

Chapter 23

■ Product Metrics

Slide đi kèm với

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

by Roger S. Pressman

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

Chỉ dùng cho mục đích giáo dục phi lợi nhuận.

Có thể sửa đổi slide chỉ nhằm mục đích phục vụ sinh viên đại học trong những môn học liên quan tới sách *Software Engineering: A Practitioner's Approach, 7/e*. Nghiêm cấm mọi hoạt động sửa đổi khác hoặc sử dụng không được sự cho phép của tác giả.

Mọi thông tin bản quyền phải được đi kèm nếu những slide này được đăng lên mạng để phục vụ sinh viên.

Tam giác McCall chất lượng



Một bình luận

Các yếu tố chất lượng McCall đã được đề xuất trong năm 1970. Họ có giá trị như ngày hôm nay khi họ đang ở trong thời điểm đó. Đó là khả năng mà phần mềm được xây dựng để phù hợp với những yếu tố này sẽ triển lãm chất lượng cao cũng vào thế kỷ 21, thậm chí nếu có những thay đổi đáng kể trong công nghệ.

Những biện pháp, số liệu, chỉ số

- Một biện pháp cung cấp chỉ định về mức độ, số lượng, kích thước, công suất, hoặc kích thước của một và thuộc tính của sản phẩm hoặc tiến trình.
- Các thuật ngữ IEEE định nghĩa một metric là "một biện pháp định lượng về mức độ một hệ thống, thành phần, hoặc quá trình sở hữu một thuộc tính nhất định."
- Chỉ tiêu là một thước đo hoặc kết hợp số liệu cung cấp cái nhìn sâu sắc vào quá trình phần mềm, một dự án phần mềm, hoặc các sản phẩm.

Những nguyên tắc đo lường

- Những mục tiêu đo lường nên được thiết lập trước bắt đầu tập hợp dữ liệu.
- Mỗi giá trị kỹ thuật nên được định nghĩa một cách rõ ràng.
- Những giá trị nên được bắt nguồn giả thiết trên một giải thiết mà có giá trị trên lĩnh vực ứng dụng (ví dụ số lượng thiết kế cho thiết kế vẽ trên các định nghĩa và nguyên tắc và cố gắng cung cấp một dấu hiệu sự hiện diện một thuộc tính như mong muốn).
- Các giá trị (Metrics) nên được thay đổi để đáp ứng tốt các sản phẩm và quy trình cụ thể [Base 84].

Quy trình đo lường

- **Xây dựng.** Nguồn gốc của các biện pháp phần mềm và số liệu thích hợp cho các đại diện của các phần mềm đang được xem xét.
- **Tập hợp.** Cơ chế sử dụng để tích lũy dữ liệu cần thiết để lấy được các số liệu xây dựng.
- **Phân tích.** Tính toán các số liệu và các ứng dụng của các công cụ toán học.
- **Giải thích.** Việc đánh giá các số liệu kết quả trong một nỗ lực để đạt được cái nhìn sâu sắc vào chất lượng của các đại diện.
- **Thông tin phản hồi.** Khuyến nghị xuất phát từ việc giải thích chi phí sản phẩm truyền tới các nhóm phần mềm.

Mục tiêu định hướng đo lường phần mềm

- **Mục tiêu/Câu hỏi**
 - (1) thiết lập một mục tiêu đo lường rõ ràng đó là cụ thể cho các tiến trình hoạt động hoặc các đặc tính sản phẩm được đánh giá.
 - (2) xác định một tập câu hỏi mà cần phải trả lời để tiến đến mục tiêu.
 - (3) Xác định số liệu nổi bật trợ giúp để trả lời những câu hỏi này.
- **Khuôn mẫu định nghĩa mục tiêu**
 - phân tích {tên của hoạt động hoặc thuộc tính được đo}
 - mục đích {mục tiêu tổng thể của việc phân tích}
 - Đối với {các khía cạnh của hoạt động này hoặc thuộc tính đó được coi} với
 - quan điểm của {những người có lợi ích trong việc đo lường}
 - trong bối cảnh của {môi trường mà trong đó việc đo lường diễn ra}.

Các thuộc tính Metrics

- ***Đơn giản và sự tính toán.*** Nên đơn giản dễ dàng để học thế nào để lấy được số liệu và tính toán giá trị không nên yêu cầu nỗ lực và thời gian.
- ***Tin tưởng theo kinh nghiệm và trực giác.*** Các số liệu phải thoả mãn những ý niệm trực giác của kỹ sư về các thuộc tính sản phẩm đang được xem xét.
- ***Nhất quán và khách quan.*** Các số liệu nên luôn luôn mang lại kết quả rõ ràng.
- ***Nhất quán khi sử dụng các đơn vị và kích thước:*** Việc tính toán toán học của các số liệu nên sử dụng các biện pháp đó không dẫn đến sự kết hợp kỳ lạ của đơn vị.
- ***Sự phụ thuộc ngôn ngữ lập trình:*** Giá trị này được dựa trên các mô hình phân tích, mô hình thiết kế, hoặc cấu trúc của chương trình.
- ***Tổ chức hiệu quả phản hồi chất lượng:*** Đó là các số liệu cần cung cấp một kỹ sư phần mềm với thông tin mà có thể dẫn đến một sản phẩm cuối cùng có chất lượng cao hơn.

Các nguyên tắc thu thập và phân tích

- Thu thập và phân tích dữ liệu bất cứ khi nào có thể.
- Kỹ thuật thống kê nên được áp dụng để thiết lập mối quan hệ giữa thuộc tính bên trong và đặc tính chất lượng bên ngoài
- Hướng dẫn diễn giải và kiến nghị cần được thiết lập cho mỗi giá trị.

Metrics cho yêu cầu mô hình

- **Chức năng-Chi phí cơ bản:** sử dụng các điểm chức năng là một yếu tố bình thường hoặc đo lường kích thước rõ ràng.
- **Những chi phí cụ thể:** sử dụng như một dấu hiệu của chất lượng bằng cách đo số lượng yêu cầu theo mỗi loại.

Chức năng- Metrics base

- Các điểm chức năng (FP), lần đầu tiên bởi Albrecht [ALB79], có thể được sử dụng có hiệu quả như một phương tiện để đo chức năng được cung cấp bởi một hệ thống.
- Điểm chức năng có nguồn gốc sử dụng một mối quan hệ thực nghiệm dựa vào các biện pháp đếm được (trực tiếp) của thông tin phần mềm và đánh giá của các phần mềm phức tạp
- Những giá trị thông tin được xác định theo cách sau đây:
 - Số lượng giá trị đầu vào từ bên ngoài (EIs)
 - Số lượng đầu ra từ bên ngoài (E N Os)
 - Số lượng yêu cầu từ bên ngoài (EIs)
 - Số lượng tập tin logic bên trong (ILFs)
 - Số lượng tập tin giao diện bên ngoài (EIs)

Những điểm chức năng

Information Domain Value	Count	Weighting factor					
		simple	average	complex			
External Inputs (EIs)	<input type="text"/>	3	3	4	6	=	<input type="text"/>
External Outputs (EOs)	<input type="text"/>	3	4	5	7	=	<input type="text"/>
External Inquiries (EQs)	<input type="text"/>	3	3	4	6	=	<input type="text"/>
Internal Logical Files (ILFs)	<input type="text"/>	3	7	10	15	=	<input type="text"/>
External Interface Files (EIFs)	<input type="text"/>	3	5	7	10	=	<input type="text"/>
Count total	<div><div></div></div>						<input type="text"/>

Metrics thiết kế kiến trúc

- **Chi phí thiết kế kiến trúc**
 - Độ phức tạp thời gian tính = $g(\text{fan-out})$
 - Độ phức tạp lưu trữ dữ liệu = $f(\text{input \& output variables, fan-out})$
 - Độ phức tạp của hệ thống = $h(\text{structural \& data complexity})$
- **HK metric:** architectural complexity as a function of fan-in and fan-out
- **Morphology metrics:** a function of the number of modules and the number of interfaces between modules

Metrics cho thiết kế hướng đối tượng

- Whitmire [Whi97] mô tả chín đặc điểm riêng biệt và có thể đo lường trong thiết kế hướng đối tượng :
 - **Kích thước**
 - Kích thước được xác định theo bốn điểm: tổng công nhân số, khối lượng, chiều dài, và chức năng phức tạp.
 - **Phức tạp**
 - Những lớp trong bản thiết kế có quan hệ với nhau như thế nào?
 - **Thiết bị kết nối**
 - Thiết bị vật lý kết nối giữa các thành phần trong thiết kế hướng đối tượng.
 - **Sự đầy đủ**
 - Mức độ mà tổng quan các tính năng cần thiết , hoặc mức độ mà một thành phần thiết kế sở hữu các tính năng tổng quan từ các điểm bên trong các ứng dụng hiện đại.
 - **Sự đầy đủ**
 - Một ý nghĩa gián tiếp về mức độ mà các khái niệm trừu tượng hoặc thiết kế thành phần có thể được tái sử dụng.

Metrics cho thiết kế hướng đối tượng II

■ Sự gắn kết

- Mức độ mà tất cả phép tính làm việc cùng nhau để đưa ra 1 kiến trúc, 1 mục đích rõ ràng.

■ Nguyên thủy

- Áp dụng cho các hoạt động và các lớp, mức độ mà một hoạt động là nguyên tử

■ Tính tương đồng

- Mức độ mà hai hay nhiều lớp tương tự nhau về cấu trúc, chức năng, hành vi, hoặc mục đích.

■ Tính thay đổi

- Tất cả các biện pháp đều có khả năng thay đổi.

Đặc điểm phân biệt

■ **Berard [Ber95] chứng tỏ rằng các đặc điểm sau đây đòi hỏi các số liệu OO đặc biệt được phát triển:**

- Địa phương hóa—cách thức mà thông tin tập trung trong một chương trình.
- Sự đóng gói—đóng gói dữ liệu và xử lý.
- Che dấu thông tin—thông tin về chi tiết vận hành được che dấu dưới giao diện an toàn.
- Kế thừa—phương thức, thuộc tính của lớp này có thể truyền tới lớp khác
- Trừu tượng—cách thức cho phép thiết kế tập trung vào chi tiết cần thiết

Metrics lớp định hướng

Đề xuất của Chidamber và Kemerer [Chi94]:

- Phương pháp trọng số cho mỗi lớp
- Độ sâu của cây thừa kế
- Số lượng con
- Ghép nối giữa các lớp đối tượng
- Đặc trưng của lớp
- Phương pháp gắn kết thiếu

Metrics lớp định hướng

Đề xuất của Lorenz và Kidd [Lor94]:

- Quy mô lớp
- Số hoạt động ghi đè bởi một lớp con
- Số hoạt động thêm bởi một lớp con
- Chuyên môn hóa chỉ số

Metrics lớp định hướng

MOOD Metrics Suite [Har98b]:

- Hệ số kế thừa
- Hệ số ghép nối
- Hệ số đa hình

Metrics lớp định hướng

Đề xuất của Lorenz và Kidd [Lor94]:

- Quy mô hoạt động trung bình
- Độ phức tạp thao tác
- Số lượng trung bình của tham số cho mỗi thao tác

Metrics thành phần thiết kế

- **Metrics mức gắn kết:** hàm số của đối tượng dữ liệu và vị trí của nó trong định nghĩa
- **Metrics độ gắn kết:** hàm số của thông số đầu vào, đầu ra, biên toàn cục và module gọi tới
- **Metrics độ phức tạp:** được đề xuất (ví dụ cyclomatic complexity)

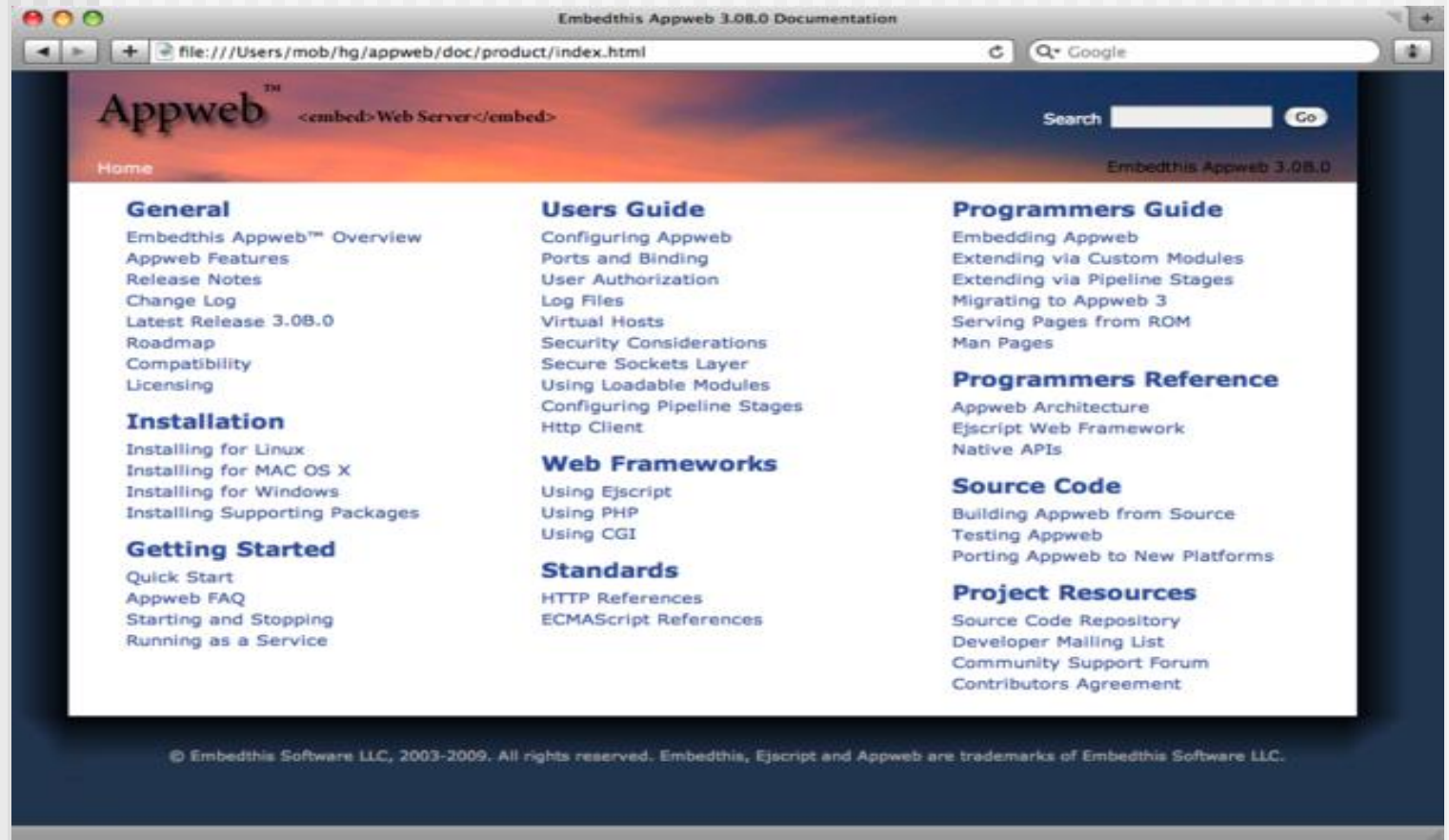
Metrics giao diện thiết kế

- **Bố cục phù hợp:** hàm số của vị trí thực thể, tọa độ địa lý và chi phí cho quá trình chuyển đổi giữa các thực thể

Thiết kế Metrics cho ứng dụng Web

- Giao diện người sử dụng có hữu dụng?
- Tính thẩm mỹ của AppWeb có thích hợp và hài lòng người sử dụng?
- Nội dung thiết kế đã ngắn gọn và đủ thông tin?
- Chuyển hướng đã hiệu quả và đơn giản nhất?
- Kiến trúc AppWeb đã được thiết kế phù hợp với mục đích người sử dụng chưa? Cấu trúc của nội dung và chức năng, luồng chuyển hướng yêu cầu sử dụng hệ thống hiệu quả không?
- Thành phần được thiết kế có làm giảm các thủ tục phức tạp và tăng cường tính chính xác, độ tin cậy, hiệu suất?

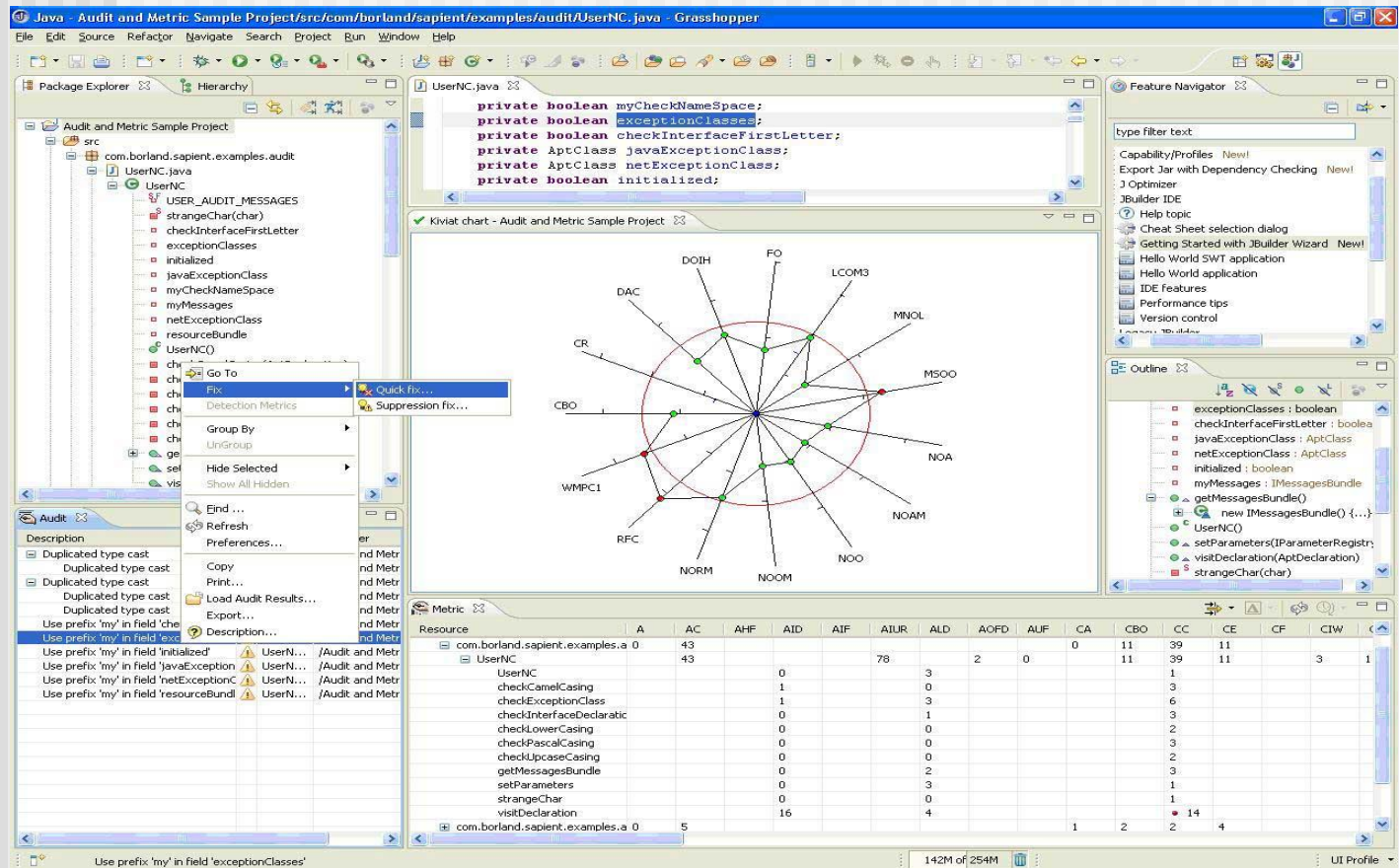
AppWeb



Code Metrics

- **Phần mềm Halstead:** bộ đầy đủ các metrics vị từ về số lượng (đếm and số lần xuất hiện) của toán tử và toán hạng trong một đoạn hoặc cả chương trình.
 - Cần lưu ý rằng "luật" Halstead đã tạo ra tranh cãi đáng kể, và nhiều người tin rằng các lý thuyết cơ bản có sai sót. Tuy nhiên, thực nghiệm trên các ngôn ngữ lập trình được chọn đã được chứng minh (ví dụ [FEL89]).

Code Metrics



Metrics for Testing

- Nỗ lực kiểm thử có thể ước lượng bằng cách sử dụng metrics thu được từ biện pháp Halstead
- Binder [Bin94] đề xuất một loạt các thiết kế metrics có ảnh hưởng trực tiếp đến khả năng kiểm thử của hệ thống hướng đối tượng.
 - Phương pháp gắn kết thiếu(LCOM).
 - Tỷ lệ công khai và bảo mật(PAP).
 - Truy nhập công khai cho các thành phần dữ liệu(PAD).
 - Số lớp gốc(NOR).
 - Fan-in (FIN).
 - Số con (NOC) và độ sâu cây thừa kế(DIT).

Maintenance Metrics

- IEEE Std. 982.1-1988 [IEE94] suggests a *chỉ số chất lượng phần mềm* (SMI) cho biết tính ổn định của sản phẩm phần mềm đã được phát triển. Các thông tin sau được xác định
 - M_T = số lượng các module trong bản phát hành hiện tại
 - F_c = số lượng các module trong bản phát hành hiện đã được thay đổi
 - F_a = số lượng các module trong bản phát hành hiện tại đã được thêm vào
 - F_d = số lượng các mô-đun từ việc phát hành trước đó đã bị xóa trong bản phát hành hiện tại
- *Chỉ số chất lượng phần mềm* được xác định như sau
$$SMI = [M_T - (F_a + F_c + F_d)] / M_T$$
- SMI xấp xỉ 1 thì phần mềm ổn định.