



# Quy trình phát triển phần mềm

Bởi:

Khoa CNTT ĐHSPTK Hưng Yên

## Quy trình phát triển phần mềm

Mục đích của phần này là hướng dẫn cho bạn mọi thứ về quy trình phát triển phần mềm sẽ được áp dụng trong môn học này. Mục đích là cho bạn một cái nhìn tổng quan về tất cả các phần bên trong một sản phẩm phần mềm và thấy được một vài hướng tiếp cận chung thường được sử dụng ngày nay. Với những hiểu biết này, bạn sẽ tự có cách tốt nhất để áp dụng các kỹ năng kiểm thử phần mềm mà bạn sẽ được học.

Phần chính của bài này bao gồm:

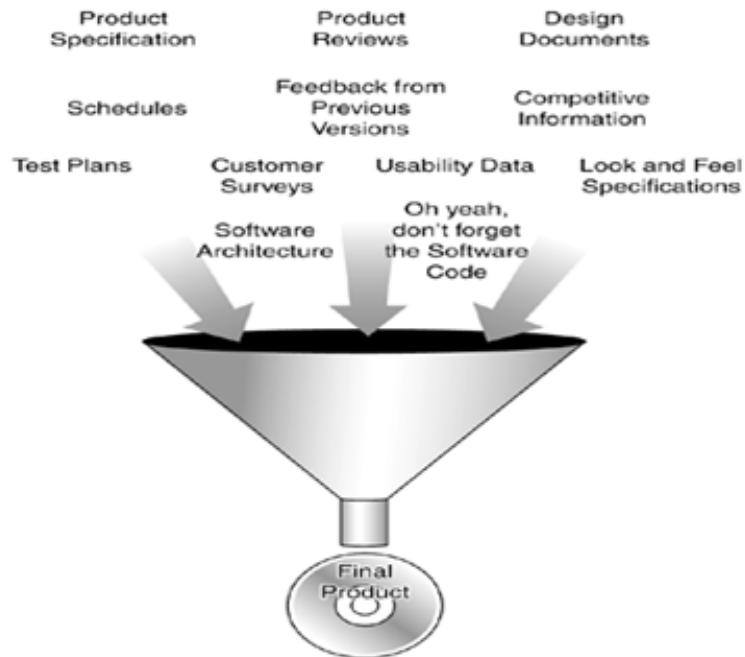
- Các thành phần (component) chính nào bên trong một sản phẩm phần mềm
- Những ai và các kỹ năng nào đóng góp vào một sản phẩm phần mềm
- Xử lý phần mềm như thế nào để từ một ý tưởng xây dựng lên một sản phẩm cuối cùng

## Các thành phần của phần mềm (product components)

Một sản phẩm phần mềm chính xác là cái gì? Nhiều người cho rằng, đơn giản nó là thứ mà người ta down được từ internet hoặc cài đặt được từ DVD để nó chạy được trên máy tính. Đây là một mô tả tốt, nhưng là một mô tả tốt trong một phạm vi nhỏ, nhưng thật sự, nhiều thành phần được ẩn bên trong phần mềm. Có nhiều phần bên trong hộp “*come in the box*”, mà chúng thường được lấy ra để trợ giúp hoặc có thể bị bỏ qua. Mặc dù rất dễ quên tất cả các phần này, nhưng là một tester, bạn cần biết về chúng. Bởi vì chúng là những nội dung cần kiểm tra và chúng có thể chứa lỗi.

1. Lỗi lực đằng sau một sản phẩm phần mềm như thế nào?

Trước tiên, bạn hãy nhìn nhận những lỗi lực phía sau một sản phẩm phần mềm. Hình 2.1 chỉ ra một vài những thành phần trừu tượng mà bạn có thể không hề nghĩ tới.



Rất nhiều nỗ lực (effort) ẩn dấu phía dưới một sản phẩm phần mềm

Vậy, đâu sẽ là tất cả những thứ này, bên cạnh mã nguồn thật sự, liệu đây có phải là một cái phễu (*funnel*) phần mềm? Nhìn lướt qua, có lẽ chúng là những thứ hiển nhiên mà một lập trình viên tạo ra. Và rõ ràng chúng không phải là những thứ mà có thể được xem trực tiếp từ CD – ROM. Nhưng để dùng một dòng diễn tả cho “*món mì ống thương mại*”: “chúng ở đây”. Ít nhất cũng là như vậy.

Những thuật ngữ được dùng trong ngành công nghiệp phần mềm để mô tả thành phần một sản phẩm phần mềm, mà đã được tạo ra và tiếp tục tới một nơi nào khác có thể được lan truyền. Cách dễ nhất để giải thích những thứ mà tất cả sự chuyển giao này là tổ chức chúng thành những loại lớn.

### 1. Yêu cầu của khách hàng

Phần mềm cần được viết một cách đầy đủ các yêu cầu mà một người hoặc một nhóm người đưa ra. Đó là những khách hàng. Để làm hợp lý những yêu cầu này, một nhóm phát triển phần mềm phải tìm ra những cái mà khách hàng muốn. Một nhóm thì phỏng đoán những yêu cầu, nhưng hầu hết các thông tin chi tiết được thu thập trong quá trình khảo sát, hỏi đáp từ những phiên bản trước của phần mềm, cạnh tranh các thông tin về sản phẩm, các nhìn tổng quan, các nhóm trọng tâm, và một số các phương thức khác, một số formal, một số các khác. Tất cả những thông tin này được tìm hiểu, xem xét và làm sáng tỏ để quyết định chính xác những đặc trưng nào mà sản phẩm phần mềm cần có.

HÃY ĐƯA CÁC FEATURES (ĐẶC TRUNG) NÀY VÀO PERSPECTIVE VỚI CÁC FOCUS GROUP (NHÓM TRỌNG TÂM): một phương tiện phổ biến để nhận những hồi đáp trực tiếp từ các khách hàng tiềm năng là sử dụng các *focus group*. *Focus group* thường được tổ chức bởi các công ty khảo sát độc lập - những người thiết đặt các cơ quan trong các phố mua bán lớn. Các cuộc khảo sát hoặc những chuyến đi bộ điền hình vòng quanh phố mua bán với một bìa kẹp hồ sơ và hỏi những người đi qua nếu họ muốn đóng góp một phần vào quá trình nghiên cứu. Họ sẽ hỏi một vài câu hỏi liên quan đến chất lượng như: “Bạn có một PC ở nhà không? Bạn sử dụng phần mềm X như thế nào? Bạn sử dụng bao nhiêu thời gian để online?” và tiếp tục... Nếu bạn phù hợp với yêu cầu về đối tượng, họ sẽ mời bạn quay trở lại một vài giờ để tham gia cùng với một vài người khác trong *focus group*. Ở đây bạn sẽ được hỏi các câu hỏi chi tiết hơn về phần mềm máy tính. Bạn có thể được biểu diễn những hộp phần mềm khác nhau và hỏi về những sở thích của bạn để bạn lựa chọn. Hoặc bạn có thể thảo luận như một đặc trưng nhóm (*group features*) giống như bạn nhìn thấy một sản phẩm mới. Tốt hơn hết bạn nên bỏ ra chút thời gian của mình. Hầu hết các *focus group* được hướng dẫn nhưng với tư cách là một công ty phần mềm mà yêu cầu thông tin được giữ kín. Nhưng thường thì rất dễ dàng để đoán ra họ là ai.

### 1. Đặc tả

Kết quả của quá trình nghiên cứu các yêu cầu của khách hàng chỉ là những dữ liệu thô. Nó không mô tả được những sản phẩm đề xuất, nó chỉ xác nhận những thứ nên hay không nên tạo ra và các đặc trưng mà khách hàng mong muốn. Các bản đặc tả lưu giữ các thông tin trên với các yêu cầu bắt buộc và đưa ra những định hình xem phần mềm sẽ là gì, sẽ làm gì, và trông nó như thế nào.

Định dạng của các bản đặc tả thay đổi rất nhiều. Đặc biệt, một số công ty mà các sản phẩm của họ dành cho chính phủ, cho vũ trụ, tài chính, và ngành công nghiệp được phẩm sử dụng một quy trình rất nghiêm khắc với nhiều sự kiểm tra ngặt nghèo và cân nhắc kỹ lưỡng. Kết quả thu được là một bản đặc tả vô cùng kỹ lưỡng, chi tiết được chốt lại, có nghĩa là không được phép thay đổi nó dưới mọi điều kiện. Mọi người trong nhóm phát triển biết chính xác chúng đang tạo nên cái gì.

Đây là các nhóm phát triển phần mềm, thường tạo ra những sản phẩm ít bị chê trách, những người đã đưa ra những bản đặc tả trên bàn ăn (*on cocktail napkins*), nếu họ tạo ra tất cả chúng. Đây là những thuận lợi dễ nhận thấy, chúng rất mềm dẻo, nhưng lại chứa đựng đầy rủi ro. Và sản phẩm cuối cùng không được biết đến cho đến khi nó được tung ra thị trường.

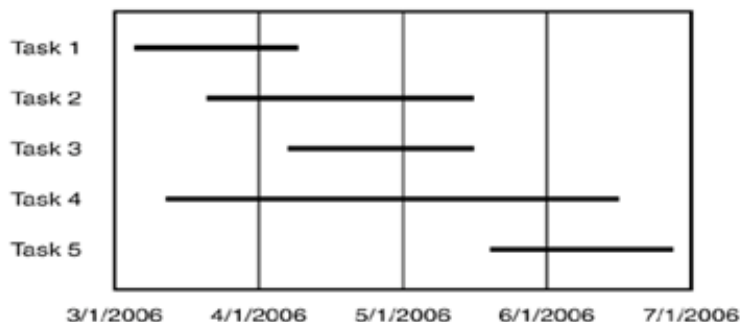
#### d) Kế hoạch làm việc (schedule)

Một phần rất quan trọng của quá trình sản xuất phần mềm là kế hoạch làm việc của nó. Là một dự án phát triển rất phức tạp với rất nhiều phần và nhiều người cùng tham gia.

## Quy trình phát triển phần mềm

Vì vậy, cần một số cơ cấu để theo dõi quá trình xử lý này. Nó có thể là một danh sách các nhiệm vụ đơn giản để hình thành nên biểu đồ *Gantt* (hình 2.2) để theo dõi chi tiết mỗi nhiệm vụ với phần mềm quản lý dự án.

Mục đích của *Schedule* là biết được công việc nào đã được hoàn thành, có bao nhiêu việc bị bỏ quên, và khi nào thì công việc được hoàn thành.



Một biểu đồ *Gantt* biểu diễn các nhiệm vụ của một dự án tương phản với các đường ngang biểu diễn thời gian (*horizontal timeline*)

### e) Tài liệu thiết kế phần mềm

Một nhận thức sai lầm rất phổ biến là khi một lập trình viên tạo ra một chương trình, đơn giản là anh ta ngồi xuống và bắt đầu viết code. Điều này có thể xảy ra trong một cửa hàng phần mềm nhỏ và không chuyên nghiệp. Nhưng ở các công ty lớn, dù là với phần mềm nhỏ nhất cũng phải trải qua quá trình thiết kế để lập kế hoạch về cách mà phần mềm sẽ được viết. Trong cuốn sách này, phần mềm cũng yêu cầu được phác thảo và lập kế hoạch trước khi những đoạn mã đầu tiên được gõ, hoặc được xây dựng.

Những tài liệu mà những lập trình viên tạo ra biến đổi rất nhiều phụ thuộc vào công ty, dự án, và nhóm phát triển, những mục đích của chúng đều là lập kế hoạch và tổ chức mã được viết.

Đây là một danh sách một vài tài liệu thiết kế phần mềm rất phổ biến:

- Kiến trúc (*Architecture*): Một tài liệu mô tả cho toàn bộ thiết kế của phần mềm, bao gồm mô tả của tất cả các thành phần lớn và cách mà chúng gây ảnh hưởng tới các bộ phận khác.
- Sơ đồ luồng dữ liệu (*Data flow diagram*): Một sơ đồ chính thức biểu diễn dữ liệu xuyên suốt một chương trình. Thỉnh thoảng nó tham chiếu tới một *bubble chart* bởi vì nó sẽ gây chú ý hơn với các vòng tròn (*circle*) và các dòng (*line*)

## Quy trình phát triển phần mềm

- Sơ đồ chuyển trạng thái (*State transition diagram*): Một sơ đồ chính thức khác phá vỡ phần mềm ở trạng thái cơ bản, hoặc các điều kiện, và biểu diễn các phương tiện di chuyển từ trạng thái này đến trạng thái khác.

- Sơ đồ luồng (*Flow chart*): Các phương tiện truyền thống diễn đạt bằng hình ảnh mô tả luồng logic của phần mềm. Ngày nay, flowcharting không còn phổ biến, nhưng khi nó được sử dụng, viết code từ một flowchart chi tiết là một quá trình xử lý đơn giản.

- Mã chú giải (*commented code*): Có một cách nói cũ rằng bạn có thể viết code một lần, nhưng nó sẽ được đọc bởi bất kỳ ai, ít nhất là 10 lần. Bởi vậy, những lời chú giải hợp lý cho các đoạn code là rất quan trọng, vì vậy, các lập trình viên được giao nhiệm vụ bảo trì code coa thể dễ dàng hiểu được đoạn mã đó làm gì và làm như thế nào.

### f) Tài liệu kiểm thử

Tài liệu kiểm thử được thảo luận chi tiết từ bài 17 đến bài 20. Nhưng nó vẫn được đề cập ở đây bởi vì nó là thành phần không thể thiếu để tạo nên một sản phẩm phần mềm. Với các lý do này, các lập trình viên phải lập kế hoạch và xây dựng tài liệu cho công việc của họ, tester phải hiểu rõ điều này. Không ai nghe thấy rằng một nhóm kiểm thử phần mềm phải tạo ra nhiều khả năng chuyển giao (*deliverables*) hơn các lập trình viên.

Đây là một danh sách *test deliverables* quan trọng:

- Kế hoạch kiểm thử (*test plan*) mô tả toàn bộ các phương thức được sử dụng để thay đổi phần mềm sao cho phù hợp với bản đặc tả và các yêu cầu của khách hàng. Nó bao gồm mục tiêu về chất lượng, các yêu cầu về tài nguyên, kế hoạch làm việc, những nhiệm vụ được giao, phương thức, và những thứ tương tự như thế (*and so forth*)

- Danh sách các trường hợp kiểm thử (*test case list*) Những phần sẽ được kiểm tra và mô tả từng bước chi tiết và sẽ được thực hiện theo để kiểm tra phần mềm.

- Báo cáo lỗi (*bug reports*) mô tả các vấn đề được phát hiện nhờ các *test case*. Có thể chúng không được ghi ra giấy nhưng chúng sẽ được theo dõi qua database.

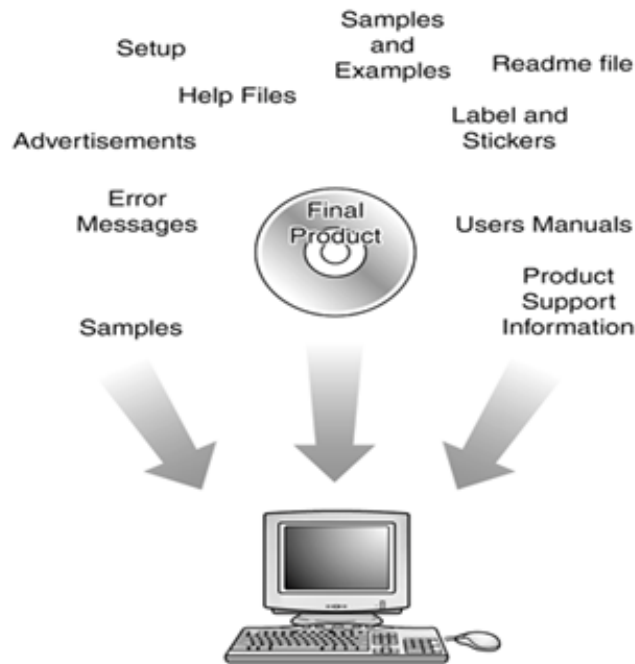
- Các công cụ kiểm thử và kiểm thử tự động (*Test tools and automation*) được mô tả chi tiết trong bài 13. Nếu nhóm của bạn sử dụng các công cụ tự động để kiểm thử phần mềm, thì hoặc là chúng được mua hoặc được tự viết, và chúng đưa ra kết quả bằng tài liệu.

### g) Thành phần tạo nên một sản phẩm phần mềm

Như vậy, trong bài này bạn đã được tìm hiểu về các lỗi lực để tạo ra một sản phẩm phần mềm. Cũng cần phải nhận thức rằng khi một sản phẩm đã sẵn sàng để được đóng gói và mang đi thì không phải mỗi code được chuyển đi, còn rất nhiều những bộ phận khác đi

## Quy trình phát triển phần mềm

cùng với nó (hình 2.3). Bởi vì tất cả những phần này cũng được này cũng được thấy và sử dụng bởi khách hàng, chúng cũng cần được kiểm tra.



CD-ROM phần mềm là một trong rất nhiều phần tạo nên một sản phẩm phần mềm

Nhưng thật không may, các thành phần này thường xuyên bị bỏ qua trong quy trình kiểm tra phần mềm. Chắc hẳn rằng bạn cũng đã thử sử dụng các trợ giúp gắn liền với sản phẩm và thấy nó không được tiện dụng lắm thậm chí rất là tồi tệ. Hoặc có lẽ bạn đã kiểm tra yêu cầu hệ thống trên một sticker ở bên cạnh hộp phần mềm (*software box*) chỉ khám phá ra sau khi bạn mua phần mềm nhưng nó lại không hoạt động trên PC của bạn. Dường như, việc kiểm thử là rất đơn giản, nhưng không hẳn thế, hãy kiểm tra lại chúng một lần nữa trước khi đưa phần mềm ra thị trường. Bạn sẽ làm vậy chứ.

Sau khi đọc cuốn sách này, bạn sẽ được biết về những bộ phận không phải là phần mềm này (*non-software pieces*) và cách để kiểm tra chúng một cách hợp lý. Sau đó, hãy giữ lại danh sách này trong đầu như một ví dụ rằng một sản phẩm phần mềm thì không chỉ là code:

|                      |                            |
|----------------------|----------------------------|
| Help files           | User's manual              |
| Samples and examples | Labels and stickers        |
| Product support info | Icons and art              |
| Error messages       | Ads and marketing material |

|                        |             |
|------------------------|-------------|
| Setup and installation | Readme file |
|------------------------|-------------|

ĐỪNG QUÊN KIỂM TRA NHỮNG THÔNG ĐIỆP LỖI: thông điệp lỗi (error message) là những phần dễ bị bỏ qua nhất trong một sản phẩm phần mềm. Các lập trình viên, không phải những người thực sự có kinh nghiệm, mà cụ thể là những người viết ra chúng. Hiếm khi họ lập kế hoạch mà thường sửa chữa dần chương trình trong khi kiểm tra các lỗi. Vì vậy, thật khó khăn khi các tester muốn tìm thấy và hiển thị đầy đủ các lỗi. Đừng đưa ra những thông điệp lỗi gây sợ e sợ trong phần mềm của bạn.

Error: Keyboard not found. Press F1 to continue.

Can't instantiate the video thing.

Windows has found an unknown device and is installing a driver

for it.

A Fatal Exception 006 has occurred at 0000:00000007.