

	VIETTEL AI RACE	TD061
	KHẢO SÁT ĐẶC TẢ VÀ MÃ NGUỒN	Lần ban hành: 1

Chương này giới thiệu các kỹ thuật khảo sát đặc tả và mã nguồn. Mã nguồn được phát triển dựa trên đặc tả và vì thế việc khảo sát đặc tả cần được tiến hành trước khi phát triển mã nguồn để tránh các rủi ro về các lỗi có thể có trong đặc tả về sản phẩm phần mềm. Vì đặc tả không thể thực thi được trên máy nên chúng ta chỉ có thể phát hiện các lỗi bằng các kỹ thuật khảo sát đặc tả. Tuy nhiên, mã nguồn thì có thể thực thi được và có thể được kiểm thử thông qua việc chạy trên máy. Vì vậy, liệu chúng ta có cần phải khảo sát mã nguồn trước không? Câu trả lời là rất cần vì khảo sát giúp ta phát hiện các lỗi sớm, các lỗi về logic, các lỗi về cấu trúc và giúp đề xuất các ca kiểm thử hiệu quả hơn. Khảo sát mã nguồn và không tiến hành phần mềm để phát hiện lỗi trong quá trình phát triển phần mềm được gọi là kiểm thử hộp trắng tĩnh (static white-box testing). Đây là một kỹ thuật kiểm thử bổ sung vào các kỹ thuật kiểm thử khác để đảm bảo chất lượng phần mềm. Còn khảo sát đặc tả được liệt vào phạm trù của kiểm thử hộp đen tĩnh (static black-box testing) vì việc này thường được tiến hành khi chưa có mã nguồn.

1. Khảo sát đặc tả

Đặc tả phần mềm là một tài liệu quan trọng, mô tả các chức năng mà phần mềm cần có cũng như các ràng buộc mà phần mềm cần thỏa mãn. Tài liệu này được tạo ra từ nhiều nguồn khác nhau như thông qua các nghiên cứu về nhu cầu sử dụng của người dùng, về các biểu mẫu, về thị trường, v.v. Việc tài liệu này được tạo ra thế nào và viết ra dưới dạng nào không phải là mối quan tâm của người kiểm thử, miễn là nó đã được đúc kết thành một tài liệu mô tả sản phẩm để phát triển và người kiểm thử sẽ tiến hành các khảo sát trên tài liệu này để tìm các lỗi đặc tả có thể có. Cũng có trường hợp đặc tả không được viết ra. Trong trường hợp này, nó ở trong đầu của người thiết kế và lập trình, và người kiểm thử phải khảo sát các tài liệu này bằng việc phỏng vấn họ. Mục này giới thiệu hai kỹ thuật khảo sát đặc tả là duyệt đặc tả mức cao và duyệt đặc tả mức thấp nhằm nâng cao chất lượng của các đặc tả phần mềm. Đây là những công việc đầu tiên và không thể thiếu trước khi tiến hành các hoạt động đảm bảo chất lượng về sau.

1.1 Tiến hành duyệt đặc tả mức cao

Định nghĩa một sản phẩm phần mềm là một việc khó. Đặc tả thường liên quan đến nhiều thứ chưa biết, việc xây dựng đặc tả lấy vô số đầu vào thay đổi, kết hợp chúng với nhau để tạo nên một tài liệu mô tả sản phẩm mới. Quá trình này là khoa học không chính xác và rất dễ mắc sai lầm.

Bước thứ nhất để kiểm thử đặc tả không phải đi vào chi tiết ngay để tìm lỗi mà là xem xét nó từ mức cao. Hãy khảo sát đặc tả để tìm các lỗi cơ bản lớn, những lỗi về bỏ sót trước đã. Việc xem xét đặc tả lúc này là theo quan điểm nghiên cứu nhiều hơn là kiểm thử. Chỉ khi đã hiểu những “tại sao” và “làm thế nào” ở đằng sau đặc tả, bạn mới có thể phân biện tốt các chi tiết trong nó. Sau đây là các kỹ thuật để tiến hành duyệt đặc tả mức cao.

1.1.1 Hãy là khách hàng của sản phẩm

	VIETTEL AI RACE	TD061
	KHẢO SÁT ĐẶC TẢ VÀ MÃ NGUỒN	Lần ban hành: 1

Khi nhận một tài liệu đặc tả để kiểm thử, điều dễ nhất người kiểm thử cần làm là hãy giả định mình là khách hàng của sản phẩm. Vì thế, trước hết cần tìm hiểu ai sẽ là khách hàng của sản phẩm. Hãy nói chuyện với những người chào hàng và ma-ket-ting cho sản phẩm để tìm hiểu về người dùng cuối cùng của sản phẩm. Nếu sản phẩm là trung gian trong một dự án phần mềm khác, hãy tìm hiểu ai sẽ dùng nó và nói chuyện với họ.

Điều quan trọng là cần phải hiểu khách hàng chờ đợi gì ở sản phẩm này. Thỏa mãn yêu cầu người dùng là yếu tố chất lượng quan trọng nhất. Để hiểu yêu cầu người dùng không nhất thiết phải là một chuyên gia trong lĩnh vực ứng dụng. Tuy nhiên, hiểu biết chút ít về nó sẽ giúp cho việc kiểm thử tốt hơn.

Không được giả thiết bất cứ cái gì khi tìm hiểu về đặc tả. Khi ta khảo sát một phần của đặc tả và không hiểu thì không được cho là nó đúng. Ta sẽ dùng đặc tả để thiết kế các ca kiểm thử sau này. Vì thế, nếu không hiểu được đặc tả thì sẽ không thiết kế tốt các ca kiểm thử. Xem xét đặc tả theo quan điểm người dùng giúp ta phát hiện những lỗi bỏ sót hoặc sai với yêu cầu của họ. Một điều cần lưu ý là tính an ninh và bảo mật thường được giả thiết bởi người dùng. Khi giả định là người dùng, người kiểm thử không được bỏ qua yêu cầu này.

1.1.2 Hãy nghiên cứu các chuẩn và hướng dẫn hiện hành

Trên thế giới đã có nhiều nghiên cứu về cách con người sử dụng máy tính, và hiện nay đã có những chuẩn cả về phần cứng lẫn phần mềm về giao diện với người sử dụng, tương tác người máy, v.v. Vì thế, người phát triển phần mềm tốt nhất hãy tuân thủ nghiêm ngặt các chuẩn này. Các chuẩn về tương tác người máy được cải tiến để đáp ứng tốt nhất các yêu cầu của người dùng. Sau đây là một số chuẩn có thể nghiên cứu để áp dụng trong đặc tả phần mềm:

Hợp thức các thuật ngữ và quy ước. Nếu phần mềm được làm riêng cho một công ty nào đó hãy sử dụng các thuật ngữ và quy ước của họ.

- **Yêu cầu công nghiệp.** Trong mỗi lĩnh vực ứng dụng như y tế, dược phẩm, tài chính, v.v. đều có các chuẩn riêng và nghiêm ngặt của họ mà phần mềm phải tuân thủ.
- **Chuẩn quy định bởi chính phủ.** Chính phủ có những quy định, đặc biệt trong các lĩnh vực quốc phòng, an ninh và quản lý mà phần mềm phải tuân thủ.
- **Giao diện đồ họa với người sử dụng (GUI).** Các phần mềm chạy trong Windows hoặc Macintosh phải tuân thủ các quy định về giao diện đồ họa của các hệ điều hành này.
- **Chuẩn bảo mật.** Phần mềm có thể phải thỏa mãn một số quy định về bảo mật mà cần phải được chứng nhận và cấp phép.

Người kiểm thử cần nắm được các chuẩn này để kiểm thử xem phần mềm có thỏa mãn hay không, có gì bị bỏ qua hay không. Các chuẩn này được coi là một phần của đặc tả khi thẩm định phần mềm.

1.1.3 Hãy xem xét và kiểm thử các phần mềm tương tự

Một trong các phương pháp để hiểu phần mềm của ta sẽ như thế nào là nghiên cứu các sản phẩm tương tự. Đó có thể là sản phẩm cạnh tranh hoặc sản phẩm nào đó giống như sản phẩm ta đang phát triển. Rất có thể điều đó đã được làm bởi người quản lý dự án hoặc chính người viết đặc tả cho sản phẩm ta đang phát triển. Các phần mềm không thể giống hệt nhau (là lý do để ta xây dựng sản phẩm tương tự), nhưng việc nghiên cứu các sản phẩm tương tự này giúp ta

	VIETTEL AI RACE	TD061
	KHẢO SÁT ĐẶC TẢ VÀ MÃ NGUỒN	Lần ban hành: 1

xây dựng cách tiếp cận kiểm thử sản phẩm của mình.

Những điểm cần tìm kiếm khi xem xét các sản phẩm tương tự bao gồm:

Ôm

- **Kích cỡ.** Các đặc trưng sẽ nhiều hơn hay ít hơn? Chương trình sẽ ít hay nhiều lệnh hơn, việc kiểm thử có bị ảnh hưởng bởi kích cỡ không?
- **Độ phức tạp.** Tương tự như kích cỡ, độ phức tạp sẽ cao hơn hay thấp hơn, và điều đó ảnh hưởng thế nào đến kiểm thử?
- **Tính khả kiểm thử.** Liệu ta có đủ tài nguyên, thời gian và trình độ để kiểm thử phần mềm như vậy?
- **Chất lượng và độ tin cậy.** Liệu phần mềm này đại diện cho chất lượng của phần mềm đang xây dựng, độ tin cậy sẽ cao hơn hay thấp hơn?
- **Tính bảo mật.** Phần mềm cạnh tranh này có tính bảo mật so với sản phẩm ta đang phát triển thế nào?

Ta sẽ thu được nhiều kinh nghiệm để khảo sát đặc tả của sản phẩm của mình bằng việc xem xét các vấn đề trên.

1.2 Các kỹ thuật kiểm thử đặc tả ở mức thấp

Sau khi đã hoàn thành việc khảo sát đặc tả bậc cao, ta hiểu rõ hơn về sản phẩm của mình và những yếu tố bên ngoài ảnh hưởng đến thiết kế của sản phẩm phần mềm. Khi được trang bị những thông tin này, chúng ta sẽ tiếp tục khảo sát đặc tả ở mức thấp và chi tiết hơn. Mục này sẽ giải thích các chi tiết cần tiến hành nhằm đạt được mục tiêu này.

1.1.4 Danh sách các hạng mục cần thẩm định về các thuộc tính của đặc tả

Một đặc tả sản phẩm phần mềm được xây dựng tốt cần thỏa mãn tám thuộc tính sau đây:

- **Đầy đủ.** Liệu đặc tả còn thiếu cái gì không? Đã đủ chi tiết chưa? Liệu nó đã bao gồm mọi điều cần thiết để không phụ thuộc vào tài liệu khác?
- **Trúng đích.** Liệu nó đã cung cấp lời giải đúng đắn cho bài toán, liệu nó đã xác định đầy đủ các mục tiêu và không có lỗi.

	VIETTEL AI RACE	TD061
	KHẢO SÁT ĐẶC TẢ VÀ MÃ NGUỒN	Lần ban hành: 1

- **Chính xác, không nhập nhằng và rõ ràng.** Mô tả có chính xác không, có rõ ràng và dễ hiểu không, liệu còn có gì là nhập nhằng với ý nghĩa không xác định?
- **Tương thích.** Các đặc trưng và chức năng được mô tả có bị xung đột với nhau không?
- **Hợp lệ.** Các khẳng định có thực sự cần thiết để mô tả đặc trưng sản phẩm không? Có gì thừa không? Có thể truy ngược về yêu cầu của người dùng không?
- **Khả thi.** Liệu đặc tả có thể được cài đặt trong khuôn khổ nhân lực, công cụ, tài nguyên, thời gian và kinh phí cho phép hay không?
- **Phi mã lệnh.** Trong đặc tả không được dùng các câu lệnh hoặc thuật ngữ cho người lập trình. Ngôn ngữ dùng trong đặc tả phải là phổ biến với người dùng.
- **Khả kiểm thử.** Liệu các đặc trưng có thể kiểm thử được? Liệu đã cung cấp đủ thông tin để có thể kiểm thử và xây dựng các ca kiểm thử?

Khi xây dựng đặc tả phần mềm, hãy duyệt đặc tả (văn bản, hình vẽ, v.v.) và đánh giá xem nó có thỏa mãn các thuộc tính nêu trên chưa.

1.1.5 Danh sách các hạng mục cần thẩm định về các thuật ngữ của đặc tả

Bên cạnh danh sách các thuộc tính trên đây là danh sách các từ hay có vấn đề cần tìm trong đặc tả khi phản biện. Sự xuất hiện của các từ này có thể tiết lộ rằng các đặc trưng được mô tả chưa được suy nghĩ thấu đáo và không gặp đủ các thuộc tính trên đây. Hãy tìm các từ này trong đặc tả và xem xét cẩn thận xem có lỗi ở đó không.

- **Luôn luôn, mỗi một, tất cả, không có, không bao giờ.** Những từ như vậy mô tả sự tuyệt đối và chắc chắn. Hãy xét xem tình trạng có đúng như vậy không, có trường hợp nào vi phạm các khẳng định đó hay không.
- **Tất nhiên, do đó, rõ ràng là, hiển nhiên là.** Các từ này được dùng để thuyết phục người khác chấp nhận cái gì đó. Đừng có rơi vào các bẫy đó.
- **Nào đó, đôi khi, thông thường, thường gặp, hầu hết.** Các từ này là nhập nhằng, không có nghĩa rõ ràng và không thể kiểm thử, chẳng hạn bạn phải kiểm thử tính “đôi khi” thế nào?
- **Vân vân, chẳng hạn, như vậy.** Các từ này thường mô tả thứ mà không thể kiểm thử được.
- **Tốt, nhanh, rẻ, hiệu quả, ổn định.** Những từ không định lượng như vậy sẽ mô tả các hạng mục không thể kiểm thử được nếu không được làm chi tiết hóa sau này.

	VIETTEL AI RACE	TD061
	KHẢO SÁT ĐẶC TẢ VÀ MÃ NGUỒN	Lần ban hành: 1

- **Xử lý, từ chối, bỏ qua, bị khử.** Những thuật ngữ này thường dấu trong nó những chức năng cần phải đặc tả đầy đủ.
- **Nếu ... thì (thiếu trái lại).** Trường hợp “trái lại” có thể bị bỏ quên. Hãy tránh các khẳng định như vậy.

Trên đây là kỹ thuật kiểm thử hộp đen tĩnh, tức là phân biện đặc tả của sản phẩm, tìm xem đặc tả có lỗi không. Khi đặc tả đã được hoàn thành và đã được phản biện, mã nguồn của chương trình được phát triển dựa trên đặc tả này. Bây giờ, chúng ta sẽ sử dụng kỹ thuật tương tự để phân biện/khảo sát mã nguồn.

2. Khảo sát mã nguồn

Mục này giới thiệu các kỹ thuật cơ sở để tiến hành kiểm chứng thiết kế và mã nguồn của phần mềm. Nội dung chính của mục này bao gồm việc phân tích lợi ích của công việc, các kiểu khảo sát mã nguồn, một số hướng dẫn về chuẩn lập trình, và cách chung để khảo sát mã nguồn nhằm tìm lỗi.

1.3 Khảo sát thiết kế và mã nguồn hay là việc kiểm thử hộp trắng tĩnh

Kiểm thử tĩnh tức là việc kiểm thử chỉ gồm việc khảo sát mà không cần tiến hành chương trình, còn kiểm thử hộp trắng là việc kiểm thử có trong tay mã nguồn của chương trình. Vì thế kiểm thử hộp trắng tĩnh chính là việc khảo sát thiết kế và mã nguồn của chương trình. Công việc này bao gồm một quy trình để khảo sát một cách cẩn thận và có phương pháp đối với thiết kế, kiến trúc và mã nguồn của phần mềm để tìm lỗi mà không cần tiến hành phần mềm. Công việc này còn có tên khác nữa là *phân tích cấu trúc*.