

Chương 14

■ **Đánh giá chất lượng**

Slide đi cùng với sách

Software Engineering: A Practitioner's Approach, 7/e

Của Roger S. Pressman

Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman

Chỉ sử dụng cho giảng dạy phi lợi nhuận

Chỉ có thể được sao chép cho sinh viên tại trường đại học khi sử dụng kết hợp với *Software Engineering: A Practitioner's Approach, 7/e*. Cấm sao chép khi không được sự cho phép của tác giả

Toàn bộ thông tin bản quyền phải được ghi đầy đủ trên các trang web cung cấp tài liệu cho sinh viên.

Chất lượng phần mềm

- Năm 2005, *ComputerWorld* [Hil05] phàn nàn rằng
 - Bệnh dịch “phần mềm tồi” lây lan gần như tất cả các tổ chức sử dụng máy tính, gây lãng phí thời gian làm việc, mất/hỏng dữ liệu, bỏ lỡ cơ hội bán hàng, nâng cao chi phí hỗ trợ, bảo trì, giảm sự hài lòng của khách hàng
- Một năm sau, *InfoWorld* [Fos06] viết
 - “Trạng thái đáng xin lỗi của chất lượng phần mềm” cho thấy chất lượng phần mềm không khá hơn chút nào
- Ngày nay, vấn đề chất lượng phần mềm vẫn là vấn đề
 - Khách hàng đổ lỗi cho nhà phát triển, cho rằng thiếu kinh nghiệm dẫn đến phần mềm chất lượng thấp
 - Nhà phát triển đổ lỗi cho khách hàng (và các bên liên quan) rằng thời gian giao hàng chưa hợp lý, các thay đổi liên tục được đưa ra khiến họ phải giao phần mềm trước khi nó sẵn sàng

Chất lượng

- *American Heritage Dictionary* định nghĩa *quality* (chất lượng):
 - “một đặc tính, thuộc tính của một cái gì đó”
- Với phần mềm, có 2 loại chất lượng đáng quan tâm
 - **Chất lượng thiết kế** bao gồm các yêu cầu, thông số kỹ thuật, thiết kế của hệ thống
 - **Sự phù hợp** tập trung vào việc cài đặt cụ thể
 - **Sự hài lòng của người dùng** = sản phẩm tương thích + chất lượng tốt + chuyển giao trong ngân sách và thời gian

Chất lượng – Góc nhìn triết học

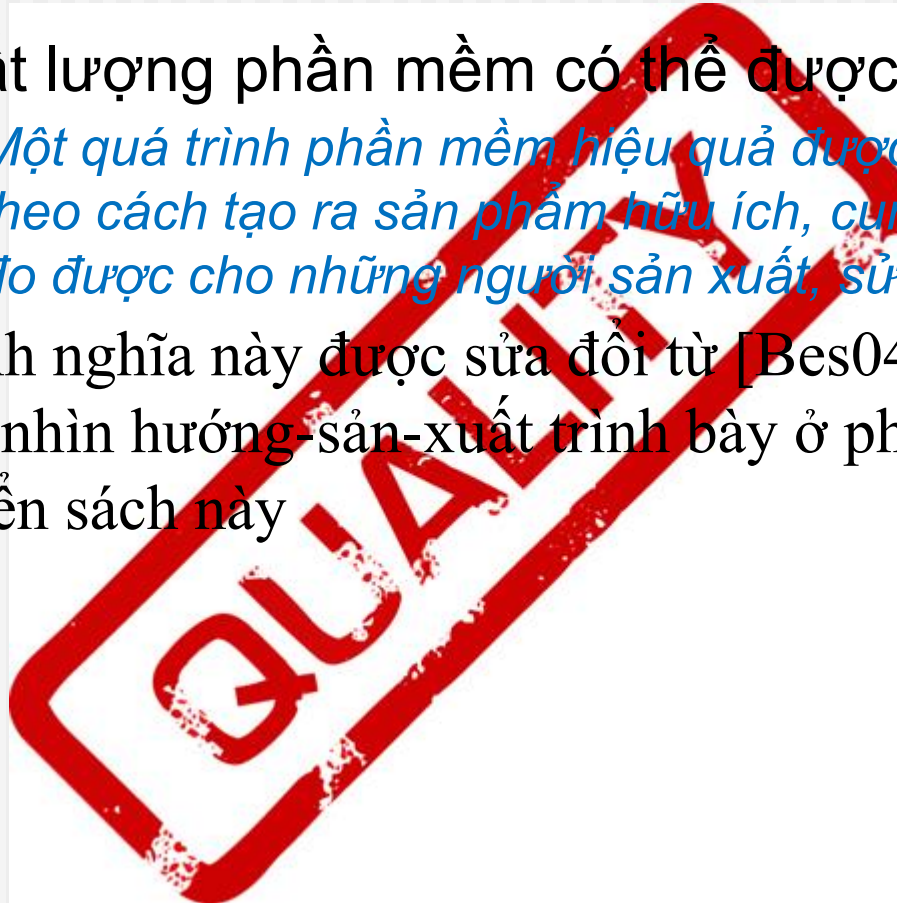
- Robert Persig [Per74] nhận xét về những điều ta gọi là chất lượng (tạm dịch)
- Chất lượng... bạn biết nó là gì, nhưng bạn không biết nó là gì. Nhưng đó là tự mâu thuẫn. Nhưng một số thứ tốt hơn so với những thứ khác, có nghĩa là, chúng có chất lượng hơn. Nhưng khi ta cố gắng để nói chất lượng là gì, ngoài những thứ có nó, ta đều mắc kẹt! Không có gì để nói. Nhưng nếu bạn không thể nói chất lượng là gì, làm thế nào bạn biết nó là gì, hoặc làm thế nào bạn biết rằng nó thậm chí còn tồn tại? Nếu không ai biết nó là gì, thì với tất cả các mục đích thực tế nó không tồn tại. Nhưng đối với tất cả các mục đích thực tế nó lại thực sự tồn tại. Các lớp học còn dựa trên điều gì khác? Tại sao khác nào con người phải thử vận may cho một số thứ và ném những thứ khác trong đồng rác? Rõ ràng là một số điều tốt hơn so với những điều khác... Nhưng tốt hơn là gì? . . . Vòng đi và vòng lại, chất lượng là cái quái gì? Nó là cái gì?

Chất lượng – cái nhìn thực dụng

- *Siêu việt luận* xem chất lượng như một thứ bạn ngay lập tức nhận ra, nhưng không thể xác định một cách rõ ràng
- *Người dùng* thấy chất lượng trong mục tiêu cụ thể cuối cùng. Sản phẩm đáp ứng mục tiêu là sản phẩm chất lượng
- *Nhà sản xuất* định nghĩa chất lượng trong đặc điểm kỹ thuật của sản phẩm. Sản phẩm phù hợp với đặc điểm kỹ thuật, nó đạt chất lượng
- *Sản phẩm* cho thấy chất lượng có thể gắn với đặc điểm vốn có của nó.
- Cuối cùng, *góc nhìn giá trị* định lượng chất lượng dựa trên số tiền khách hàng sẵn sàng trả cho sản phẩm đó
- Trong thực tế, chất lượng gồm tất cả các quan điểm trên và nhiều hơn nữa.

Chất lượng phần mềm

- Chất lượng phần mềm có thể được định nghĩa
 - *Một quá trình phần mềm hiệu quả được áp dụng theo cách tạo ra sản phẩm hữu ích, cung cấp giá trị đo được cho những người sản xuất, sử dụng nó.*
- Định nghĩa này được sửa đổi từ [Bes04] và thay thế góc nhìn hướng-sản-xuất trình bày ở phần trước quyển sách này



Quy trình phần mềm hiệu quả

- *Một quá trình phần mềm hiệu quả* thiết lập một cơ sở hạ tầng, hỗ trợ các quá trình xây dựng phần mềm chất lượng cao sau này
- Quản lý các khía cạnh của quá trình giúp kiểm tra và cân bằng giúp tránh dự án hỗn loạn – nguyên nhân dẫn đến chất lượng kém
- Thực hành xây dựng phần mềm giúp các nhà phát triển phân tích vấn đề và thiết kế giải pháp – cả hai đều quan trọng để xây dựng phần mềm chất lượng cao
- Cuối cùng, các hoạt động như thay đổi quản lý và đánh giá kỹ thuật cũng quan trọng trong việc đánh giá chất lượng như bất kỳ giai đoạn nào

Sản phẩm hữu ích

- Một *sản phẩm hữu ích* cung cấp nội dung, chức năng và tính năng mà người dùng mong muốn
- Quan trọng hơn, nó cung cấp một cách đáng tin cậy, không có lỗi.
- Một sản phẩm hữu ích luôn luôn đáp ứng những yêu cầu đã được thống nhất rõ ràng bởi các bên liên quan.
- Ngoài ra, nó đáp ứng các yêu cầu ẩn (ví dụ như dễ sử dụng) được mong đợi ở tất cả các phần mềm chất lượng cao.

Giá trị bổ sung

- Nhờ việc *bổ sung giá trị cho NSX và NSD* của một sản phẩm phần mềm, phần mềm chất lượng cao cung cấp lợi ích cho tổ chức phát triển và người sử dụng phần mềm
- Tổ chức phần mềm đạt giá trị gia tăng vì phần mềm chất lượng cao đòi hỏi phải nỗ lực bảo trì ít hơn, ít lỗi, và ít phải hỗ trợ khách hàng.
- Cộng đồng người dùng đạt giá trị gia tăng vì các ứng dụng cung cấp một khả năng hữu ích trong một số quá trình kinh doanh.
- Kết quả cuối cùng:
 - (1) Doanh thu phần mềm tăng,
 - (2) Lợi nhuận tốt hơn khi ứng dụng vào một quá trình kinh doanh,
 - (3) cải thiện tính khả dụng của thông tin.

Không gian chất lượng

- David Garvin [Gar87]:
 - **Hiệu suất.** Phần mềm có cung cấp tất cả các nội dung, chức năng và tính năng được quy định theo mô hình và cung cấp giá trị cho người dùng cuối?
 - **Tính năng.** Liệu phần mềm cung cấp các tính năng mà người dùng cuối bất ngờ và thích thú?
 - **Độ tin cậy.** Liệu các phần mềm có thực hiện được tất cả các tính năng? Phần mềm có sẵn khi nó cần thiết? Phần mềm cung cấp chức năng mà không có lỗi?
 - **Độ phù hợp.** Các phần mềm có phù hợp với tiêu chuẩn địa phương và các phần mềm ngoại vi có liên quan? Nó có phù hợp với thiết kế trên thực tế và quy chuẩn lập trình?

Không gian chất lượng

- **Độ bền.** Phần mềm có thể được duy trì (thay đổi) hoặc sửa chữa (debugged) mà không có của tác dụng phụ không mong đợi? Thay đổi có gây ra tỷ lệ lỗi hoặc độ tin cậy suy giảm theo thời gian?
- **Khả năng hoạt động.** Phần mềm có thể được duy trì (thay đổi) hoặc sửa chữa (debugged) trong một khoảng thời gian chấp nhận được. Nhân viên hỗ trợ có thể có được tất cả thông tin họ cần để thực hiện thay đổi hoặc sửa lỗi?
- **Thẩm mỹ.** Hầu hết chúng ta sẽ đồng ý rằng một thực thể thẩm mỹ có sự sang trọng nhất định, một dòng chảy độc đáo, và sự hiện diện "rõ ràng" đó là khó để định lượng nhưng hiển nhiên để thấy.
- **Nhận thức** Trong một số trường hợp, bạn có một tập hợp các định kiến sẽ ảnh hưởng đến nhận thức của chất lượng.

Góc nhìn khác

- **McCall's Quality Factors** (*SEPA*, Section 14.2.2)
- **ISO 9126 Quality Factors** (*SEPA*, Section 14.2.3)
- **Targeted Factors** (*SEPA*, Section 14.2.4)

Chất lượng phần mềm?

- Nếu bạn tạo ra một hệ thống phần mềm có chất lượng tồi, bạn thua bởi vì không ai muốn mua nó.
- Mặt khác, bạn dành rất nhiều thời gian, nỗ lực, và các khoản tiền rất lớn để xây dựng phần mềm, hoàn toàn hoàn hảo, nó sẽ mất quá lâu để hoàn thành và quá tốn kém, khiến bạn không thể kinh doanh
- Bạn sẽ bỏ qua nhu cầu thị trường, hoặc bạn sẽ hết tất cả các nguồn lực.
- Vì vậy trong ngành công nghiệp, các công ty cố đạt ngưỡng trung bình, sản phẩm đủ tốt để không bị từ chối, nhưng cũng không đạt mức hoàn hảo để không tốn quá nhiều nguồn lực phát triển

Phần mềm “đủ tốt”

- Phần mềm đủ tốt cung cấp các chức năng chất lượng cao và các tính năng người dùng mong muốn, tuy nhiên cũng tồn tại các chức năng tối nghĩa, chuyên ngành và một số chứa năng chứa lỗi.
- Luận điểm chống lại “đủ tốt”
 - Phần mềm "đủ tốt" có thể làm việc trong một số lĩnh vực ứng dụng và cho một ít các công ty phần mềm lớn. Sau đó, nếu một công ty có một ngân sách tiếp thị lớn và có thể thuyết phục khách hàng đủ để mua phiên bản 1.0, họ đã thành công.
 - Nếu bạn làm việc cho một công ty nhỏ nên cảnh giác với triết lý này. Nếu bạn cung cấp một sản phẩm “đủ tốt” (có chứa lỗi), bạn có nguy cơ mất vĩnh viễn đến danh tiếng của công ty bạn.
 - Bạn có thể không bao giờ có cơ hội để cung cấp phiên bản 2.0 bởi vì tiếng xấu có thể làm cho doanh thu của bạn để giảm mạnh và công ty của bạn phá sản
 - Nếu bạn làm việc trong một số lĩnh vực ứng dụng (phần mềm nhúng, phần mềm ứng dụng được tích hợp với phần cứng nấu cầu thả có thể khiến công ty của bạn chịu kiện tụng tốn kém.

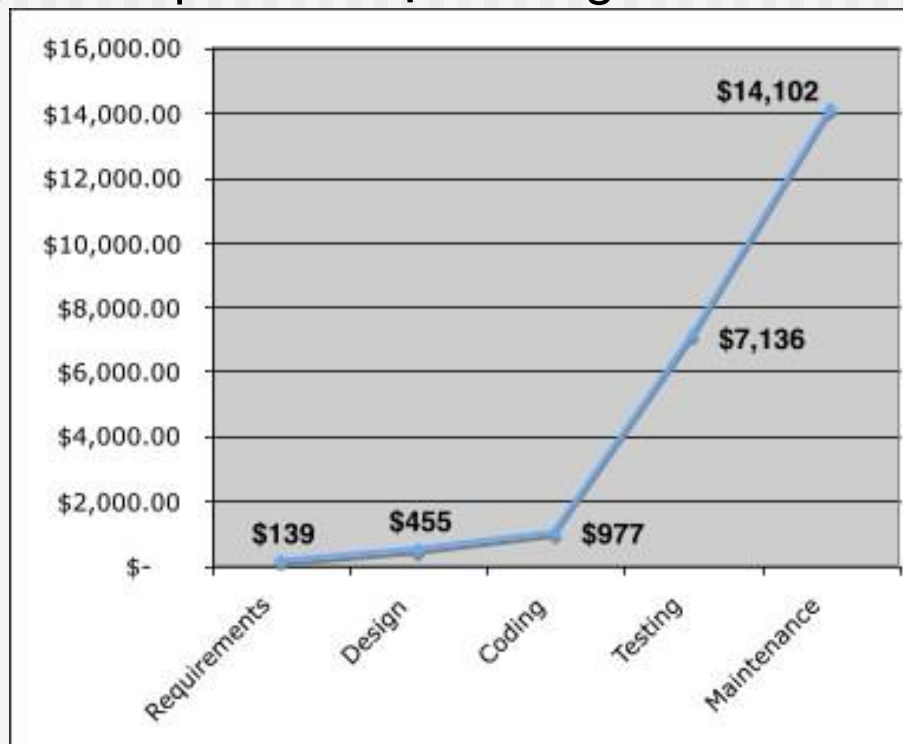
Chi phí của chất lượng

- **Chi phí phòng ngừa** bao gồm
 - Chất lượng kế hoạch
 - Đánh giá kỹ thuật chính thức
 - Thiết bị kiểm tra
 - Đào tạo
- **Chi phí thất bại bên trong** bao gồm
 - Làm lại
 - Sửa chữa
 - Phân tích thất bại
- **Chi phí thất bại bên ngoài** bao gồm
 - Giải quyết khiếu nại
 - Sản phẩm trả lại và thay thế
 - Hỗ trợ khách hàng
 - Bảo hành



Chi phí

- Các chi phí liên quan để tìm và sửa chữa lỗi tăng đáng kể khi ta đi từ chi phí phòng chống đến chi phí thất bại bên trong và chi phí thất bại bên ngoài



Chất lượng và rủi ro

- "Mọi người đặt cược công việc của họ, Tiện nghi của họ, an toàn của họ, sự giải trí của họ, quyết định của họ và cuộc sống rất của họ vào phần mềm máy tính. Nó tốt hơn nên đúng." SEPA, Chapter 1
- Ví dụ:
- Trong suốt tháng 11, năm 2000 tại một bệnh viện ở Panama, 28 bệnh nhân nhận được lớn liều tia gamma khi điều trị một loạt các bệnh ung thư. Trong những tháng sau đó, năm người trong số những bệnh nhân đã chết vì nhiễm độc phóng xạ và 15 người khác phát triển biến chứng nghiêm trọng. Điều gì gây ra thảm kịch này? Một gói phần mềm, phát triển bởi một công ty Mỹ, sửa đổi bởi kỹ thuật viên bệnh viện để tính toán liều bức xạ cho từng bệnh nhân.

Sơ suất và trách nhiệm pháp lý

- Câu chuyện đã trở nên quá phổ biến: Một tổ chức chính phủ hoặc công ty thuê một nhà phát triển phần mềm lớn hoặc công ty tư vấn để phân tích yêu cầu và sau đó thiết kế và xây dựng một phần mềm dựa trên "hệ thống" để hỗ trợ một số hoạt động lớn.
 - Hệ thống có thể hỗ trợ một chức năng lớn (ví dụ, quản lý trợ cấp) hoặc một số chức năng chính phủ (ví dụ như, quản lý chăm sóc sức khỏe hoặc bảo mật nội địa).
- Công việc bắt đầu với ý định tốt nhất ở cả hai bên, nhưng khi hệ thống được phân phối, nhiều điều không tốt có thể xảy ra.
- Hệ thống cung cấp muộn, không cung cấp tính năng và chức năng mong muốn, dễ bị lỗi, và không đáp ứng được sự chấp thuận của khách hàng. Tranh chấp nảy sinh.

Chất lượng và an ninh

Obama to confront China on cyber-spying...

- Gary McGraw bình luận [Wil05]:
- “An ninh phần mềm liên quan hoàn toàn đến chất lượng. Bạn phải suy nghĩ về bảo mật, độ tin cậy, tính sẵn sàng, độ phụ thuộc — ngay khi bắt đầu, trong việc thiết kế, kiến trúc, kiểm tra, và giai đoạn mã hóa, tất cả thông qua chu kỳ sống phần mềm. Mọi người thường nhận thức vấn đề an ninh phần mềm khi đã tập trung vào cuối chu kỳ phần mềm. Bạn tìm thấy vấn đề phần mềm càng sớm càng tốt. Và còn có hai loại vấn đề phần mềm. Một là bug, là vấn đề cài đặt. Hai là lỗi trong phần mềm — vấn đề trong thiết kế kiến trúc. Con người chú ý quá nhiều đến bug và không chú trọng đến sai sót kiến trúc.”

DAVE GRANLUND © www.davegranlund.com

Xây dựng phần mềm chất lượng

- Các yếu tố quan trọng:
 - **Phương pháp xây dựng phần mềm**
 - **Kỹ thuật quản lý dự án**
 - **Kiểm soát chất lượng**
 - **Đảm bảo chất lượng**