án. Cần xây dựng một mô hình nguyên

mẫu cho hệ thống nhằm kiểm tra công nghệ được sử dụng và khảo sát các kiến trúc có thể của hệ thống. Giai đoạn khảo sát kéo dài từ vài tuần đến vài tháng, phụ thuộc nhiều vào mức độ quen thuộc công nghệ của các lập trình viên.

2.1.5.2. Giai đoạn lập kế hoạch

Mục đích của giai đoạn lập kế hoạch là cho phép khách hàng và những người phát triển thoả thuận một ngày đưa ra những chức năng quan trọng nhất.

Công việc phải thực hiện là thiết đặt mức độ ưu tiên cho các yêu cầu và thống nhất nội dung của phiên bản đầu tiên. Đầu tiên, các lập trình viên sẽ ước lượng công sức cần để giải quyết các yêu cầu, sau đó thống nhất lịch trình làm việc. Thời gian cho lịch trình của phiên bản đầu tiên trong khoảng từ hai đến sáu tháng. Việc lập kế hoạch kéo dài một vài ngày.

2.1.5.3. Các vòng lặp phát triển

Lịch trình được thiết lập trong giai đoạn lập kế hoạch được chia nhỏ thành một vài vòng thời gian kéo dài từ một đến bốn tuần. Ở vòng lặp đầu tiên, cần tạo ra một hệ thống có kiến trúc của hệ thống tổng thể. Việc này được thực hiện bằng cách chọn các nhiệm vụ mà buộc phải xây dựng kiến trúc cho hệ thống tổng thể.

Tại mỗi vòng lặp khách hàng quyết định nhiệm vụ cho mỗi vòng lặp, và ở cuối mỗi vòng lặp, các kết quả được đưa ra và được tiến hành kiểm thử chức năng.

Kết thúc vòng lặp cuối, hệ thống sẵn sàng chuyển sang giai đoạn đưa ra sản phẩm đầu tiên.

2.1.5.4. Giai đoạn đưa ra sản phẩm

Ở giai đoạn này, các sản phẩm được kiểm thử song song, và có thể vẫn có những thay đổi. Những thay đổi này được ghi nhận và áp dụng cho phiên bản hiện thời hoặc phiên bản kế tiếp.

Ngoài ra, trong giai đoạn này cần phải tiến hành cải tiến hiệu năng, tối ưu hoá chương trình.

Và công việc chính đó là chuyển giao sản phẩm cho khách hàng và bắt đầu đưa vào vận hành.

2.1.5.5. Bảo trì

Sau khi phiên bản đầu tiên được chuyển giao cho khách hàng sử dụng, dự án XP phải đồng thời duy trì hoạt động của sản phẩm và bắt đầu những vòng lặp mới. Để làm điều này, giai đoạn bảo trì đòi hỏi công sức cho việc hỗ trợ khách hàng. Do đó, tốc độ phát triển có thể chậm lại. Giai đoạn bảo trì có thể yêu cầu phải kết nạp thêm các thành viên mới vào đội dự án và thay đổi cấu trúc đội.

2.1.5.6. Kết thúc

Khi khách hàng không còn nhiệm vụ nào cần thực hiện nữa. Các yêu cầu mà hệ thống phục vụ khách hàng cần thoả mãn trên các phương diện như tính năng, độ tin cậy... Trong giai đoạn này, cần hoàn thiện các tài liệu cần thiết về hệ thống khi không có thêm sự thay đổi về kiến trúc, thiết kế hay mã nguồn.

Ngoài ra, cũng có thể tiến hành kết thúc khi hệ thống không đưa ra được đầu ra mong muốn, hay sẽ rất tốn kém nếu phát triển tiếp.

2.1.6. Hướng dẫn thực hiện

Để có thể áp dụng được tốt hơn, XP đưa ra một loạt các hướng dẫn trong việc thực hiện quy trình XP nói riêng và phát triển phần mềm nói chung. XP hướng tới khả năng phát triển phần mềm thành công dù các yêu cầu có thể thay đổi. Để đảm bảo khả năng linh động, nhóm phát triển không nên lớn (theo Beck, nên gồm từ 3 đến 20 thành viên).

Những hướng dẫn thực hiện của XP bao gồm:

2.1.6.1. Phương án lập kế hoạch

Nhanh chóng xác định phạm vi của phiên bản sắp đưa ra. Ngoài ra, cũng có thể phải điều chỉnh kế hoạch cho phù hợp với thực tại.

Việc lập kế hoạch do cả khách hàng và những người phát triển cùng thực hiện. Cần phải có sự cân bằng giữa mong muốn của khách hàng và khả năng của những người thực hiện.

Khách hàng cần phải quyết định về phạm vi chức năng của sản phẩm sẽ được tạo ra, về độ ưu tiên của các chức năng và ngày phát hành sản phẩm. Trong khi đó những người phát triển phải ước lượng được thời gian thực hiện, lựa chọn quy trình và lập lịch thực hiện dựa trên các quyết định của khách hàng.

2.1.6.2. Phương án phát hành

Tại một thời điểm, nên lập kế hoạch trong vòng một đến hai tháng hơn là sáu tháng hay một năm. Nhanh chóng phát hành một phiên bản nhỏ với những yêu cầu quan trọng nhất và đưa vào sử dụng. Sau đó, liên tục phát hành các phiên bản tiếp theo với chu kì ngắn, thậm chí hàng ngày.

2.1.6.3. Tổng thể hệ thống

Hướng dẫn cho toàn bộ những người phát triển biết một cách chung nhất về hệ thống cần làm. Điều này sẽ giúp những người tham gia hiểu được hệ thống gồm những thành phần gì và liên hệ giữa chúng như thế nào.

XP đưa ra khái niệm *metaphor* với ý nghĩa là một cái nhìn tổng thể về hệ thống. Khái niệm này gần giống với khái niệm kiến trúc tổng thể, tuy nhiên metaphor được mô tả dễ hiểu hơn, dễ sử dụng để trao đổi với nhau và với khách hàng.

2.1.6.4. *Thiết kế đơn giản*

Thiết kế đơn giản nhất có thể, khả thi tại thời điểm hiện tại. Loại bỏ tất cả những phần phức tạp, không cần thiết.

2.1.6.5. *Kiểm thử*

Tất cả các chức năng của chương trình cần phải được kiểm thử. Những người lập trình sẽ xây dựng các bộ dữ liệu kiểm tra để có thể kiểm tra tính tin cậy của các chức năng chương trình. Trong khi đó, khách hàng tiến hành kiểm tra các chức năng để đảm bảo chương trình hoạt động như mong muốn của họ.

2.1.6.6. *Phân tích lại*

Quá trình phân tích lại là quá trình cải tiến chương trình, làm chương trình đơn giản hơn, nhỏ gọn hơn, thực hiện nhanh hơn nhưng vẫn đảm bảo đầy đủ các tính năng và thoả mãn tất cả các kiểm thử.

Thường thì khi thực hiện một tính năng nào đó, người lập trình thường cố gắng sao cho chức năng hoạt động được. Điều này có thể dẫn đến những dư thừa, lặp lại, hoặc thuật toán chưa được tối ưu. Vì thế cần phải tiến hành phân tích lại để tối ưu nhất có thể.

2.1.6.7. *Lập trình theo cặp*

Ý tưởng của XP là hai lập trình viên cùng thực hiện viết mã trên một máy. Một người trực tiếp thực hiện với máy và nghĩ cách cài đặt tốt nhất các chức năng, trong khi người kia thì nghĩ xem rằng cách làm đó có thực sự đúng không, có bài thử nào mà cách cài đặt này không làm việc không, có cách tiếp cận nào đơn giản hơn nhưng vẫn đảm bảo chức năng tương đương...

Việc cặp đôi này cần linh động, một người có thể cặp đôi với người này trong thời gian này về một lĩnh vực nào đó, sau đó có thể ghép với người khác để cùng làm công việc khác.

2.1.6.8. *Sở hữu tập thể*

Mã nguồn là sở hữu của tập thể, và mọi người đều có cơ hội được tiếp cận. Mô hình này trái ngược với hai mô hình sở hữu mã là không sở hữu (mã không thuộc về ai, mọi người đều có thể tự ý sử dụng và thay đổi) và sở hữu cá nhân.

Tuy nhiên, theo mô hình này mọi người đều có thể thay đổi mã theo ý của mình, hoặc để phù hợp với mục đích của mình. Kết quả là lượng mã sẽ tăng lên nhanh chóng, có nhiều phiên bản và có thể trở thành một mớ hỗn độn.

Để khắc phục, trong XP, mọi người đều phải có trách nhiệm về toàn bộ hệ thống. Nếu một cặp làm việc thấy rằng cần thiết phải cải tiến mã, thì họ có cơ hội tiến hành cải tiến mã.

2.1.6.9. *Tích hợp liên tục*

Mã được tích hợp vào hệ thống và được kiểm tra chỉ sau vài giờ. Khi mã nguồn được tích hợp vào, cần kiểm tra và giải quyết những xung đột và tiến hành kiểm thử sao cho vượt qua tất cả các bài kiểm thử.

2.1.6.10. Tuần làm 40 giờ

Không làm quá 40 giờ một tuần, điều này sẽ tạo cho người làm việc giảm căng thẳng và có tính sáng tạo.

Việc phải làm thêm giờ cho thấy có vấn đề trong dự án. Theo XP, không làm việc thêm giờ hai tuần liên. Ai đó có thể làm thêm giờ một tuần, nhưng nếu tuần tiếp theo mà vẫn phải làm thêm giờ tức là công việc của anh ta có vấn đề mà không thể giải quyết bằng việc tăng thời gian.

2.1.6.11. Khách hàng cùng làm việc

Cần có một khách hàng cùng làm việc với nhóm phát triển. Khách hàng trợ giúp cho nhóm phát triển những thắc mắc tính chuyên môn nghiệp vụ, đánh giá các chức năng dưới góc độ người sử dụng thực sự.

Khách hàng được lựa chọn phải là người làm công việc bình thường, không quá khác biệt so với những người khách hàng khác. Ngoài ra, họ còn cần biết nhiều lĩnh vực trong công việc để có thể trợ giúp tốt hơn.

2.1.6.12. Chuẩn viết mã

Cần phải đưa ra các chuẩn, các quy tắc viết mã. Điều này sẽ giúp cho việc quản lý mã tốt hơn, tăng khả năng trao đổi mã. Các chuẩn viết mã cần được thiết kế thống nhất, và được các thành viên tuân thủ một cách tự nguyện.

2.1.7. Nhận xét

XP đưa ra một cách chi tiết các hướng dẫn thực hiện, được mô tả cách rõ ràng từ cách thực hiện đến những lợi ích thu được từ những hoạt động này.

Việc thực hiện theo các hướng dẫn này có thể giải quyết được khá nhiều những vấn đề thường gặp phải trong các dự án hiện nay và tăng thêm khả năng thành công của dự án.

Tuy nhiên, có một số hướng dẫn mà việc thực hiện tốt không phải là dễ. Ví dụ như việc yêu cầu có một khách hàng thường trực, làm việc cùng đội phát triển phụ thuộc nhiều vào khách hàng. Thường thì khách hàng có rất nhiều việc phải làm hiện thời, và không phải ai cũng có tinh thần hợp tác tốt. Hay việc lập trình theo cặp là khá xa lạ đối với chúng ta hiện nay. Không dễ gì để một người ngồi bên cạnh một người khác với đúng chức năng như được đưa ra trong XP. Ngoài ra, người trực tiếp làm việc với máy cũng phải được chuẩn bị tinh thần để làm quen với việc có người ngồi bên cạnh theo dõi công việc của mình.

Nhưng trên hết, XP đã đưa ra những ý tưởng, phương pháp hay và khả thi. Thực tế cho thấy XP đã thành công, vì thế việc áp dụng XP trong phát triển phần mềm sẽ có lợi rất nhiều.

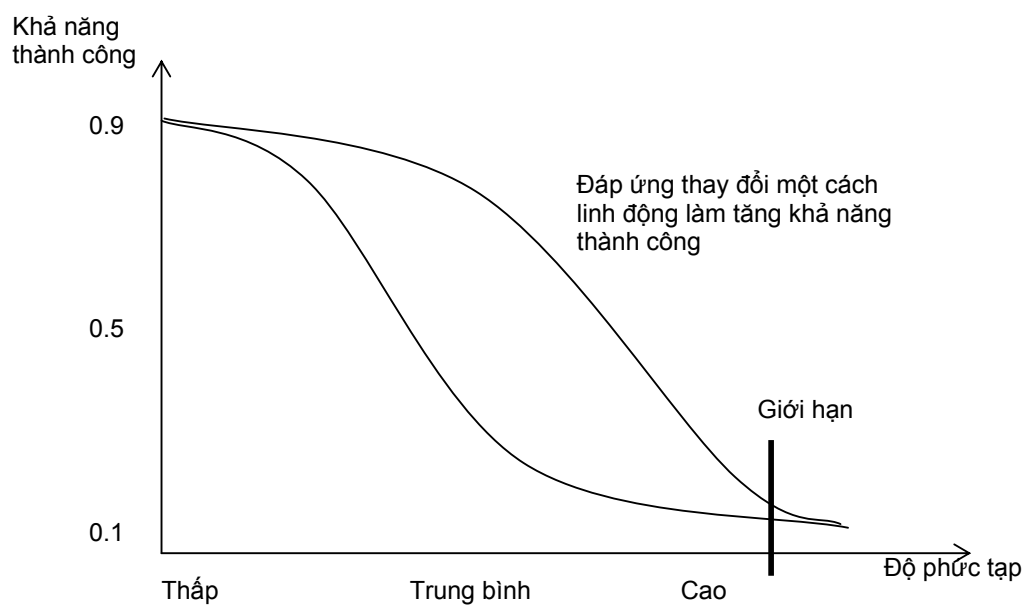
2.2. Scrum

2.2.1. Giới thiệu

Scrum là phương pháp luận được phát triển bởi Ken Schwaber và Mike Beedle. Thuật ngữ Scrum được đưa ra dựa trên một bài viết của Takeuchi và Nonaka (1986) mà trong đó có giới thiệu một quy trình phát triển phần mềm nhanh, có khả năng thích nghi [5].

Scrum được biết đến như là một phương pháp quản lý nâng cao, áp dụng cho các hệ thống hiện có. Do đó, có thể áp dụng Scrum với các phương pháp phát triển phần mềm khác.

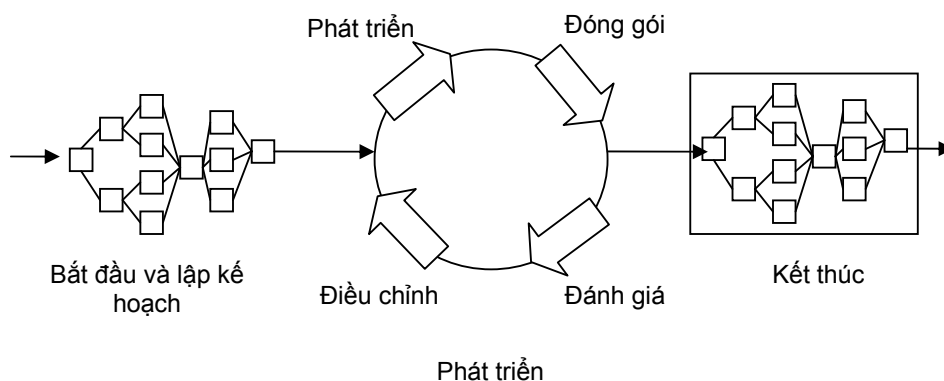
Ý tưởng chính của Scrum là cho rằng việc phát triển một hệ thống cần phải quản lý một loạt các đại lượng như các yêu cầu, thời gian, tài nguyên hay công nghệ dùng để phát triển, mà những đại lượng này hoàn toàn có thể thay đổi trong quá trình phát triển. Từ đó cho thấy quá trình phát triển dự án mang tính không ổn định, phức tạp, khó dự đoán trước. Do đó cần thiết phải có một quy trình phát triển có tính linh hoạt cao để có thể đáp ứng được những thay đổi này, và sản phẩm đầu ra phải có tính ứng dụng cao, đáp ứng được yêu cầu khách hàng.



Hình 2.2 - Tỷ lệ thành công tăng khi đáp ứng tốt những thay đổi.

2.2.2. Quy trình

Scrum là một phương pháp hướng quy trình. Quy trình Scrum chia thành ba giai đoạn, giai đoạn bắt đầu và lập kế hoạch, giai đoạn phát triển, và giai đoạn kết thúc và đưa ra sản phẩm.



Hình 2.3 - Quy trình Scrum

2.2.2.1. Giai đoạn bắt đầu và lập kế hoạch

Trước khi dự án bắt đầu, tất cả các yêu cầu hệ thống được liệt kê và tập hợp dưới dạng các phiếu công việc, được gọi là các Backlog. Tập hợp các phiếu công việc của toàn bộ sản phẩm được gọi là Product Backlog. Product Backlog cho ta cái nhìn tổng thể về các chức năng của sản phẩm cuối cùng. Các yêu cầu này có thể thu được từ khách hàng, từ bộ phận bán hàng hay từ những người phát triển phần mềm. Người tạo ra Product Backlog được gọi là Product Owner (người sở hữu), thông thường là khách hàng, hoặc là người quản lý trong công ty.

Tiếp theo, các yêu cầu này được xác định độ ưu tiên và ước lượng nhân lực cần thiết để cài đặt các tính năng yêu cầu. Danh sách này liên tục được cập nhật thêm các mục mới cũng như được xác định lại độ ưu tiên cho các công việc và ước lượng nhân lực, tài nguyên sao cho chính xác hơn.

Trong giai đoạn này còn phải đưa ra được nhóm dự án, các công cụ và tài nguyên cần thiết, đánh giá rủi ro và tiến hành đào tạo nếu thấy cần thiết. Ngoài ra, thiết kế tổng thể cũng phải được định nghĩa trong giai đoạn này.

Trước khi tiến hành phát triển, người sở hữu chọn những công việc cần tiến hành trong vòng lặp đầu tiên của giai đoạn phát triển. Các công việc này được tập hợp dưới một danh sách gọi là Sprint Backlog. Trong khi đó, nhóm phát triển chuẩn bị những gì cần thiết và ước lượng thời gian sao cho công việc có thể hoàn thành trong vòng 30 ngày, là khoảng thời gian một vòng lặp được quy định bởi Scrum. Việc thống nhất kế hoạch giữa người sở hữu và nhóm phát triển được tiến hành trong một cuộc họp.

Công việc cuối cùng là xác định mục tiêu phải hoàn thành trong vòng lặp, gọi là Sprint Goal. Việc xác định mục tiêu này để cho đội phát triển thấy được lý do của những công việc mình phải làm.

2.2.2.2. Giai đoạn phát triển

Toàn bộ giai đoạn phát triển được phân thành các vòng lặp, mỗi vòng lặp kéo dài 30 ngày, gọi là Sprint.

Trong mỗi vòng lặp, các thành viên của dự án chọn các công việc từ Sprint Backlog, làm việc sao cho đạt được mục tiêu đề ra trong Sprint Goal. Trong vòng lặp, các công việc lại được chia thành các khoảng thời gian nhỏ hơn:

- **Phát triển** – Tiến hành phân tích, thiết kế, cài đặt, kiểm thử và viết tài liệu cho những chức năng được lựa chọn.
- **Đóng gói** – Tạo bộ cài đặt của sản phẩm đến thời điểm hiện thời.
- **Xem lại** – Tất cả các thành viên trong nhóm họp với nhau để cùng xem xét lại để đưa ra cách giải quyết những vấn đề gặp phải.
- **Điều chỉnh** – Tổng hợp các thông tin thu được từ cuộc họp và tiến hành điều chỉnh.

Trong giai đoạn này, các thành viên gặp nhau trong cuộc họp hàng ngày. Cuộc họp này nên kéo dài trong khoảng 15 phút. Trong cuộc họp, các thành viên trong nhóm phải đưa ra được:

- Những gì đã làm được từ lần họp trước.
- Dự định làm gì cho tới lần họp tiếp theo.
- Có gặp vướng mắc gì không.

Việc tiến hành họp hàng ngày sẽ giúp cho việc nắm bắt rõ hiện trạng công việc, đồng thời tăng cường việc liên hệ giữa các thành viên trong nhóm.

Để giảm tác động của việc thay đổi, Scrum đưa ra nguyên tắc là mọi thay đổi trong một Sprint được ghi nhận nhưng không được áp dụng ngay. Điều này có nghĩa là trong một vòng lặp, các công việc chỉ ra trong Sprint Backlog được cố định. Mặc dù điều này có vẻ không phù hợp với tiêu chí đáp ứng thay đổi nhanh của các phương pháp phát triển nhanh, nhưng là cần thiết vì mọi thứ thay đổi liên tục, nếu thay đổi ngay lập tức theo yêu cầu có thể dẫn đến tình trạng lộn xộn.

Cuối mỗi vòng lặp, các kết quả được thể hiện cho người sở hữu, người quản lý và những ai quan tâm. Dựa vào đó, người sở hữu phải chuẩn bị để đưa ra những tính năng cần thiết phải cài đặt trong vòng lặp kế tiếp.

Nếu toàn bộ các chức năng đã hoàn thành, thì dự án bước sang giai đoạn cuối là đưa ra sản phẩm.

2.2.2.3. Giai đoạn kết thúc và đưa ra sản phẩm

Khi sản phẩm đã sẵn sàng được phát hành, người quản lý sẽ tuyên bố đóng dự án và tiến hành việc đưa ra sản phẩm. Trong giai đoạn này, một số công việc khác cần phải được chuẩn bị tiến hành như tập hợp tài liệu sử dụng, đào tạo người dùng, phát triển kinh doanh.

2.2.3. Nhóm dự án Scrum

Đối với một dự án, ngoài những người phát triển trực tiếp còn có những đối tác bên ngoài cũng tham gia, như nhân viên bán hàng, marketing, hay khách hàng. Thường thì những nhóm bên ngoài này thường được tách ra để

tránh làm phức tạp, nhưng với Scrum, thành phần nhóm dự án được đánh giá đầy đủ hơn.

Trong một dự án, những nhóm chính sau được tạo ra:

- **Những người quản lý:** đứng đầu bởi người quản lý sản phẩm, những người này định nghĩa nội dung phát hành cũng như thời gian phát hành của sản phẩm, đồng thời quản lý tiến trình thực hiện dự án.
- **Các nhóm phát triển:** Đội ngũ phát triển được chia thành các nhóm phát triển nhỏ, mỗi nhóm có thể gồm những người phát triển, người lập tài liệu, nhân viên kiểm định chất lượng. Công việc chính của nhóm là xác định yêu cầu dựa vào các backlog và cài đặt và tích hợp kết quả. Thành viên của nhóm được chọn dựa vào kiến thức và sự thành thạo của từng người về từng lĩnh vực.

2.2.4. Một số nét đặc trưng của Scrum

Scrum là một trong những phương pháp quản lý tiêu biểu trong lớp các phương pháp phát triển nhanh với đặc thù là tính linh động cao, thời gian đáp ứng thay đổi nhanh và cộng tác chặt chẽ giữa các thành viên cũng như với khách hàng. Có thể liệt kê ra đây một số đặc trưng của các dự án Scrum:

- Linh động trong việc đưa ra kết quả - nội dung đưa ra phụ thuộc vào môi trường.
- Linh động trong lập lịch – việc đưa ra có thể sớm hơn hoặc muộn hơn kế hoạch.
- Các đội phát triển nhỏ - đội ngũ phát triển nhỏ được chia thành các nhóm nhỏ, một dự án có thể có nhiều nhóm phát triển.

- Thảo luận lại thường xuyên – quá trình thực hiện của từng nhóm thường xuyên được đưa ra xem xét, thảo luận.
- Tính cộng tác – trong quá trình thực hiện dự án, việc cộng tác giữa các thành viên với nhau và với đối tác được coi trọng.

2.2.5. Một số ưu điểm của Scrum

Các phương pháp phát triển truyền thống, toàn bộ những yêu cầu được xác định trước trong giai đoạn khảo sát, và các phương pháp này cố gắng hạn chế sự thay đổi các đặc tả. Chính điều này làm giảm khả năng đáp ứng thay đổi của các phương pháp đó, tăng khả năng rủi ro cũng đồng nghĩa với việc giảm khả năng thành công của dự án, nhất là với các dự án kéo dài.

Ngược lại, Scrum được thiết kế khá linh động. Phương pháp này cung cấp cơ chế có khả năng điều khiển cho việc lập kế hoạch cho việc đưa ra sản phẩm và thực hiện dự án. Điều này cho phép thay đổi dự án và mục tiêu đưa ra tại bất cứ thời điểm nào để có thể đưa ra sản phẩm phù hợp nhất.

Trong suốt quá trình thực hiện dự án, Scrum cho phép những nhà phát triển phát huy những sáng kiến của mình để đưa ra những giải pháp tốt nhất. Ngoài ra, cơ chế chia đội phát triển thành các nhóm nhỏ, có tính cộng tác cao giúp cho việc chia sẻ kinh nghiệm, kiến thức dễ dàng hơn.

2.2.6. Nhận xét

Scrum là đưa ra một quy trình khá mới lạ, trong đó kết quả công việc đề cao. Phương pháp này tập trung trong quản lý nhiều hơn là đưa ra các hướng dẫn để phát triển phần mềm. Điều đó có nghĩa là Scrum có thể sử dụng kết hợp với các phương pháp khác, ví dụ như Extreme Programming.

Tuy nhiên việc áp dụng cơ chế theo dõi quá trình thực hiện dự án thông qua các cuộc họp hàng ngày là khó thực hiện, thậm chí ngay cả đối với dự án nhỏ. Ngoài ra, để việc áp dụng Scrum được hiệu quả thì toàn bộ đội ngũ phát triển cũng như quản lý phải cùng phối hợp thực hiện.

2.3. Phương pháp phát triển phần mềm thích nghi

2.3.1. Giới thiệu

Phương pháp phát triển phần mềm thích nghi (Adaptive Software Development – ASD) được phát triển bởi James A. Highsmith, là một trong những phương pháp thuộc lớp các phương pháp phát triển nhanh. Phương pháp này tập trung chủ yếu trong việc giải quyết các vấn đề xuất hiện trong các hệ thống phức tạp, nhiều thay đổi.

Tư tưởng chủ đạo của ASD cho rằng, trong môi trường liên tục có những thay đổi bất thường thì việc thích nghi là rất quan trọng. Xuất phát từ quan điểm đó, ASD đưa ra một quy trình mang tính lặp lại, quá trình “học” được tiến hành qua mỗi vòng lặp để tăng cường khả năng thích nghi.

2.3.2. Quy trình

Một dự án phát triển theo ASD được tiến thành thông qua các vòng lặp, các vòng lặp gồm ba giai đoạn: suy đoán, cộng tác và học.

Tên của các giai đoạn được đặt như vậy cho thấy một số đặc trưng của phương pháp này. Từ *suy đoán* (từ gốc trong tiếng Anh là “Speculate”) được sử dụng thay cho từ lập kế hoạch (Plan) là bởi vì việc lập kế hoạch thường dựa trên những gì đã rõ ràng, chắc chắn, trong khi mọi thứ đều có thể thay đổi. Từ *cộng tác* (Collaborate) được sử dụng cho quá trình phát triển nhằm nhấn mạnh vai trò của việc cộng tác giữa các thành viên, và từ *học* (Learn)

Công việc đầu tiên cần thực hiện đó là định nghĩa nhiệm vụ của dự án. Nhiệm vụ này đưa ra một khung cơ bản cho sản phẩm cuối cùng, giúp định hướng cho quá trình phát triển sản phẩm. Phạm vi dự án cũng phải được ước lượng, đồng thời đội ngũ phát triển cũng phải được định hình cơ bản. Mục tiêu của dự án là hoàn thành nhiệm vụ này, mặc dù tại thời điểm bắt đầu, những yêu cầu có thể còn rất mơ hồ, nhưng sẽ rõ ràng hơn trong quá trình thực hiện dự án.

Công việc tiếp theo cần thực hiện là phải xác định thời gian thực hiện cho toàn bộ dự án. Dựa vào khoảng thời gian này, khung thời gian cho mỗi vòng lặp được định nghĩa.

Các vòng lặp được lập kế hoạch, độ dài của một vòng lặp phụ thuộc vào kích cỡ dự án và mức độ khả thi, nhưng thường trong khoảng từ 2 đến 8 tuần. Việc lập kế hoạch này bao gồm cả việc gán các khung thời gian cho mỗi vòng lặp.

Tiếp theo là việc lên lịch trình thực hiện các vòng lặp. Trong bước này, mỗi vòng lặp được gán với một “chủ đề”, là vấn đề chính cần được chú ý khi thực hiện vòng lặp.

Cuối cùng, các tính năng được gán cho các vòng lặp. Các khách hàng là người quyết định dựa trên mức độ ưu tiên của từng tính năng. Những người phát triển sẽ hỗ trợ cho khách hàng bằng việc ước lượng thời gian, rủi ro và các đại lượng khác.

2.3.2.2. Giai đoạn phát triển các tính năng

Trong giai đoạn này, các đặc tính của sản phẩm được phát triển. Giống như Scrum, ASD không đưa ra các hướng dẫn chi tiết trong việc làm thế nào để phát triển mà chỉ đơn thuần đưa ra một bộ khung phục vụ cho việc quản lý.

Điều mà ASD muốn nhấn mạnh, đó chính là việc hợp tác giữa các thành viên. Theo Highsmith, việc hợp tác đặc biệt quan trọng, ông đưa ra quan điểm rằng một cá nhân riêng lẻ không thể có toàn bộ khả năng cần thiết cho một dự án [10]. Những người quản lý phải có nhiệm vụ tạo ra một môi trường cộng tác tốt, đồng thời các thành viên trong nhóm phải tăng cường trao đổi và hỗ trợ nhau trong công việc.

2.3.2.3. Đánh giá lại chất lượng công việc

Mọi người không ai là biết tất cả mọi thứ, và do đó cần phải học. Kiến thức thu được sau khi suy ngẫm về một cái gì đó, rồi tiến hành thử nghiệm và đánh giá kết quả.

Trong giai đoạn này, những tính năng được phát triển trong giai đoạn trước được trình bày với khách hàng. Những người phát triển học được rất nhiều từ những đánh giá dưới góc độ khách hàng hoặc người sử dụng. Nhiều khi, về mặt tính năng thì những chức năng đã cài đặt là thỏa mãn, nhưng về mặt sử dụng, thuật ngữ, bố trí có thể cần phải cải tiến để thuận tiện hơn.

Ngoài ra, các thành viên trong nhóm cũng cần phải thảo luận về những vấn đề kỹ thuật gặp phải, từ đó rút ra được những bài học để tránh mắc phải hoặc biết cách xử lý trong trường hợp tương tự. ASD cho rằng, không thể có quan điểm luôn phải đúng, mà cho rằng ai cũng có thể mắc lỗi, và điều quan trọng là phải học được những kinh nghiệm từ những lỗi đó.

Để đánh giá lại chất lượng công việc, cần phải xem xét lại những tính năng đã làm có hoạt động không, và nó hoạt động như thế nào, công việc của nhóm có vướng mắc gì không.

Cuối cùng, cần đánh giá hiện trạng của dự án, những gì đã làm được và chưa làm được và có kế hoạch cho những công việc tiếp theo.

2.3.3. Nhận xét

Chúng ta thấy, ASD là phương pháp mà tập trung chủ yếu vào kết quả và chất lượng công việc hơn là chỉ ra những quy trình, thao tác cần phải thực hiện để đạt được kết quả. Quá trình thực hiện dự án thông qua một loạt các vòng lặp, trong đó việc lập kế hoạch cũng là một phần trong vòng lặp. Điều này có nghĩa là các chức năng cần phát triển sẽ liên tục được điều chỉnh, làm mịn dựa vào các thông tin mới thu được qua mỗi vòng lặp.

Mặc dù ASD không đưa ra nhiều hướng dẫn thực hiện, mà chỉ đưa ra một khung quản lý gọn nhẹ, nhưng chứa đựng nhiều ý tưởng quan trọng trong việc thích nghi với những thay đổi thường xuyên. Điều này cho phép những nhà quản lý có thể tùy biến cách áp dụng phương pháp này.

Cũng giống như Scrum, ASD không định nghĩa cụ thể cách thức thực cài đặt các chức năng, do đó có thể áp dụng kết hợp với một phương pháp phát triển khác như XP.

2.4. Đánh giá và so sánh các phương pháp

Trong phần này, chúng ta sẽ đánh giá và so sánh các phương pháp được đề cập trong phần 2.3. Việc đánh so sánh các phương pháp là tương đối khó khăn, không thể nói đơn giản là phương pháp này hay hơn phương pháp kia,

mà kết quả còn phụ thuộc vào các tiêu chí đánh giá cũng như quan điểm của người đánh giá.

Việc so sánh này không có ý định chỉ ra cái nào tốt hơn, có giá trị hơn cái nào, mà mục đích so sánh là chỉ ra những điểm giống nhau, khác nhau giữa các phương pháp, để từ đó có một cái nhìn tổng quát về các phương pháp này.

Có nhiều kỹ thuật để so sánh, và phương pháp so sánh được sử dụng ở đây là chọn lọc ra những đặt trưng quan trọng nhất từ một vài phương pháp và dựa vào đó để so sánh với nhau.

2.4.1. Những đặc điểm chính

Trong phần này, những đặc điểm chính của các phương pháp được đánh giá và so sánh. Những đặc điểm này được trích dẫn từ chính nội dung của các phương pháp, không phụ thuộc vào việc áp dụng những phương pháp này như thế nào, đó là: vai trò khách hàng, chiến lược phát hành, khả năng đáp ứng thay đổi và các hướng dẫn cụ thể của từng phương pháp.

Trong tất cả các phương pháp đã được giới thiệu, vai trò của khách hàng luôn được đề cao. Khách hàng đóng vai trò định hướng cho dự án. Trong XP, Scrum và ASD, khách hàng là người lựa chọn những chức năng cần được thực hiện qua mỗi giai đoạn cũng như xác định độ ưu tiên của các chức năng đó. Đặc biệt, trong XP, khách hàng còn tham gia vào đội phát triển dự án với tư cách là một thành viên thường trực.

Về chiến lược phát hành, các phương pháp đều đưa ra kết quả sau một thời gian ngắn. Scrum trình bày kết quả sau mỗi Sprint 30 ngày, đối với ASD là khoảng từ 2 đến 8 tuần. Trong khi đó, chiến lược của XP là đưa ra nhanh

chóng một phiên bản nhỏ và đưa vào sử dụng, sau đó liên tục phát hành các phiên bản mới hơn, thậm chí hàng ngày.

Về khả năng đáp ứng thay đổi, các phương pháp đều cho rằng việc thay đổi yêu cầu là tất yếu, khó tránh khỏi và việc đáp ứng thay đổi sẽ làm giảm rủi ro và tăng khả năng thành công của dự án. Một trong những nguyên tắc của XP là “Embracing change” [2], với nghĩa là cần phải đổi mới, chấp nhận sự thay đổi cho thấy XP chú trọng vào việc đáp ứng những thay đổi. Đối với Scrum, để đảm bảo tính ổn định, trong thời gian 30 ngày của một Sprint, các yêu cầu được cố định, những thay đổi được ghi nhận, không được xử lý ngay. Còn với ASD, việc đáp ứng thay đổi và tiếp nhận ý kiến khách hàng còn được coi là một quá trình học hỏi.

Tiêu chí cuối cùng được đưa ra đánh giá trong mục này, là những hướng dẫn thực tiễn mà các phương pháp cung cấp. ASD không cung cấp nhiều hướng dẫn cụ thể về quản lý cũng như việc làm thế nào để phát triển phần mềm, mà phương pháp này chỉ cung cấp một khuôn mẫu chung, tổng quát để những nhà quản lý áp dụng. Scrum thì đưa ra nhiều hướng dẫn khá chi tiết và khắt khe trong quản lý, như họp mặt hàng ngày, khoảng thời gian phát triển 30 ngày, nhưng lại không đưa ra nhiều các kỹ thuật lập trình. Ngược lại, XP đưa ra các hướng dẫn thực hiện trong lập trình khá chi tiết, nhưng lại đề cập ít trong lĩnh vực quản lý.

2.4.2. Khả năng và phạm vi áp dụng

Trong phần này, chúng ta sẽ đánh giá khả năng áp dụng, phạm vi áp dụng cũng như việc áp dụng các phương pháp.

Một điều hiển nhiên là một phương pháp phát triển phần mềm chỉ thực sự có ý nghĩa khi nó được sử dụng trong quá trình tạo ra sản phẩm. Và để có

thể áp dụng được, thì điều cần thiết là các phương pháp này phải đủ đơn giản để có thể áp dụng được ngay, vì các công ty hay tổ chức sẽ không đủ khả năng cho việc dừng công việc hiện thời để học hay tổ chức lại. Vì thế, để đánh giá được khả năng áp dụng thì điều một điều quan trọng cần phải được đề cập đó chính là tính dễ áp dụng.

Trước tiên là ASD, phương pháp này đưa ra một mô hình khá chung, mang tính chất định hướng, nên việc áp dụng ASD khá linh động, tính tùy biến cao. Tuy nhiên, phương pháp này không đưa ra nhiều hướng dẫn cụ thể nên đòi hỏi những người quản lý phải có kỹ năng quản lý tốt.

Scrum đưa ra những tiêu chuẩn, hướng dẫn đủ chi tiết để có thể áp dụng được ngay, mặc dù có thể là không dễ. Tuy nhiên, phương pháp này thiếu những hướng dẫn cụ thể về mặt kỹ thuật lập trình, nên có thể áp dụng kết hợp với một phương pháp phát triển phần mềm khác.

Và cuối cùng là XP, như chúng ta đã thấy XP đưa ra nhiều ý tưởng khá hay và hiệu quả, như khách hàng cùng làm việc hay lập trình theo cặp. XP không cứng nhắc việc áp dụng, mà chỉ mang tính chất chỉ đạo, nên việc áp dụng XP dễ hơn. Chúng ta có thể áp dụng XP đầy đủ, nhưng cũng có thể sử dụng những gợi ý của phương pháp này và kết hợp với các phương pháp khác để có thể thu được hiệu quả cao nhất.

Từ đó dẫn đến ý tưởng kết hợp Scrum và XP để có thể tận dụng những ưu điểm của hai phương pháp này. Chương 3 sẽ đưa ra một mô hình áp dụng kết hợp Scrum và XP dựa trên những điều kiện thực tế áp dụng.

CHƯƠNG 3 - PHÁT TRIỂN PHẦN MỀM ÁP DỤNG SCRUM VÀ EXTREME PROGRAMMING

Trong chương 2, chúng ta đã tìm hiểu một số phương pháp quản lý và phát triển phần mềm thuộc lớp các phương pháp phát triển nhanh. Mỗi phương pháp đều có những ưu điểm riêng, và phù hợp trong các điều kiện khác nhau. Để có thể tận dụng thế mạnh của các phương pháp, cần phải xác định những điểm mạnh của từng phương pháp và xem xét khả năng áp dụng. Từ đó dẫn đến ý tưởng kết hợp giữa Scrum và XP và đưa ra một mô hình áp dụng phù hợp, trong đó kết hợp các điểm mạnh của các phương pháp.

Việc áp dụng được tiến hành ở cả trong lĩnh vực quản lý và trong kỹ thuật lập trình. Trong quản lý những điểm mạnh của Scrum được sử dụng, và trong lập trình, một số kỹ thuật quan trọng của XP sẽ được đưa vào sử dụng. Ngoài ra, một số biện pháp khác nhằm tăng cường chất lượng phần mềm, giảm lỗi cũng được đề xuất áp dụng.

Mô hình đưa ra áp dụng tốt nhất cho các dự án với phạm vi vừa, thời gian phát triển ngắn, yêu cầu khách hàng thường xuyên thay đổi. Ngoài ra, do tính linh hoạt của các phương pháp, nên việc áp dụng phù hợp với những công ty có số lượng thành viên không lớn, tổ chức gọn nhẹ linh hoạt.

Phần này sẽ đưa ra một cách chi tiết mô hình áp dụng các phương pháp Scrum và Extreme Programming trong phát triển phần mềm.

3.1. Quy trình phát triển phần mềm

Quy trình phát triển phần mềm đưa ra dựa trên quy trình của Scrum, trong đó sẽ cụ thể hoá một số giai đoạn, công việc.

3.1.1. Xác định mục tiêu dự án

Mục tiêu dự án là công việc đầu tiên cần phải thực hiện. Việc xác định chi tiết các mục tiêu của dự án là một việc tương đối khó khăn, vì thế khi dự án bắt đầu, chỉ những mục tiêu chung nhất, mang tính định hướng được xác định.

Việc xác định mục tiêu dự án là cần thiết, dựa vào đó chúng ta có thể xác định được phạm vi của dự án, cũng như định hướng cho việc khảo sát, lấy yêu cầu.

Mục tiêu này được đề xuất, sau đó thảo luận và đồng ý giữa khách hàng và đội phát triển.

3.1.2. Khảo sát và lấy yêu cầu khách hàng

Khảo sát và lấy yêu cầu của khách hàng là một trong những công việc cần thực hiện trong giai đoạn bắt đầu dự án. Vì quy trình, nghiệp vụ và các công việc của khách hàng rất nhiều và phức tạp, nên cần phải được định hướng khi khảo sát và lấy yêu cầu khách hàng dựa vào mục tiêu của dự án được đề ra.

Thực tế cho thấy, khách hàng mặc dù rất thông thạo trong công việc, nắm rõ quy trình nghiệp vụ, nhưng họ thường không định hình được việc sản phẩm phần mềm sẽ thực hiện những gì, giúp đỡ được họ như thế nào. Vì vậy, khó có hi vọng rằng khách hàng sẽ cung cấp được một cách có hệ thống các yêu cầu ngay lập tức.

Từ đó cho thấy, việc khảo sát và lấy yêu cầu khách hàng cần phải được thực hiện qua nhiều bước và phải được chuẩn bị kỹ. Các bước để thực hiện việc lấy yêu cầu được đề xuất như sau:

Bước 1: Khảo sát – Tìm hiểu về đối tác, những lĩnh vực hoạt động của đối tác, gặp gỡ với những người có thẩm quyền, để từ đó có một cái nhìn chung về những công việc của đối tác. Việc này có thể thực hiện bằng cách nghiên cứu các tài liệu, thảo luận hoặc quan sát trực tiếp. Sau bước này, cần chỉ ra các mảng cần phải lấy yêu cầu, cũng như cần phải làm việc với những ai để cụ thể hoá yêu cầu.

Bước 2: Xác định yêu cầu chung – Sau khi khảo sát, cần tiến hành gặp gỡ những người có chuyên môn, nghiệp vụ phù hợp. Tuy nhiên tại bước này, cả phía khách hàng và người lấy yêu cầu vẫn chưa thực sự hiểu nhau. Khách hàng không rõ họ sẽ phải trình bày những gì, trong khi đó người lấy yêu cầu vẫn chưa hiểu nhiều về công việc của khách hàng. Vì thế, ở bước này người lấy yêu cầu nên đưa ra các câu hỏi chung, mang tính gợi mở như:

- Nhiệm vụ của anh/chị là gì?
- Với nhiệm vụ đó thì anh/chị phải thực hiện những công việc gì?
- Thực hiện các công việc đó theo những bước nào?

Sau khi có được những thông tin bước đầu về công việc của khách hàng, cần phải phân tích sơ bộ những thông tin này, và xác định đâu là những phần sẽ được đưa vào hệ thống, đâu là những thông tin không cần thiết.

Bước 3: Chi tiết hoá các yêu cầu – Các thông tin ban đầu sau khi được phân tích sơ bộ sẽ được tập hợp lại, phân thành các đầu mục nhỏ hơn. Mỗi một đầu mục cần lấy yêu cầu được viết trên một phiếu dưới dạng câu hỏi. Các phiếu này sẽ được đánh mã để dễ quản lý. Thông tin được hỏi trong các câu hỏi này cần chi tiết, hẹp, cụ thể, và thường có dạng như:

- Công việc / bước ... của công việc XYZ được thực hiện như thế nào?
- Việc thực hiện công việc XYZ cần tuân thủ những quy định, văn bản nào?
- Đại lượng ABC được tính theo công thức như thế nào?
- Kết quả đưa ra của công việc XYZ được thể hiện dưới dạng nào, có mẫu không?
- Trong trường hợp ... thì phải giải quyết như thế nào?
- Với việc này, anh/chị có đề xuất, mong muốn gì?

Những câu hỏi này sẽ giúp cho khách hàng dễ dàng cung cấp thông tin, đồng thời các thông tin thu được sẽ có hệ thống hơn.

Các phiếu được chuyển cho khách hàng, và trong khi đội dự án chuẩn bị những gì cần thiết thì khách hàng sẽ tiến hành viết những câu trả lời và những yêu cầu, đề xuất.

Việc này có thể phải được tiến hành nhiều lần để có thể thu được một tập hợp các phiếu yêu cầu đầy đủ và rõ ràng.

3.1.3. Phân tích yêu cầu

Các phiếu yêu cầu được tập hợp, phân loại, sau đó các thông tin trong các phiếu đó được phân tích để từ đó đưa ra được các chức năng một cách chi tiết. Mỗi chức năng này sẽ được ghi trên một phiếu công việc (đóng vai trò như Backlog trong Scrum), trong đó cần chỉ rõ những thông tin như tên chức năng, mô tả chức năng, đầu vào, đầu ra và hoạt động của chức năng, những yêu cầu phi chức năng khác đối với chức năng đó.

Tập các phiếu này sẽ được duyệt lại giữa đối tác khách hàng và người quản lý dự án. Sau khi các phiếu công việc được duyệt, phía khách hàng sẽ xác định độ ưu tiên của các chức năng, và chọn ra những công việc cần phải thực hiện ở vòng lặp đầu tiên. Trong khi đó, đội phát triển sẽ trợ giúp khách hàng trong việc lựa chọn các chức năng dựa vào năng lực hiện thời, đồng thời ước lượng nhân lực, vật lực, công nghệ.

3.1.4. Cài đặt các chức năng

Việc cài đặt các chức năng sẽ diễn ra thông qua một loạt các vòng lặp, mỗi vòng lặp có thể kéo dài một vài tuần tùy theo tính chất của công việc, điều kiện của khách hàng.

Trong mỗi vòng lặp, các chức năng được chọn sẽ được cài đặt, tích hợp, thử nghiệm, thiết kế lại để sao cho kết thúc vòng lặp, các chức năng có thể trình bày được.

Việc họp đánh giá được tiến hành thường xuyên, ít nhất một tuần một lần, tốt nhất là hàng ngày và kéo dài không quá 15 phút. Nội dung cuộc họp thực hiện theo các hướng dẫn của Scrum, đã được giới thiệu trong 2.2.2.2.

Ngoài ra, trong giai đoạn này các yêu cầu mới, hoặc thay đổi cũng được tiếp nhận và phân tích và được đưa vào tập hợp những chức năng chưa được lựa chọn.

3.1.5. Trình bày kết quả

Khi kết thúc mỗi vòng lặp, các kết quả được trình bày cho phía khách hàng, người quản lý, người phát triển và những người quan tâm. Việc trình bày phải chỉ ra tối đa những gì đã làm được và chưa làm được.

Dựa vào những gì đã được trình bày, phía khách hàng sẽ quyết định những chức năng nào hoàn thành tốt, những chức năng nào cần cải tiến, và thậm chí những chức năng nào cần loại bỏ. Tiếp theo là phải đưa ra được những chức năng sẽ được cài đặt trong vòng lặp kế tiếp.

Thông qua buổi trình bày, người phát triển cần rút ra được những kinh nghiệm, tiếp thu những ý kiến đánh giá của khách hàng.

3.1.6. Đưa ra các sản phẩm thử nghiệm

Sau một hoặc một số vòng lặp, những chức năng cơ bản nhất, quan trọng nhất đã được thực hiện. Nếu kết quả thu được đủ khả năng hoạt động được, sản phẩm thử nghiệm sẽ được đưa ra và chuyển giao cho khách hàng.

Sản phẩm bước đầu này sẽ được sử dụng song song với quá trình phát triển các chức năng khác, và được cập nhật thường xuyên qua mỗi vòng lặp.

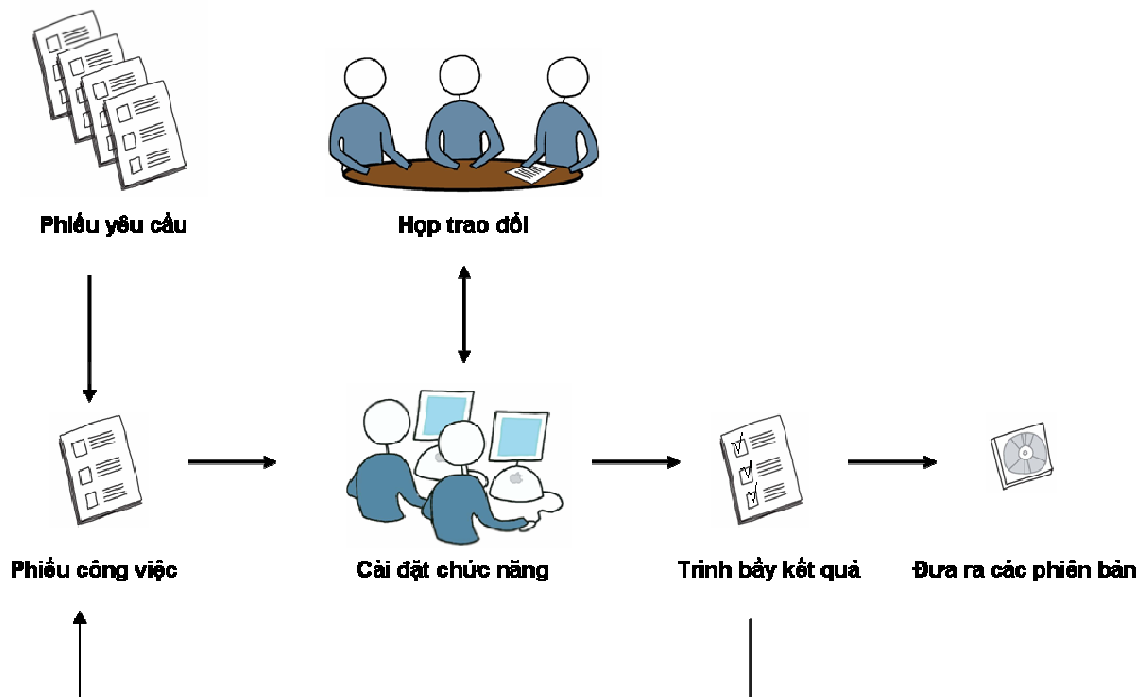
Trong quá trình sử dụng sản phẩm thử nghiệm, các ý kiến đóng góp của khách hàng, các lỗi phát sinh sẽ được tiếp nhận, phân tích và đưa vào tập các phiếu công việc chờ thực hiện.

3.1.7. Kết thúc

Khi tất cả các chức năng đã được thực hiện, sản phẩm cuối được chuyển giao cho khách hàng thì tiến hành kết thúc dự án. Trong giai đoạn này, một số công việc cần phải thực hiện như:

- Tiến hành chuyển giao sản phẩm, tài liệu liên quan, đào tạo người sử dụng.
- Hoàn thành các thủ tục pháp lý.
- Tổng kết, đánh giá kết quả công việc, rút kinh nghiệm.

Trong trường hợp sản phẩm là phần mềm đóng gói, được phát hành rộng rãi thì những công việc như vận hành, hỗ trợ kỹ thuật, giới thiệu sản phẩm và bán hàng, sẽ được chuyển giao cho những nhóm chuyên trách khác. Đồng thời, kế hoạch về các phiên bản tiếp theo cũng được chuẩn bị.



Hình 3.1 – Quy trình phát triển phần mềm đề xuất

3.2. Một số biện pháp tăng cường trong quản lý

Trong phần này sẽ đưa ra một số biện pháp nhằm nâng cao khả năng quản lý dự án phần mềm.

3.2.1. Làm việc tập trung

Cần cung cấp cho người phát triển một khoảng thời gian làm việc liên tục mà không bị ngắt quãng. Khái niệm về khoảng thời gian làm việc tập trung được gọi tên là Jam Session [10].

Người ta chỉ có thể làm việc hiệu quả khi tập trung vào công việc, và để tập trung vào công việc cần mất một khoảng thời gian. Thực tế cho thấy, để tập trung vào công việc người ta phải mất khoảng 15 phút hoặc nhiều hơn, điều đó có nghĩa nếu bị ngắt quãng 5 phút cho công việc khác, thì thời gian làm việc sẽ mất tới 20 phút. Do đó, cần giảm tối đa việc ngắt quãng trong quá trình làm việc.

Có rất nhiều nguyên nhân khiến người phát triển bị ngắt quãng. Có thể kể ra đây một số nguyên nhân như việc gọi, nhận điện thoại, chat, hay việc phải chuyển sang xử lý một vấn đề gì đó. Có nhiều công việc là do bắt buộc, hoặc cũng có nguyên nhân có thể hạn chế được, do đó việc tập trung làm việc có thực hiện được tốt hay không phụ thuộc nhiều vào ý thức của từng người và những quy định cụ thể của tổ chức.

Ngoài ra, để tăng khả năng tập trung, một người không nên làm song song nhiều công việc cùng một lúc. Điều này nghe có vẻ đơn giản, nhưng thực tế rất khó thực hiện. Đôi khi, những người phát triển đang thực hiện cài đặt một chức năng, thì phải chuyển sang sửa một chức năng khác, hay thậm chí phải chuyển sang khắc phục lỗi của một dự án khác do điều kiện nhân lực thiếu. Do đó, người quản lý cần phải có kế hoạch cụ thể, sắp lịch phù hợp cho những công việc cần phải làm để giảm tối đa việc chuyển đổi công việc.

Làm việc tập trung cần thiết cho bất cứ công việc gì, vì vậy biện pháp này có thể áp dụng cho tất cả các giai đoạn của quy trình.

3.2.2. Giảm chu kỳ phát hành

Giảm chu kỳ phát hành có nghĩa là nhanh chóng đưa ra được sản phẩm với những chức năng quan trọng nhất, cơ bản nhất, và liên tục cập nhật các phiên bản mới cho khách hàng trong khoảng thời gian ngắn nhất.

Nếu một phần mềm mà chỉ chuyển giao cho khách hàng khi hầu hết các chức năng đã hoàn thành thì thường chất lượng rất thấp, đồng thời sẽ rất tốn kém nếu phải thay đổi.

Mặt khác, nếu sản phẩm nhanh chóng được chuyển giao cho khách hàng sẽ làm cho khách hàng hiểu được về phần mềm, từ đó có thể đưa ra những đóng góp hay những yêu cầu mới chính xác hơn. Thêm vào đó, những gì không phù hợp sẽ nhanh chóng được nhận ra và khắc phục, điều này có nghĩa là khả năng đáp ứng thay đổi sẽ tăng lên.

Hầu hết các phương pháp phát triển nhanh đều đưa ra các chu kỳ phát hành ngắn, như XP là khoảng từ 2 đến 4 tuần hay Scrum là 30 ngày. Các phiên bản cập nhật được chuyển giao cho khách hàng ở cuối mỗi vòng lặp.

Biện pháp này áp dụng trong giai đoạn cài đặt các chức năng và đưa ra các kết quả.

3.2.3. Thảo luận hàng ngày

Các cuộc họp được tiến hành hàng ngày, trong đó các thành viên trình bày về những công việc mới hoàn thành, dự định công việc sẽ thực hiện sắp tới, nêu ra những khó khăn vướng mắc hoặc đề xuất những ý kiến cải tiến.

Những kết quả thu được qua buổi họp sẽ được đánh dấu vào lịch trình thực hiện dự án, để từ đó có thể đánh giá được chính xác tiến độ thực hiện dự án. Ngoài ra, qua cuộc họp, các thành viên cũng có thể nắm rõ được tiến trình thực hiện các chức năng của các thành viên khác, để từ đó có những điều chỉnh phù hợp, tránh việc phải đợi chờ nhau do sự phụ thuộc của các công việc.

Cuộc họp có thể được thực hiện theo từng nhóm nếu công việc mang tính độc lập, hoặc họp chung nếu công việc của các nhóm phụ thuộc nhau. Người quản lý có thể tham gia cuộc họp nhưng chủ yếu là lắng nghe những người phát triển trình bày. Để tránh mất thời gian, các cuộc họp nên kéo dài không quá 15 phút, và được thực hiện vào cuối buổi làm việc.

Việc thảo luận hàng ngày có thể áp dụng cho hầu hết các giai đoạn của quy trình, nhưng tập trung chủ yếu trong giai đoạn cài đặt các chức năng.

3.2.4. Khách hàng cùng tham gia phát triển

Đây là một trong các biện pháp thực hiện được đề xuất bởi XP, trong đó khách hàng tham gia phát triển cùng với các thành viên của dự án. Nhiệm vụ của khách hàng là hỗ trợ người phát triển về mặt chuyên môn, nghiệp vụ hoặc những ý kiến đánh giá dưới góc độ người sử dụng.

Tuy nhiên, việc có một khách hàng thường trực tại công ty trong suốt thời gian phát triển dự án là điều tương đối khó khăn. Điều đó có nghĩa là phải có giải pháp để sao vừa đảm bảo tính cộng tác của khách hàng, vừa đảm bảo không làm ảnh hưởng nhiều đến công việc của khách hàng. Một số giải pháp có thể áp dụng như:

- **Liên lạc thường xuyên với khách hàng** – Bất cứ điều gì chưa rõ ràng, hoặc muốn lấy ý kiến khách hàng sẽ được gửi cho khách hàng thông qua thư điện tử hoặc gọi điện trực tiếp.
- **Phát triển tại nơi khách hàng làm việc** – Đối với một số công việc mang tính đặc thù cao, người phát triển có thể được tạo điều kiện để làm việc ngay tại nơi khách hàng làm việc. Điều này sẽ đảm bảo khách hàng luôn có sẵn mà không ảnh hưởng tới công việc của khách hàng.

- **Tạo điều kiện làm việc cho khách hàng** – Tạo điều kiện để khách hàng có thể làm những công việc của mình ngay tại nơi phát triển phần mềm. Tùy từng công việc cụ thể mà có thể áp dụng phương án này hay không.
- **Khách hàng cộng tác định kỳ** – Khách hàng sẽ tham gia một cách định kỳ một số ngày trong một tuần, ngoài thời gian đó khách hàng vẫn làm các công việc bình thường. Việc sắp lịch tùy từng điều kiện cụ thể, nhưng cần phải được đảm bảo đều đặn.

Như vậy, mặc dù đây là một biện pháp mang lại rất nhiều lợi ích nhưng việc thực hiện tương đối khó, cần phải có sự phối hợp tốt và đặc biệt cần phải được sự quan tâm tạo điều kiện của những người có thẩm quyền.

Giai đoạn lấy yêu cầu tất nhiên cần phải có sự tham gia của khách hàng. Khách hàng cùng phát triển là biện pháp chủ yếu tập trung vào giai đoạn cài đặt các chức năng và đưa ra các kết quả.

3.3. Một số biện pháp tăng cường trong phát triển phần mềm

Phần này sẽ đưa ra một số phương pháp nhằm tăng cường chất lượng công việc phát triển phần mềm. Những biện pháp này sẽ giúp tăng năng suất lao động, giảm thiểu lỗi và tăng chất lượng sản phẩm.

Hầu hết các biện pháp này xuất phát từ những hướng dẫn của XP đã được nêu ở phần 2.1.6. Nội dung phần này nhằm mục đích cụ thể hoá việc thực hiện các biện pháp.

3.3.1. Lập trình theo cặp

Lập trình theo cặp là một kỹ thuật trong đó hai người cùng làm việc trên cùng một máy tính. Các công việc bao gồm thiết kế, cài đặt, kiểm thử... Theo như hướng dẫn của XP, thì một người sẽ trực tiếp thiết kế, cài đặt mã, trong khi người còn lại nghĩ về chiến lược áp dụng, tích hợp những gì đang được tạo ra sao cho phù hợp nhất.

Kỹ thuật này đòi hỏi cả hai người đều phải tăng cường trao đổi với nhau, đồng thời phải có năng lực chuyên môn tốt, khả năng phân tích và tổng quát hoá.

Một số ưu điểm của lập trình theo cặp có thể kể ra như sau:

- Tăng cường trao đổi trong quá trình giải quyết vấn đề.
- Tăng cường sự tập trung trong công việc.
- Chất lượng của công việc được nâng cao vì thường xuyên được theo dõi, xem lại.

Mặc dù kỹ thuật này được đưa ra nhằm mục làm tăng chất lượng phần mềm cũng như giảm thời gian phát triển, nhưng kỹ thuật này tương đối xa lạ so với thực tế chúng ta hiện nay, vì thế nếu áp dụng một cách máy móc, rập khuôn sẽ dẫn đến việc gượng ép, không hiệu quả. Việc áp dụng phải theo từng bước, từ bước đầu làm quen cho tới khi mà việc lập trình theo cặp làm cho người lập trình cảm thấy hứng thú.

Mục này sẽ đề xuất một số giải pháp áp dụng dựa trên những điều kiện thực tế, đó là:

Người mới học hỏi người có kinh nghiệm, kỹ năng – Người có kinh nghiệm, kỹ năng làm công việc trên máy đồng thời hướng dẫn những kỹ thuật cho người mới, ít kinh nghiệm. Theo cách này người mới sẽ học được về các

kỹ thuật, các chuẩn và tiếp thu được những kinh nghiệm từ người có kinh nghiệm. Ngoài ra, người theo dõi có thể suy nghĩ và tạo ra những bộ dữ liệu kiểm thử sao cho phản ánh tốt nhất hoạt động của các chức năng đang được thực hiện.

Người có kinh nghiệm giúp đỡ cho người mới – Đây là cách áp dụng ngược với biện pháp trên. Người mới sẽ thực hiện công việc trên máy và người có kinh nghiệm, kỹ năng sẽ theo dõi, đóng góp ý kiến, giúp đỡ cho người mới.

Áp dụng tùy theo tính chất công việc – Áp dụng lập trình theo cặp đối với những công việc quan trọng, trong đó cần phải trao đổi nhiều. Điều này đặc biệt quan trọng khi ghép các tính năng với nhau, khi đó để có thể tích hợp tốt cần phải hai người cùng thực hiện.

Lập trình theo cặp có thể áp dụng trong bước phân tích, nhưng chủ yếu áp dụng trong giai đoạn cài đặt các chức năng.

3.3.2. Áp dụng các phương pháp kiểm thử

Chúng ta đều phải công nhận rằng việc kiểm thử rất quan trọng. Một chức năng được đưa ra chỉ có thể đảm bảo hoạt động khi nó được kiểm thử. Thông thường việc kiểm thử được giao cho người kiểm thử, và người kiểm thử sẽ kiểm thử các chức năng của sản phẩm đưa ra thông qua giao diện người dùng. Với cách này, người kiểm thử thường kiểm tra không kỹ, hoặc bị quá chú trọng vào một vài mảng chức năng mà bỏ qua các chức năng khác. Do đó, các lỗi tiềm ẩn có thể không được phát hiện ra.

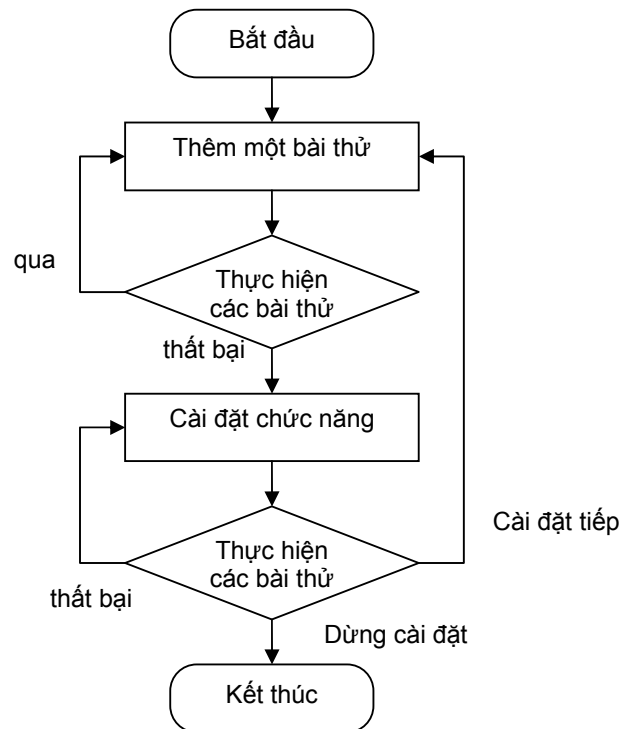
Như vậy, việc kiểm thử là một công việc tương đối tốn thời gian. Nhiều người phát triển cho rằng họ không phải là người có trách nhiệm thực hiện việc kiểm thử, và hầu hết đều ngại công việc này.

Để việc kiểm thử được hiệu quả, thì việc kiểm thử phải được thực hiện một cách có phương pháp. Đã có nhiều phương pháp được nghiên cứu, phần này sẽ đề cập một số cách tiếp cận phù hợp và được chấp nhận rộng rãi.

3.3.2.1. Tạo bộ kiểm thử trước khi cài đặt

Cách tiếp cận này được biết đến với cái tên Test Driven Development (TDD). Trước tiên là tạo các bài kiểm thử rồi mới cài đặt các chức năng. Điều này nghe vẻ ngược với những cách kiểm thử mà thường được sử dụng: các bài kiểm thử được viết dựa trên các chức năng đã được cài đặt.

Quá trình được thực hiện qua các bước sau [8]:



Hình 3.2 – Quy trình kiểm thử TDD

- **Bước 1: Viết bài thử**

Dựa vào yêu cầu hệ thống và dự định chức năng cần cài đặt, viết một bài thử cho chức năng sẽ cài đặt.

- **Bước 2: Chạy tất cả các bài thử**

Chạy thử các bài thử. Kết quả phải là thất bại vì chức năng chưa được cài đặt, nếu mọi bài kiểm thử đều qua có nghĩa là bài kiểm thử mới thêm vào không chính xác, hoặc phải sửa lại hoặc là đưa ra bài kiểm thử khác.

- **Bước 3: Cài đặt chức năng**

Cài đặt chức năng mà đã được viết bài kiểm thử, và chỉ chức năng đó mà thôi.

▪ **Bước 4: Chạy tất cả các bài thử**

Chạy lại tất cả các bài thử, nếu không qua nghĩa là chức năng mới cần phải được chỉnh sửa lại (quay lại bước 3). Nếu qua toàn bộ, thì nếu công việc hoàn thành thì kết thúc, ngược lại quay lại bước 1 để bắt đầu việc cài đặt một chức năng mới.

Với cách tiếp cận này, thì để có thể viết được một bài kiểm thử cho một chức năng thì chúng ta phải suy nghĩ về chức năng đó và phải xác định được cách sử dụng chức năng đó. Nếu không có bộ kiểm thử nào cho chức năng, thì có nghĩa là chức năng đó không cần thiết.

3.3.2.2. Kiểm thử đơn vị

Kiểm thử đơn vị là kiểm thử một đơn vị nhỏ nhất mà có thể kiểm thử được. Các đơn vị có thể là một phương thức, hay một chức năng nhỏ. Kiểm thử đơn vị kiểm tra xem đơn vị đó có thực hiện đúng như những gì đã thiết kế không. Kiểm thử đơn vị được áp dụng cho từng phương thức, các thành phần khác chỉ mang tính chất phụ trợ nếu cần thiết.

Kiểm thử đơn vị được thực hiện dựa trên các bộ kiểm thử. Bộ kiểm thử là một danh sách các bài kiểm thử, trong đó có các cột: chức năng cần kiểm thử, dữ liệu đưa vào hoặc thao tác, kết quả đầu ra mong đợi hoặc đáp ứng của hệ thống. Và người kiểm thử sẽ tiến hành thực hiện và ghi kết quả kiểm thử vào cột kết quả kiểm thử.

3.3.2.3. Kiểm thử chức năng

Kiểm thử chức năng là việc kiểm tra các chức năng lớn hơn, kiểm tra hoạt động của toàn hệ thống. Ở mức này, việc kiểm thử nhằm kiểm tra xem hệ thống hoạt động có đúng như yêu cầu, mong muốn của khách hàng hay

không. Việc quyết định kết quả có đạt hay không phải là sự đánh giá của khách hàng, người dùng cuối.

Để có thể đánh giá được chính xác, toàn diện, kiểm thử chức năng phải được thực hiện trong một khoảng thời gian dài, với các bộ dữ liệu thực tế và thực hiện trong điều kiện thực tế.

Các phương pháp kiểm thử đơn vị được áp dụng trong giai đoạn cài đặt các chức năng, mỗi khi một chức năng mới được tạo ra. Kiểm thử chức năng được áp dụng chủ yếu trong giai đoạn đưa ra các kết quả, vì khi đó về cơ bản hệ thống đã có thể hoạt động được và các chức năng đã có thể tương tác với nhau.

3.3.3. Thiết kế đơn giản

Đây là một trong những khuyến cáo được đề xuất bởi XP dựa trên những giá trị của phương pháp này, là tính đơn giản. Thiết kế đơn giản có nghĩa là chọn phương án đơn giản nhất có thể thực hiện được, chỉ thiết kế những gì hiện thời đang cần, không phải là những gì có thể có sau này.

Nhưng thế nào thì được gọi là đơn giản nhất, Kent Beck đưa ra một số tiêu chí đánh giá như sau [2]:

- Hệ thống phải truyền tải được tất cả những gì bạn muốn truyền tải.
- Hệ thống không được chứa mã bị lặp lại.
- Hệ thống phải có ít nhất có thể các lớp.
- Hệ thống phải có ít nhất có thể các phương thức.

Kent Beck cũng đưa ra một cách thực hành, đó là xoá đi tất cả những thứ mà không có mục đích nào cả, tất nhiên vẫn phải đảm bảo vượt qua toàn bộ các bài kiểm thử. Khi đó những gì còn lại sẽ là một thiết kế đơn giản nhất có thể thực hiện được.

3.3.4. Tích hợp liên tục

Đây cũng là một trong những hướng dẫn thực hiện được chỉ ra bởi XP. Tích hợp liên tục có nghĩa đưa những gì vừa làm được vào trở thành một phần của toàn bộ hệ thống ngay sau khi tạo ra, kiểm thử.

- Việc này có thể thực hiện theo các bước sau:
- Sao lưu toàn bộ dự án hiện thời.
- Tích hợp những gì vừa hoàn thành.
- Giải quyết tất cả những xung đột nếu có.
- Chạy tất cả các bài kiểm thử để đảm bảo mọi thứ vẫn hoạt động như mong đợi.

Việc tích hợp liên tục có lợi ích là những chức năng được tích hợp một cách nhanh nhất vào hệ thống, điều này sẽ hạn chế tình trạng phải chờ đợi do hiện tượng phụ thuộc nhau giữa các chức năng.

Ngoài ra, việc tích hợp liên tục còn có một ưu điểm nữa, đó là giúp cho người phát triển tập trung vào một chức năng duy nhất, là chức năng hiện đang được phát triển. Một khi chức năng đã hoàn thành, được tích hợp và kiểm thử đầy đủ, người phát triển sẽ không còn phải bận tâm đến nó nữa.

3.3.5. Đưa ra các chuẩn trong lập trình

Chuẩn hoá viết mã là rất cần thiết. Viết mã theo chuẩn giúp cho việc trao đổi mã dễ hơn. Ngay cả đối với người viết ra mã, thì chuẩn viết mã sẽ

giúp cho họ theo dõi mã nguồn, kiểm tra lại mã nguồn nhanh hơn và chính xác hơn.

Thực tế cho thấy, nếu không đưa ra các chuẩn thì mỗi người có thể viết theo một cách mà theo họ là tốt nhất, và khi đó việc tiếp nhận và hiểu mã nguồn của một ai đó là một công việc rất khó khăn.

Từ đó cho thấy, cần phải tạo ra các chuẩn mang tính hướng dẫn cao, và phải được sự đồng thuận của những người phát triển một cách tự nguyện. Bộ các chuẩn phải cố gắng đơn giản nhất, có tính thống nhất cao và phù hợp với công nghệ đang được sử dụng.

Các chuẩn đưa ra có thể bao gồm các lĩnh vực:

- **Giao diện** – Các quy tắc, hướng dẫn về kích cỡ, màu sắc, bố cục... giao diện.
- **Viết mã** – Các quy tắc, hướng dẫn về đặt tên, bố cục, giải thích... trong mã nguồn.

Để áp dụng các chuẩn dễ hơn, có thể in và đặt ở vị trí sao cho những người phát triển có thể nhìn thấy ngay mỗi khi họ cần, ví dụ dán lên tường phía trước người phát triển. Để có thể in trên một tờ giấy, thì chuẩn phải cố gắng nhỏ gọn và tổng quát.

3.4. Kết chương

Trong chương này, tôi đã đưa ra một mô hình áp dụng các phương pháp phát triển nhanh trong phát triển các dự án có phạm vi nhỏ, yêu cầu thường xuyên thay đổi.

Cơ sở lý thuyết sử dụng để đưa ra mô hình này là những phương pháp phát triển nhanh đã và đang được nghiên cứu và áp dụng trên thế giới. Trong mô hình đề xuất, các ưu điểm của các phương pháp được trích chọn và áp dụng. Thêm vào đó, một số kỹ thuật tiêu biểu khác cũng được đưa vào.

Trong mỗi mục của chương này, đều đề xuất các biện pháp áp dụng cụ thể dựa trên điều kiện thực tế, trong đó có biện pháp rất cụ thể, có thể áp dụng được ngay, hoặc có những biện pháp mang tính hướng dẫn, và việc áp dụng có thể tùy biến theo người sử dụng.

Theo như đánh giá của tôi, thì mô hình đưa ra là phù hợp. Tuy nhiên, để có thể khẳng định được hiệu quả của mô hình, cần phải thử nghiệm trong thời gian dài, phải được cải tiến, sửa đổi sao cho phù hợp hơn và phải được đánh giá đầy đủ hơn.

CHƯƠNG 4 - ÁP DỤNG THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ NGHIÊN CỨU

Để có thể đánh giá được mô hình được đề xuất trong chương 3, cần phải đánh giá kết quả những dự án đã được triển khai thử nghiệm. Trong chương này, một số dự án áp dụng thử nghiệm sẽ được giới thiệu và đánh giá. Các kết quả thu được có thể dùng để đánh giá phần nào về những phương pháp đã được nghiên cứu.

4.1. Môi trường áp dụng

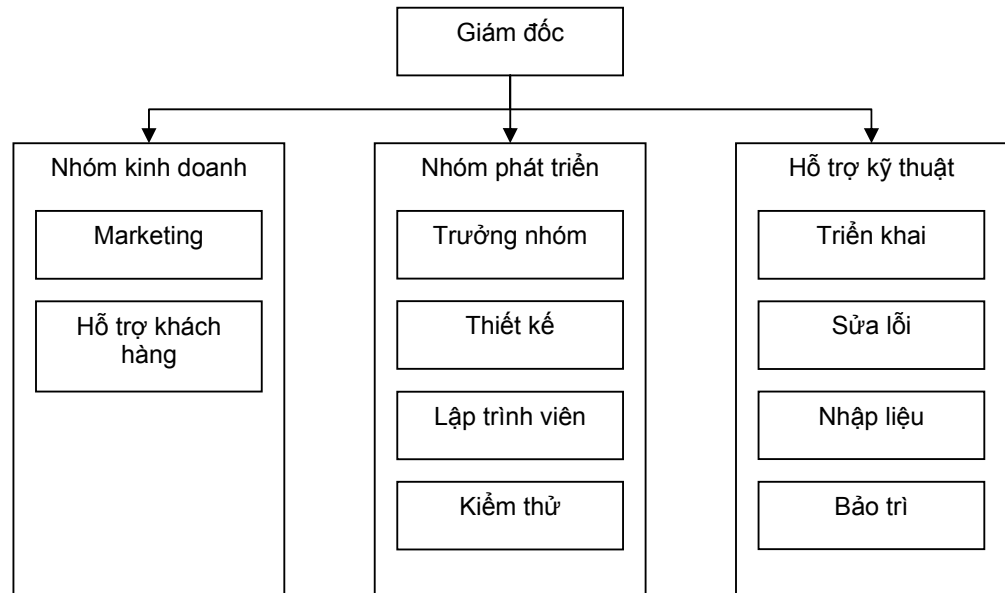
Các thử nghiệm được tiến hành trên một số dự án thuộc công ty TNHH Giải pháp kỹ thuật quốc tế (ITS). Đây là một công ty phát triển phần mềm trong các lĩnh vực Web và các ứng dụng quản lý. Phần này sẽ giới thiệu sơ lược một số đặc điểm của công ty.

4.1.1. Về tổ chức

Vì là một công ty nhỏ nên tổ chức cũng rất gọn nhẹ. Đứng đầu là giám đốc, người đóng vai trò đưa ra các quyết định, đồng thời cũng là người trực tiếp quản lý các dự án. Ngoài ra còn có các phó giám đốc phụ trách kỹ thuật, phó giám đốc kinh doanh.

Đội ngũ phát triển được chia thành các nhóm nhỏ, mỗi nhóm gồm từ 3 đến 10 người, đứng đầu là trưởng nhóm. Việc phân chia nhóm rất linh động, phụ thuộc vào quy mô cũng như tính chất của từng dự án.

Ngoài ra, đội ngũ kinh doanh, bán hàng cũng thường xuyên được điều động vào làm việc cùng với các nhóm phát triển, để có thể hiểu rõ hơn về hệ thống cũng như việc cung cấp các phản hồi từ phía khách hàng.



Hình 4.1 – Cơ cấu tổ chức công ty

4.1.2. Về nhân lực

Đối với những người lãnh đạo, đây là những người năng động, nhanh nhạy trong công việc và luôn ủng hộ việc đưa ra và áp dụng các cải tiến. Đây cũng là một trong những điều kiện tốt để có thể áp dụng mô hình thử nghiệm.

Việc áp dụng sẽ không thành công nếu không được sự hợp tác của những người trực tiếp phát triển phần mềm. Tuy đây đều là những người có kiến thức, năng lực chuyên môn nhưng để có thể đưa vào áp dụng một phương pháp mới thì điều cần thiết là phải đưa ra một mô hình đơn giản, dễ áp dụng và áp dụng được ngay.

4.1.3. Về công nghệ

Do tính chất nhỏ gọn, linh hoạt của đội ngũ phát triển nên việc áp dụng những công nghệ, kỹ thuật mới nhất trong phát triển phần mềm được thực

hiệu thường xuyên và hiệu quả. Kỹ thuật, công nghệ tốt sẽ làm tăng năng suất lao động, giảm thiểu lỗi và tăng khả năng đáp ứng thay đổi.

4.1.4. Đánh giá

Từ những đặc điểm được vừa nêu, có thể kết luận việc đưa các phương pháp phát triển nhanh vào áp dụng trong môi trường công ty là phù hợp và có thể đem lại hiệu quả cao hơn.

Ngoài ra, mô hình đề xuất còn có thể áp dụng tốt đối với những công ty khác, có quy mô, tổ chức và các đặc điểm khác tương tự.

4.2. Giới thiệu một số dự án thử nghiệm

Vì cách tiếp cận của các phương pháp phát triển nhanh tương đối mới lạ, do đó việc áp dụng mới chỉ ở bước đầu. Các dự án được áp dụng thử nghiệm có quy mô tương đối nhỏ, thời gian phát triển ngắn và nhân lực sử dụng cũng không nhiều.

4.2.1. Dự án phần mềm lập thời khoá biểu

4.2.1.1. Mô tả dự án

Dự án Phần mềm lập thời khoá biểu là một dự án nhằm xây dựng một phần mềm cho phép tạo lịch giảng cho từng giáo viên, từng môn học cho các trạm đào tạo xa. Phần mềm được xây dựng theo yêu cầu của khoa Đại học Tại chức, trường đại học Giao thông Vận tải.

Đây là một dự án nhỏ, phần mềm được yêu cầu với những chức năng chính sau:

- Quản lý chương trình học của tất cả các ngành học.

- Quản lý thông tin của tất cả các trung tâm đào tạo liên kết với khoa.
- Quản lý thông tin của các giáo viên, các bộ môn.
- Lập thời khoá biểu trực quan trên lịch.
- In phiếu báo giảng, thời khoá biểu và các báo cáo khác.

Vì tính chất công việc, khách hàng yêu cầu thời gian thực hiện dự án yêu cầu không quá 8 tuần, bắt đầu từ 4/2006.

4.2.1.2. Giải pháp thực hiện

Lựa chọn công nghệ:

Công nghệ được sử dụng đối với dự án này là công nghệ .NET của Microsoft, sử dụng môi trường Microsoft Visual Studio 2003 và cơ sở dữ liệu SQL Server 2000.

Nhân lực:

Do dự án nhỏ, nên đội phát triển chỉ gồm một nhóm có 3 thành viên, trong đó có 2 thành viên phát triển chính.

Một số đặc điểm trong áp dụng:

Quy trình quản lý và phát triển được sử dụng dựa trên mô hình được đề xuất, trong đó các khung thời gian được xác định cụ thể. Với thời gian phát triển ngắn, nên các khung thời gian được giảm xuống tối đa.

Giai đoạn khảo sát và lấy yêu cầu được thực hiện trong vòng một tuần, các yêu cầu khác được bổ sung trong quá trình thực hiện dự án.

Khung thời gian cho mỗi vòng lặp được tính theo tuần. Phiên bản thử nghiệm sẽ được đưa ra khi các chức năng chính được cài đặt. Các phiên bản tiếp theo được đưa ra hàng tuần.

Các kỹ thuật được sử dụng

Trong dự án này, một số kỹ thuật được áp dụng như giảm chu kỳ phát hành, áp dụng các kỹ thuật kiểm thử, thiết kế đơn giản và sử dụng các chuẩn trong thiết kế và trong lập trình.

4.2.1.3. Đánh giá kết quả thực hiện

Dự án đã kết thúc trong khoảng thời gian yêu cầu, đang được khai thác và được người sử dụng đánh giá cao.

Trong quá trình thực hiện dự án, một số kết quả thu được như sau:

- Thời gian khảo sát và lấy yêu cầu: khoảng 1 tuần.
- Thời gian đưa ra phiên bản đầu tiên: 3 tuần.
- Thời gian đưa ra các phiên bản cập nhật: 1 tuần.
- Số phiên bản cập nhật: 3.

Dự án được đánh giá là thành công, thời gian thực hiện đảm bảo và thoả mãn được yêu cầu khách hàng cũng như những mong muốn của người sử dụng. Bảng 4.1 đánh giá một cách tương đối kết quả thực hiện dự án trên một số tiêu chí.

STT	Tiêu chí đánh giá	Đánh giá
1	Tiến độ thực hiện dự án	Đúng hạn
2	Chất lượng từng giai đoạn	Hoàn thành tiến độ
3	Lỗi phát sinh	Thấp
4	Các thay đổi	Trung bình

5	Độ thoả mãn yêu cầu khách hàng	Cao
---	--------------------------------	-----

Bảng 4.1 – Đánh giá kết quả dự án 1

Do đây là một dự án nhỏ, nên kết quả thực hiện dự án chưa thể đánh giá được nhiều về mô hình được đề xuất. Tuy nhiên, dự án cho thấy bước đầu có thể áp dụng những phương pháp phát triển mới, mặc dù mới ở mức đơn giản.

4.2.2. Dự án Phần mềm quản lý bán hàng

4.2.2.1. Mô tả dự án

Phần mềm Quản lý bán hàng là một phần mềm bán hàng dành cho các cửa hàng vừa và nhỏ. Đây là một phần mềm đóng gói, với mục tiêu phục vụ cho nhiều cửa hàng khác nhau, với quy mô từ nhỏ tới trung bình và áp dụng cho nhiều loại mặt hàng khác nhau. Dự án được thực hiện từ 6/2006.

Phần mềm được xây dựng với một số nội dung chính sau:

- Quản lý danh mục mặt hàng, phân loại theo các tiêu chí khác nhau.
- Quản lý nhập kho, tồn kho.
- Quản lý bán hàng.
- Đưa ra các cảnh báo các mặt hàng thừa, thiếu, hết hạn...
- Đưa ra các báo cáo thống kê.

Với đặc điểm là phần mềm đóng gói, nên phần mềm phải được xây dựng với tính mở cao, dễ tùy biến.

4.2.2.2. Giải pháp thực hiện

Lựa chọn công nghệ:

Phần mềm được xây dựng sử dụng công nghệ .NET của Microsoft, được phát triển trên môi trường Microsoft Visual Studio 2005 và cơ sở dữ liệu SQL Server 2005. Đây là một công nghệ mới lần đầu tiên được áp dụng ở công ty, mặc dù được đánh giá là rất mạnh nhưng để khai thác tốt công nghệ này cần phải có một thời gian học và khai thác từng bước.

Nhân lực:

Đội phát triển dự án gồm một nhóm 4 thành viên, trong đó gồm cả nhóm trưởng.

Một số đặc điểm trong áp dụng:

Quy trình được áp dụng theo mô hình đã được đề xuất, trong đó có một số điểm khác biệt do tính chất của dự án.

Do đặc thù là phần mềm đóng gói, nên giai đoạn khảo sát và lấy yêu cầu cần phải được thực hiện lâu hơn, phải thu thập nhiều ý kiến khác nhau cũng như tham khảo những chương trình thương mại đã có.

Trong giai đoạn phát triển, việc kiểm thử đơn vị và kiểm thử chức năng được thực hiện bởi những người phát triển. Phiên bản đầu tiên được đưa ra khi hầu hết các chức năng đã được cài đặt. Phiên bản này được cung cấp cho một số khách hàng được mời sử dụng. Dựa vào những phản hồi của khách hàng để tiến hành cải tiến phần mềm.

Các kỹ thuật được sử dụng

Trong dự án này, một số kỹ thuật được áp dụng như: áp dụng các kỹ thuật kiểm thử, thiết kế đơn giản và sử dụng các chuẩn trong thiết kế và trong lập trình.

4.2.2.3. Đánh giá kết quả thực hiện

Hiện nay phiên bản beta 3 của sản phẩm đang được một số khách hàng sử dụng và phản hồi. Các phiên bản cập nhật của phần mềm vẫn được tiếp tục đưa ra.

Trong quá trình phát triển, đã thu được một số kết quả sau:

- Thời gian đưa ra phiên bản đầu tiên: 8 tuần.
- Thời gian đưa ra các phiên bản cập nhật: từ 2 đến 4 tuần.
- Số phiên bản cập nhật: 3.

Một số đánh giá được đưa ra trong bảng 4.2.

STT	Tiêu chí đánh giá	Đánh giá
1	Tiến độ thực hiện dự án	Đúng hạn
2	Chất lượng từng giai đoạn	Hoàn thành tiến độ
3	Lỗi phát sinh	Trung bình
4	Các thay đổi	Cao
5	Độ thoả mãn yêu cầu khách hàng	Trung bình

Bảng 4.2 – Đánh giá kết quả dự án 2

Dự án đang trong quá trình khai thác thử nghiệm. Các khách hàng tham gia sử dụng đều đánh giá tốt, mặc dù cần phải có thêm thời gian mới có thể đánh giá chính chính xác hiệu quả của dự án.

Đây là một dự án mà yêu cầu không rõ ràng, thường xuyên thay đổi. Phần mềm làm ra phải tùy biến theo mong muốn của từng người dùng cụ thể, do vậy việc áp dụng các quy trình, kỹ thuật được giới thiệu trong các phương pháp phát triển nhanh đã góp phần vào sự thành công của dự án.

4.2.3. Dự án Phần mềm quản lý nhà hàng phiên bản 2

4.2.3.1. Mô tả dự án

Phần mềm quản lý nhà hàng phiên bản 2 là phần mềm được phát triển dựa trên phần mềm Quản lý nhà hàng đã được thực hiện trước đó. Dự án được bắt đầu từ giữa 10/2006.

Phần mềm được xây dựng với một số chức năng chính sau:

- Quản lý các thực đơn.
- Quản lý nguyên vật liệu.
- Quản lý nhập, xuất kho.
- Quản lý đặt bàn, chế biến các món ăn.
- Đưa ra các báo cáo thống kê.

Tuy đây là phiên bản 2, nhưng được thiết kế lại hoàn toàn, sử dụng công nghệ mới và đưa vào nhiều cải tiến so với phiên bản trước đó.

4.2.3.2. Giải pháp thực hiện

Lựa chọn công nghệ:

Phần mềm xây dựng trên nền .NET của Microsoft, được phát triển trên môi trường Microsoft Visual Studio 2005 và cơ sở dữ liệu SQL Server 2005.

Nhân lực:

Đội phát triển dự án gồm một nhóm 3 thành viên.

Một số đặc điểm trong áp dụng:

Do đây là phần mềm đóng gói, nên quy trình áp dụng có những điểm tương tự như đã được đưa ra trong 4.1.2.2.

Vì đây là phiên bản nâng cấp, nên hầu hết các chức năng được thực hiện dựa trên phiên bản đã có. Ngoài ra, những yêu cầu mới thu được trong quá trình khai thác cũng được tập hợp lại để đưa vào sản phẩm mới.

4.2.3.3. Đánh giá kết quả thực hiện

Dự án vẫn đang được triển khai thực hiện. Các kết quả bước đầu cho thấy với việc áp dụng các quy trình, kỹ thuật của các phương pháp phát triển nhanh đã đem lại hiệu quả về tốc độ làm việc, tỷ lệ lỗi giảm, các giai đoạn hoàn thành đúng hạn.

4.3. Đánh giá chung

Qua việc áp dụng thử nghiệm trên một số dự án, có thể đưa ra một số đánh giá chung về khả năng, hiệu quả của việc áp dụng các phương pháp phát triển nhanh.

Thứ nhất, việc áp dụng một quy trình quản lý mới không phải là điều đơn giản. Việc này cần phải có được sự hỗ trợ của những người lãnh đạo, những người phát triển. Thực tế cho thấy, việc áp dụng thường không được ngay, mà phải thực hiện qua từng bước.

Thứ hai, vì áp dụng một mô hình mới, nên chỉ có thể đưa vào thử nghiệm trên một số dự án nhỏ, với thời gian phát triển ngắn. Kết quả các dự

án này chỉ phần nào đánh giá được hiệu quả của việc áp dụng. Tuy nhiên, các kết quả đó cũng đã chứng minh được tính khả thi của các phương pháp.

Từ đó cho thấy, để đánh giá được đầy đủ về những lợi ích mà các phương pháp đó đem lại, cần phải có thời gian áp dụng tương đối dài, đồng thời cần phải cải tiến mô hình áp dụng sao cho phù hợp hơn.

KẾT LUẬN

Qua luận văn này, tôi đã giới thiệu những điểm chính trong lĩnh vực quản lý dự án và các phương pháp triển phần mềm, trong đó giới thiệu chi tiết một số phương pháp phát triển phần mềm tiêu biểu thuộc lớp các phương pháp phát triển nhanh. Từ đó đề xuất ra mô hình áp dụng các phương pháp được nghiên cứu dựa trên hoàn cảnh cụ thể của môi trường áp dụng.

Luận văn được trình bày trong bốn chương. Chương 1 trình bày tổng quan về các nội dung cơ bản trong lĩnh vực quản lý dự án nói chung và dự án công nghệ thông tin nói riêng. Các phương pháp phát triển phần mềm cũng được đề cập sơ lược trong chương này.

Trong chương 2, một số phương pháp phát triển phần mềm tiêu biểu thuộc lớp các phương pháp phát triển nhanh được đề cập chi tiết. Các phương pháp được giới thiệu bao gồm: Extreme Programming, Scrum và Adaptive Software Development. Ngoài ra, phần đánh giá và so sánh các phương pháp cho thấy đặc điểm của từng phương pháp để có thể áp dụng hiệu quả.

Dựa trên những kiến thức nghiên cứu được trong chương 2, một mô hình áp dụng được đề xuất trong chương 3. Mô hình áp dụng là sự kết hợp những điểm mạnh của các phương pháp Extreme Programming và Scrum, và đưa vào một số kỹ thuật phụ trợ như làm việc tập trung, các kỹ thuật kiểm thử.

Mô hình đề xuất được áp dụng thử nghiệm trên một số dự án. Chương 4 giới thiệu sơ lược về các dự án được thử nghiệm và kết quả đánh giá của từng dự án. Những kết quả này cho thấy việc áp dụng các phương pháp phát triển nhanh đem lại hiệu quả bước đầu trong phát triển phần mềm, đồng thời

cũng cho thấy việc áp dụng còn nhiều khó khăn, cần phải có thêm nhiều sự ủng hộ hơn nữa từ phía những người lãnh đạo cũng như những người phát triển.

Những đóng góp khoa học của luận văn:

Luận văn đã đề cập đến một lĩnh vực nghiên cứu khá mới mẻ và có nhiều áp dụng trong thực tiễn, đó là các phương pháp phát triển nhanh. Một số phương pháp tiêu biểu được giới thiệu và đánh giá ưu điểm, hạn chế cũng như khả năng áp dụng của từng phương pháp.

Trong luận văn, người làm luận văn đã đưa ra một quy trình phát triển phần mềm và các kỹ thuật phát triển dựa trên việc kết hợp các phương pháp Scrum và Extreme Programming. Các bước của quy trình, các hướng dẫn thực hiện đều được chi tiết hoá thông qua các biện pháp cụ thể.

Những nghiên cứu mới đã được áp dụng trong một số dự án phần mềm. Các kết quả bước đầu cho thấy hiệu quả của việc áp dụng cũng như tính khả thi của các phương pháp.

Hướng phát triển tiếp theo của đề tài:

Tiếp tục nghiên cứu các phương pháp phát triển nhanh, bao gồm cả các phương pháp đã được đánh giá tương đối đầy đủ, cũng như các phương pháp mới được đề xuất.

Tăng cường việc áp dụng những nghiên cứu vào thực tiễn, để từ đó có thể cải tiến mô hình đề xuất dựa trên những kết quả thu được, đồng thời nghiên cứu, áp dụng các ưu điểm của các phương pháp mới.

TÀI LIỆU THAM KHẢO

1. Abrahamsson P., Salo O., Ronkainen J., Warsta J. (2002), *Agile software development methods Review and analysis*, VTT Publications 478, Finland.
2. Beck K. (2000), *Extreme Programming Explained: Embrace Change*, Addison Wesley, Boston.
3. Marchewka J. T. (2003), *Information Technology Project Management*, John Wiley & Son, United State.
4. Multiple authors (2001), *Manifesto for Agile Software Development*, <http://agilemanifesto.org>.
5. Schwaber K. (1996), *SCRUM Development Process*.
6. Theunissen W. H. M. (2003), *A case-study based assessment of Agile software development*, University of Pretoria, South Africa.
7. Wikipedia (2006), *Capability Maturity Model*, http://en.wikipedia.org/wiki/Capability_Maturity_Model.
8. Wikipedia (2006), *Test Driven Development*, http://en.wikipedia.org/wiki/Test_driven_development.
9. Wikipedia (2006), *Waterfall Model*, http://en.wikipedia.org/wiki/Waterfall_model.
10. Zagrodnick C. (2005), *Agile Development in Small and Medium Sized Projects*, Netherlands.

PHỤ LỤC

MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

PRINCIPLES BEHIND THE AGILE MANIFESTO

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.