

Chapter 1

■ Software & Software Engineering

Slide được xây dựng theo cuốn

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

by Roger S. Pressman

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

Chỉ dành cho mục đích giáo dục phi lợi nhuận.

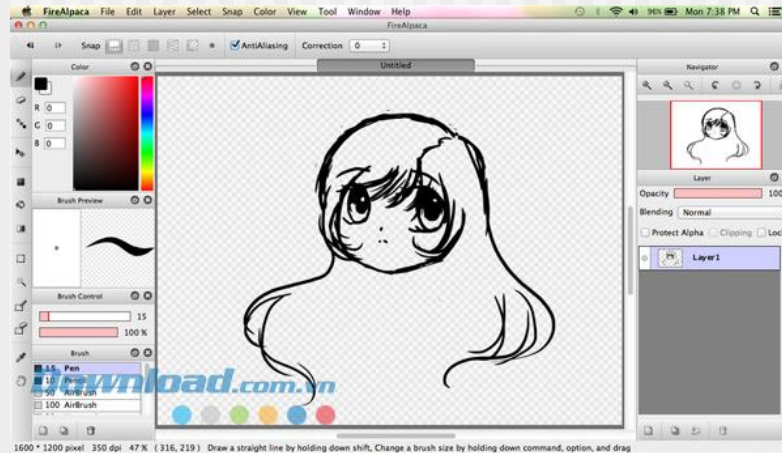
Sinh viên đại học có thể sao chép slide này nếu dùng chung với cuốn

Software Engineering: A Practitioner's Approach, 7/e. Mọi hình vi tái bản hoặc sử dụng đều bị cấm khi chưa có sự cho phép của tác giả.

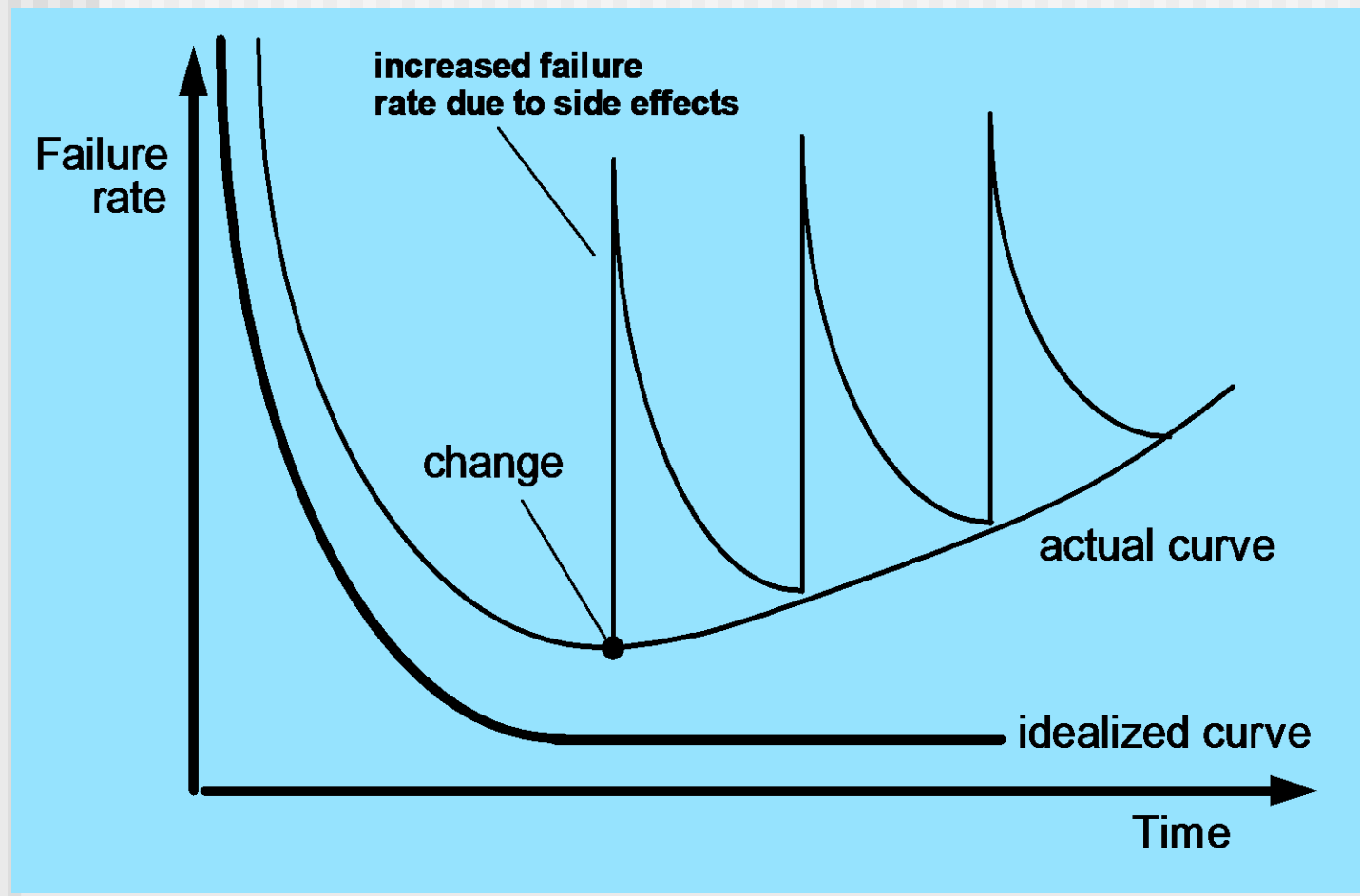
Tất cả các thông tin bản quyền phải xuất hiện trong các slide nếu xuất hiện trên trang web dành cho sinh viên.

Thế nào là phần mềm

Phần mềm gồm:(1) hướng dẫn (các chương trình máy tính) mà khi thực hiện cung cấp các tính năng mong muốn , chức năng , và hiệu suất ;(2) cấu trúc dữ liệu cho phép các chương trình vận dụng đầy đủ thông tin và (3) tài liệu mô tả các hoạt động và sử dụng các chương trình .



Wear vs. Deterioration



These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

Các loại phần mềm

- Phần mềm hệ thống
- Phần mềm ứng dụng
- Phần mềm kỹ thuật / khoa học
- Phần mềm nhúng
- Sản phẩm phần mềm trực tuyến
- Ứng dụng web (ứng dụng web)
- Phần mềm AI



Các loại phần mềm mới

- **Open world computing**—phổ biến ,phân phối rộng khắp
- **Ubiquitous computing**— mạng lưới không dây
- **Netsourcing**—Web như một công cụ tính toán
- Open source – miễn phí mã nguồn cho cộng đồng,nhiều cơ hội nhưng chứa nhiều rủi ro
- Ngoài ra ...
 - **Khai thác dữ liệu**
 - **Tính toán mạng lưới**
 - **Máy nhận thức**
 - **Phần mềm công nghệ nano**

Legacy Software

Tại sao cần thay đổi

- Phần mềm phải được điều chỉnh để đáp ứng nhu cầu của môi trường hoặc công nghệ mới
- Phần mềm phải được thay đổi để thực hiện các yêu cầu kinh doanh mới. phần mềm phải được mở rộng để làm cho nó tương thích với các hệ thống hiện đại hơn hoặc cơ sở dữ liệu khác . phần mềm phải được cấu trúc lại để làm cho nó khả thi trong một môi trường mạng .

Đặc điểm của các ứng dụng Web

- **Mạng intensiveness:** Một ứng dụng web nằm trong mạng lưới và phải phục vụ nhu cầu của một cộng đồng đa dạng của khách hàng .
- **Đồng thời:** Một số lượng lớn người dùng có thể truy cập các ứng dụng web cùng một thời gian .
- **Tải không thể đoán trước:** Số lượng người dùng của ứng dụng có thể thay đổi tùy theo số lượng truy cập từ ngày này sang ngày khác .
- **Hiệu suất:** Nếu một người dùng WebApp phải chờ đợi quá lâu (để truy cập , xử lý phía máy chủ , để định dạng client -side và hiển thị) , anh ta hoặc cô ta có thể quyết định đi nơi khác .
- **Tình trạng sẵn có:** Mặc dù kỳ vọng của 100 phần trăm khả năng không hợp lý , người sử dụng các ứng dụng web phổ biến thường được yêu cầu truy cập vào một cơ sở " 24/7/365 "

Đặc điểm của các ứng dụng Web

- **Hướng dữ liệu:** Các chức năng chính của nhiều ứng dụng web là sử dụng hypermedia để trình bày văn bản, đồ họa, âm thanh và nội dung video đến người dùng cuối.
- **Cảm quan nội dung:** Chất lượng và tính chất thẩm mỹ của nội dung vẫn là một yếu tố quyết định quan trọng về chất lượng của một WebApp
- **Tiếp tục phát triển:** Không giống như các phần mềm ứng dụng thông thường cho rằng cập nhật theo một loạt các kế hoạch, phát hành thứ tự thời gian, khoảng cách, các ứng dụng web phát triển liên tục.
- **An ninh:** Bởi vì các ứng dụng web có sẵn thông qua truy cập mạng, nói là khó khăn, nếu không phải không thể, để hạn chế số lượng của người dùng cuối cùng những người có thể truy cập vào các ứng dụng.
- **Thẩm mỹ:** Một phần không thể phủ nhận sự hấp dẫn của một WebApp là cái giao diện và nội dung của nó

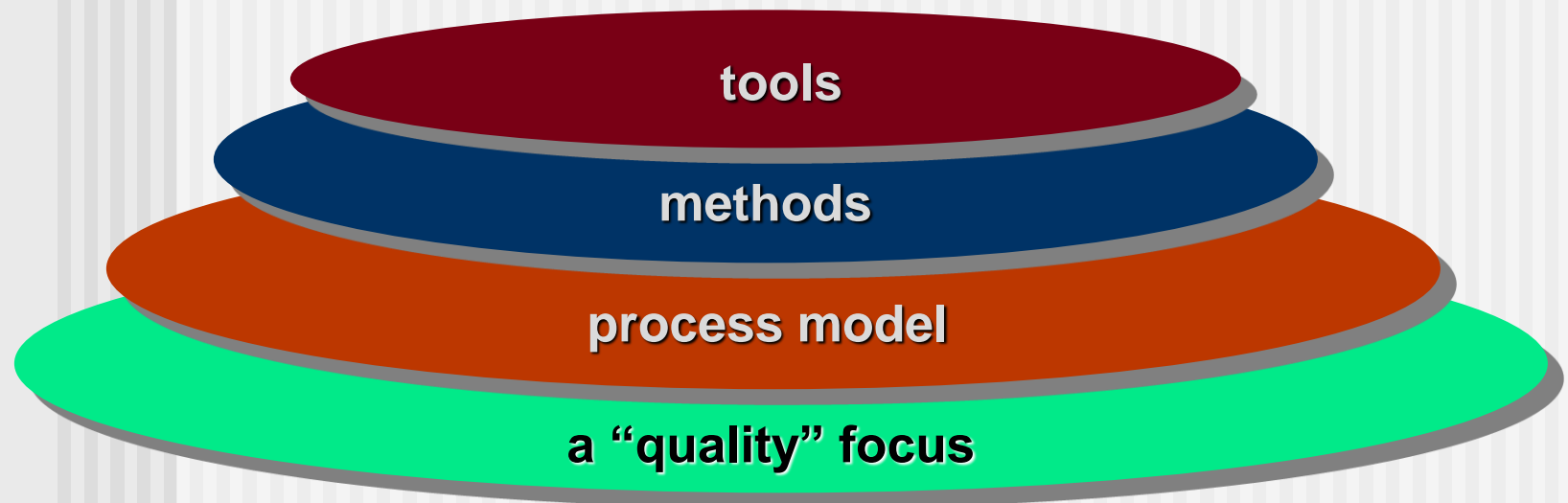
Kĩ nghệ phần mềm

- Một số thực tế : nỗ lực phối hợp nên được thực hiện để hiểu các vấn đề trước khi một giải pháp phần mềm được phát triển
- Thiết kế trở thành một phần quan trọng phần mềm
- Giới thiệu phần mềm chất lượng cao
- Phần mềm nên được duy trì

Kĩ nghệ phần mềm

- Định nghĩa IEEE : Kỹ thuật phần mềm :
 - (1) Việc áp dụng , xử lý kỷ luật , phương pháp tiếp cận có hệ thống định lượng cho sự phát triển , vận hành và bảo trì phần mềm ; đó là , việc áp dụng các kỹ thuật phần mềm.
 - (2) Các nghiên cứu về phương pháp tiếp cận như trong (1)

A Layered Technology



Software Engineering

Khung chương trình

Khuôn khổ quá trình

Khung hoạt động

- . Nhiệm vụ công việc
- . Sản phẩm công việc
- . Cột mốc & phân phối
- . Điểm kiểm tra bảo đảm chất lượng

Umbrella Activities

Khung hoạt động

- Truyền thông
- Kế hoạch
- Mẫu
 - Phân tích các yêu cầu
 - Thiết kế
- Xây dựng
 - Mã số thể hệ
 - Thử nghiệm
- Triển khai

Umbrella Activities

- Quản lý dự án phần mềm
- Đánh giá kỹ thuật chính thức
- Phần mềm đảm bảo chất lượng
- Quản lý cấu hình phần mềm
- Chuẩn bị sản phẩm công việc và sản xuất
- Quản lý có thể dùng lại sự đo lường
- Quản lý rủi ro

Thích ứng với quy trình mẫu

- Các dòng chảy chung của các hoạt động , hành động , và các nhiệm vụ và phụ thuộc lẫn nhau giữa chúng
- Mức độ mà hành động và nhiệm vụ được quy định trong từng hoạt động khung
- Mức độ mà các sản phẩm công việc được xác định và yêu cầu
- Cách thức mà các hoạt động đảm bảo chất lượng đang áp dụng
- Cách thức mà các hoạt động theo dõi và kiểm soát dự án được áp dụng
- Mức độ tổng thể của chi tiết và chặt chẽ với quá trình đó được mô tả
- Mức độ mà các khách hàng và các bên liên quan khác có liên quan đến dự án
- Mức độ tự chủ cho các nhóm phần mềm mức độ mà tổ chức đội ngũ và vai trò được quy định

Bản chất thực nghiệm

- Polya suggests:
 - 1. Hiểu được vấn đề (truyền thông và phân tích) .
 - 2. Kế hoạch một giải pháp (mô hình hóa và thiết kế phần mềm) .
 - 3. Thực hiện các kế hoạch (hệ mã) .
 - 4. Kiểm tra các kết quả cho chính xác (thử nghiệm và đảm bảo chất lượng) .

Hiểu được vấn đề

- Ai có phần trong giải pháp cho vấn đề này? các bên liên quan là ai?
- Các ẩn số là gì ? Những dữ liệu , chức năng , và các tính năng được yêu cầu để giải quyết đúng đắn các vấn đề?
- Vấn đề có thể được chia ra ? Là nó có thể để đại diện cho vấn đề nhỏ hơn có thể được dễ dàng hơn để hiểu ?
- Vấn đề có thể được biểu diễn bằng đồ thị ? Một mô hình phân tích có thể được tạo ra?

Lập phương án giải quyết

- Bạn đã thấy vấn đề tương tự trước đây ? Có mô hình đó được nhận biết trong một giải pháp tiềm năng? Có phần mềm hiện có mà thực hiện các dữ liệu , chức năng , và các tính năng được yêu cầu ?
- Có một vấn đề tương tự được giải quyết ? Nếu vậy , những yếu tố của các thể tái sử dụng giải pháp?
- Bài toán có thể được xác định? Nếu vậy , những giải pháp sẵn sàng rõ ràng cho các bài toán ?
- Bạn có thể đại diện cho một giải pháp một cách dẫn đến việc thực hiện có hiệu quả ? Một mô hình thiết kế có thể được tạo ra?

Đề ra kế hoạch

- Có các giải pháp phù hợp với kế hoạch? Là mã nguồn để theo dõi các mô hình thiết kế? Là mỗi bộ phận cấu thành của các giải pháp có lẽ là đúng ?
- Đã thiết kế và mã được xem xét , hoặc tốt hơn , đã chứng minh tính đúng đắn được áp dụng cho các thuật toán ?

Kiểm tra kết quả

- Có thể kiểm tra từng bộ phận cấu thành của giải pháp? Có một chiến lược thử nghiệm hợp lý được thực hiện?
- Liệu kết quả giải pháp sản xuất cho phù hợp với các dữ liệu , chức năng , và các tính năng được yêu cầu ? Có những phần mềm được xác nhận đối với tất cả các yêu cầu của các bên liên quan ?

Nguyên tắc Hoker

- 1 : Lý do It mọi thứ tồn tại
- 2 : KISS (giữ đơn giản!)
- 3 : Duy trì tầm nhìn
- 4 : Những gì bạn sản xuất ,tiêu thụ
- 5 : Hãy lên kế hoạch tương lai
- 6 : Hoạch định cho tái sử dụng
- 7 : Hãy suy nghĩ !

Software Myths

- Affect managers, customers (and other non-technical stakeholders) and practitioners
- Are believable because they often have elements of truth,
but ...
- Invariably lead to bad decisions,
therefore ...
- Insist on reality as you navigate your way through software engineering

How It all Starts

■ *SafeHome:*

- Every software project is precipitated by some business need—
 - the need to correct a defect in an existing application;
 - the need to the need to adapt a ‘legacy system’ to a changing business environment;
 - the need to extend the functions and features of an existing application, or
 - the need to create a new product, service, or system.