

Chapter 25

■ Quy trình và tiêu chuẩn Project

Slide đi cùng

Software Engineering: A Practitioner's Approach, 7/e
by Roger S. Pressman

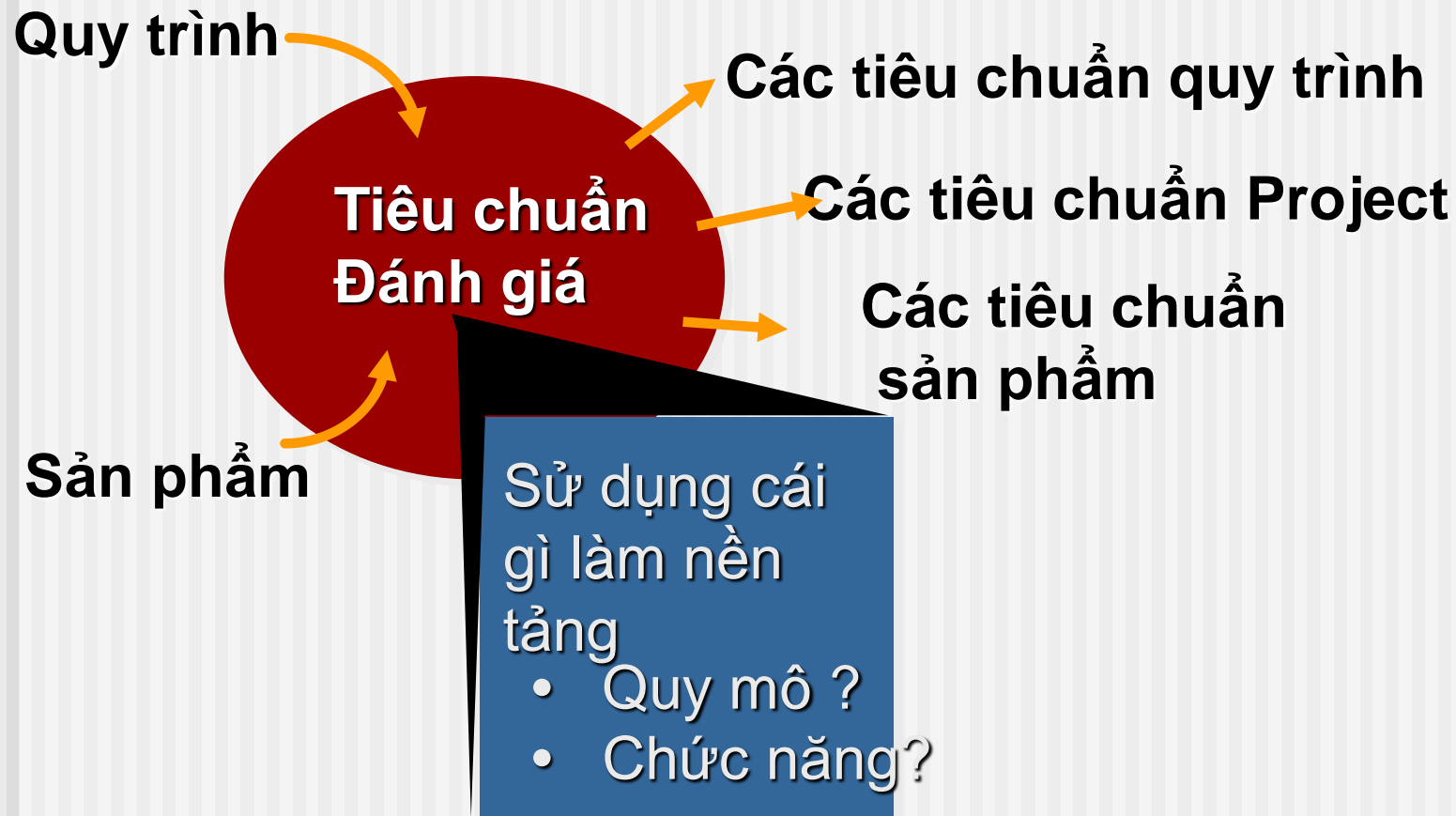
Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman

Chỉ dùng cho mục đích giảng dạy

Chỉ được sao chép cho sinh viên đại học khi sử dụng kết hợp với *Software Engineering: A Practitioner's Approach, 7/e*. Bất kỳ hình thức khác đều bị cấm nếu không có sự cho phép của tác giả.

Tất cả thông tin bản quyền PHẢI xuất hiện trên các slide được đăng tải cho sinh viên.

Các tiêu chuẩn đánh giá một sự quản lí tốt



Vì sao phải đánh giá ?

- Ước lượng trạng thái của một project đang tiến hành
- Theo dõi các rủi ro tiềm tàng
- Phát hiện phạm vi của vấn đề trước khi chúng trở thành mối nguy hiểm
- Điều chỉnh công việc hay các nhiệm vụ
- Đánh giá khả năng của nhóm thực hiện project để điều khiển chất lượng của sản phẩm phần mềm.

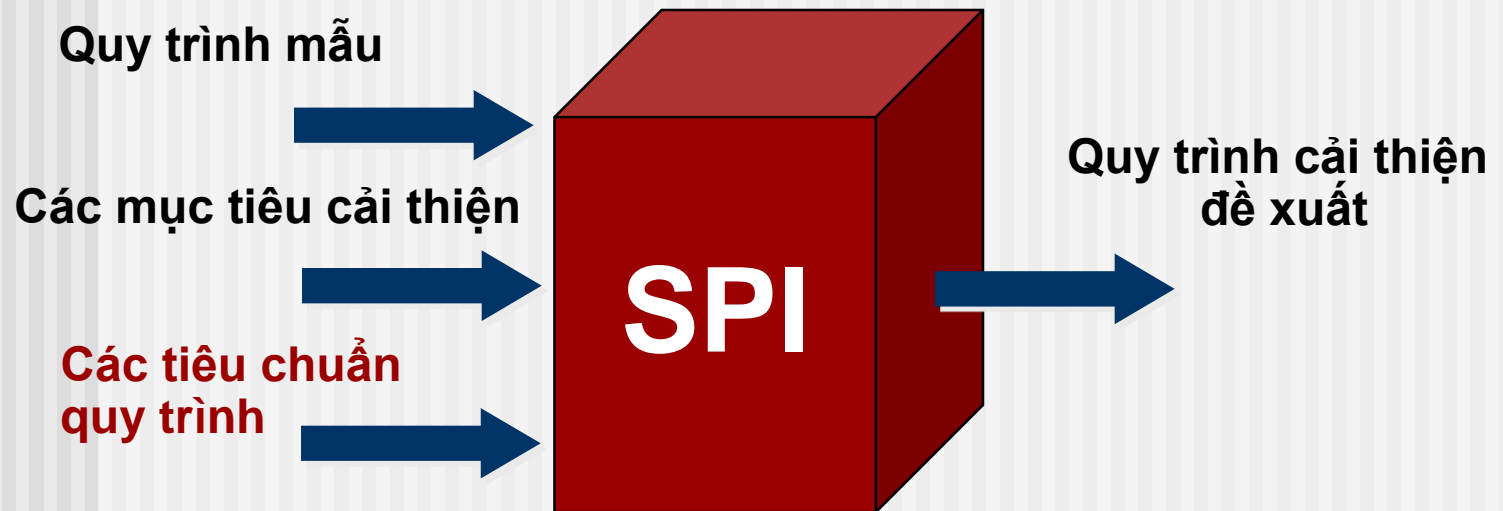
Quy trình đánh giá

- Chúng ta đánh giá sự hiệu quả của một quy trình phần mềm một cách gián tiếp.
 - Điều đó có nghĩa là chúng ta xuất phát một bộ tiêu chuẩn dựa trên các đầu ra mà có thể nhận được từ quy trình.
 - Các đầu ra bao gồm
 - Các đánh giá về lỗi chưa được phát hiện trước khi phát hành phần mềm
 - Các khuyết điểm được thông báo từ người sử dụng cuối
 - Các sản phẩm công việc được chuyển giao
 - Nhân lực sử dụng
 - Thời gian tiêu tốn
 - Hành động đúng theo lịch trình
 - Các tiêu chuẩn khác
- Chúng ta sử dụng tiêu chuẩn đánh giá bằng cách đo lường đặc trưng tiêu biểu của phần mềm.

Các nguyên tắc của tiêu chuẩn đánh giá

- Sử dụng khả năng chung và sự nhạy bén của tổ chức khi đưa ra các dữ liệu tiêu chuẩn.
- Cung cấp các phản hồi thông thường đến các cá nhân và nhóm để đánh giá và các tiêu chuẩn .
- *Không sử dụng các tiêu chuẩn để nhận xét cá nhân*
- Làm việc với những người thi hành và các nhóm để đưa ra các mục tiêu rõ ràng và các tiêu chuẩn sẽ được sử dụng để đánh giá họ.
- *Không bao giờ sử dụng các tiêu chuẩn để đe dọa các cá nhân hay các nhóm.*
- Dữ liệu tiêu chuẩn mà gây ra vấn đề nên được cân nhắc từ chối. Những dữ liệu này chỉ là một dụng cụ để cải thiện quy trình.
- Không quá chú tâm trên một tiêu chuẩn đơn lẻ để bác bỏ các tiêu chuẩn quan trọng khác.

Quy trình cải thiện phần mềm



Các tiêu chuẩn quy trình

- **Chất lượng liên quan**
 - Tập trung vào chất lượng công việc và sản phẩm được chuyển giao
- **Hiệu quả**
 - Sự đánh giá sản xuất liên quan đến nhân lực tiêu tốn
- **Dữ liệu thống kê SQA**
 - Phân loại các lỗi và phân tích
- **Hiệu quả loại bỏ lỗi**
 - Sự phổ biến của các lỗi từ quy trình hành động và hành động
- **Dữ liệu dung lại**
 - Số lượng thành phần được cung cấp và mức độ sử dụng lại của chúng

Các tiêu chuẩn Project

- Được sử dụng để giảm kế hoạch phát triển bằng cách điều chỉnh cần thiết để tránh các chậm trễ, giảm bớt các vấn đề tiềm tàng và các rủi ro
- Được sử dụng để ước định chất lượng sản phẩm trên một nền tảng đang diễn ra và khi cần thiết thay đổi công nghệ tiếp cận để cải thiện chất lượng
- Mỗi Project nên được đánh giá:
 - *Inputs* – các đánh giá về nguồn lực (e.g., people, tools) được yêu cầu để làm việc.
 - *Outputs* – các đánh giá về sự chuyển giao sản phẩm hay công việc được tạo ra trong suốt quá trình thiết kế phần mềm.
 - *Results* – Các đánh giá chỉ ra sự hiệu quả của việc chuyển giao.

Các tiêu chuẩn Project tiêu biểu

- Nhân lực/Thời gian / mỗi nhiệm vụ xây dựng phần mềm
- Các lỗi không được phát hiện / thời gian đánh giá
- Kế hoạch vs triển khai công việc thực tế
- Các thay đổi (số) và đặc điểm của chúng
- Phân bố nhân lực trong các nhiệm vụ xây dựng phần mềm

Các nguyên tắc của tiêu chuẩn đánh giá

- Sử dụng khả năng chung và sự nhạy bén của tổ chức khi đưa ra các dữ liệu tiêu chuẩn.
- Cung cấp các phản hồi thông thường đến các cá nhân và nhóm để đánh giá và các tiêu chuẩn .
- *Không sử dụng các tiêu chuẩn để nhận xét cá nhân*
- Làm việc với những người thi hành và các nhóm để đưa ra các mục tiêu rõ ràng và các tiêu chuẩn sẽ được sử dụng để đánh giá họ.
- *Không bao giờ sử dụng các tiêu chuẩn để đe dọa các cá nhân hay các nhóm.*
- Dữ liệu tiêu chuẩn mà gây ra vấn đề nên được cân nhắc từ chối. Những dữ liệu này chỉ là một dụng cụ để cải thiện quy trình.
- Không quá chú tâm trên một tiêu chuẩn đơn lẻ để bác bỏ các tiêu chuẩn quan trọng khác.

Các tiêu chuẩn Size – Oriented tiêu biểu

- Số lỗi / KLOC (1000 dòng code)
- Số khuyết điểm / KLOC
- \$ / LOC
- Số trang / KLOC
- Số lỗi / 1 người – 1 tháng
- Số lỗi / giờ kiểm tra
- LOC / 1 người – 1 tháng
- \$ / 1 trang

Các tiêu chuẩn Function-Oriented tiêu biểu

- Số lỗi / FP (1000 dòng code)
- Số khuyết điểm / FP
- \$ / FP
- Số trang / FP
- FP / 1 người – 1 tháng

So sánh LOC and FP

Programming Language	LOC per Function point			
	avg.	median	low	high
Ada	154	-	104	205
Assembler	337	315	91	694
C	162	109	33	704
C++	66	53	29	178
COBOL	77	77	14	400
Java	63	53	77	-
JavaScript	58	63	42	75
Perl	60	-	-	-
PL/1	78	67	22	263
Powerbuilder	32	31	11	105
SAS	40	41	33	49
Smalltalk	26	19	10	55
SQL	40	37	7	110
Visual Basic	47	42	16	158

Representative values developed by QSM

Vì sao chọn lựa FP?

- Độc lập với ngôn ngữ lập trình
- Được sử dụng dễ dàng để tính các đặc điểm được xác định sớm trong quy trình phần mềm
- Không phạt các sự thi hành sáng tạo mà sử dụng ít LOC hơn các phiên bản khác
- Giúp cho việc đánh giá tác động các thành phần tái sử dụng dễ dàng hơn

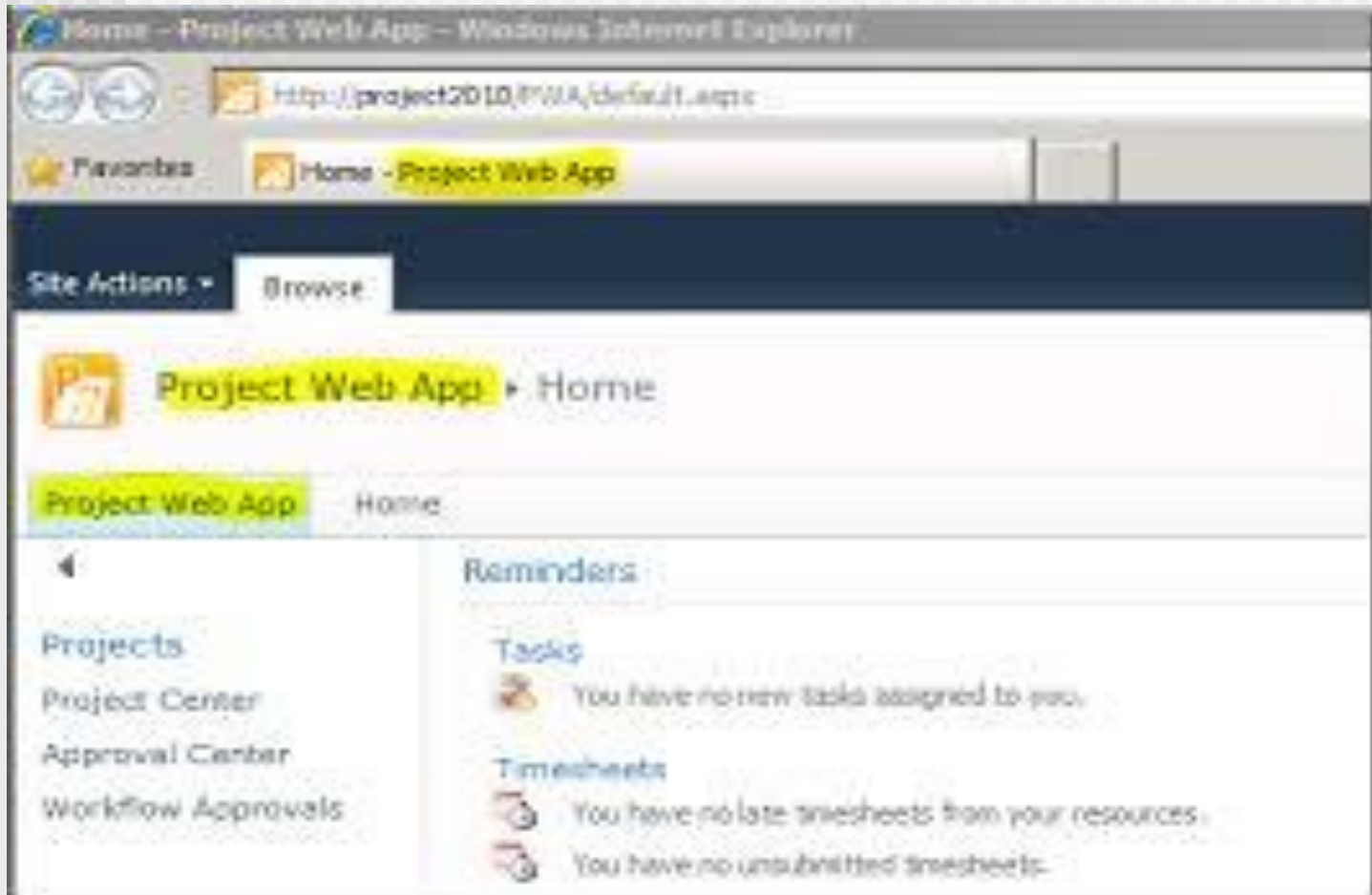
Các tiêu chuẩn Object-Oriented

- Số kịch bản sử dụng (use-cases)
- Số lớp hỗ trợ (được yêu cầu để thi hành hệ thống nhưng không quan hệ ngay lập tức đến miền vấn đề)
- Trung bình số lớp hỗ trợ / lớp khóa (lớp phân tích)
- Số hệ thống con

Các tiêu chuẩn WebApp Project

- Số trang Web tĩnh (người sử dụng cuối cùng không điều khiển nội dung hiển thị trên trang)
- Số trang Web động (kết quả hành động của người sử dụng cuối được hiển thị trên trang)
- Số trang liên kết nội bộ (trang liên kết nội bộ là các con trỏ cung cấp hyperlink đến vài trang Web khác trong WebApp)
- Số các đối tượng dữ liệu bền vững
- Số hệ thống giao tiếp bên ngoài
- Số các đối tượng nội dung tĩnh
- Số các đối tượng nội dung động
- Số các chức năng thực thi

webproject



Đánh giá chất lượng

- **Sự đúng đắn** — Mức độ mà chương trình hoạt động đúng theo đặc điểm kĩ thuật yêu cầu
- **Tính duy trì** — Mức độ chương trình tuân theo sự thay đổi
- **Tính toàn vẹn** — Mức độ chương trình có thể chống lại các tấn công từ bên ngoài
- **Khả năng sử dụng** — Mức độ dễ sử dụng của chương trình

Sử chữa các nhược điểm hiệu quả

$$DRE = E / (E + D)$$

Với:

E là số lượng lỗi được phát hiện trước khi phần mềm đến người sử dụng cuối

D là số lỗi được phát hiện sau khi phần mềm đã được giao.

Các tiêu chuẩn cho những tổ chức nhỏ

- Thời gian (giờ hoặc ngày) kể từ khi yêu cầu bắt đầu được thực hiện cho đến khi hoàn thành, t_{queue} .
- Nhân lực (người – giờ) để thực hiện đánh giá, W_{eval} .
- Thời gian (giờ hoặc ngày) kể từ khi hoàn thành đến khi phân công các yêu cầu mới, t_{eval} .
- Nhân lực (người – giờ) được yêu cầu để thay đổi, W_{change} .
- Thời gian (giờ hoặc ngày) để thực hiện thay đổi, t_{change} .
- Các lỗi không được phát hiện trong suốt công việc để thay đổi, E_{change} .
- Các nhược điểm không được phát hiện sau khi thay đổi được phát hành đến người sử dụng, D_{change} .

Xây dựng tiêu chuẩn chương trình

- Xác định mục tiêu công việc của bạn.
- Xác định những thứ bạn muốn biết hoặc học.
- Xác định các mục tiêu nhỏ.
- Xác định các thực thể và thuộc tính liên quan đến mục tiêu của bạn.
- Lập các mục tiêu đánh giá.
- Xác định các câu hỏi định lượng và các chỉ số liên quan mà bạn sẽ sử dụng để đánh giá các mục tiêu.
- Xác định các yếu tố dữ liệu mà bạn sẽ thu thập để xây dựng các câu trả lời cho câu hỏi của bạn.
- Xác định các biện pháp được sử dụng và định nghĩa các hoạt động
- Xác định các hoạt động mà bạn sẽ thực hiện để thi hành đánh giá.
- Chuẩn bị một kế hoạch thực hiện các tiêu đánh giá.