



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

# NHẬP MÔN CNPM

## Nội dung / Chương 1: Giới thiệu chung

Thông tin GV

# Nội dung chương

Phần 1. ABC

Phần 2. ABC

Phần 3. ABC

Phần 4. ABC

Phần 5. ABC

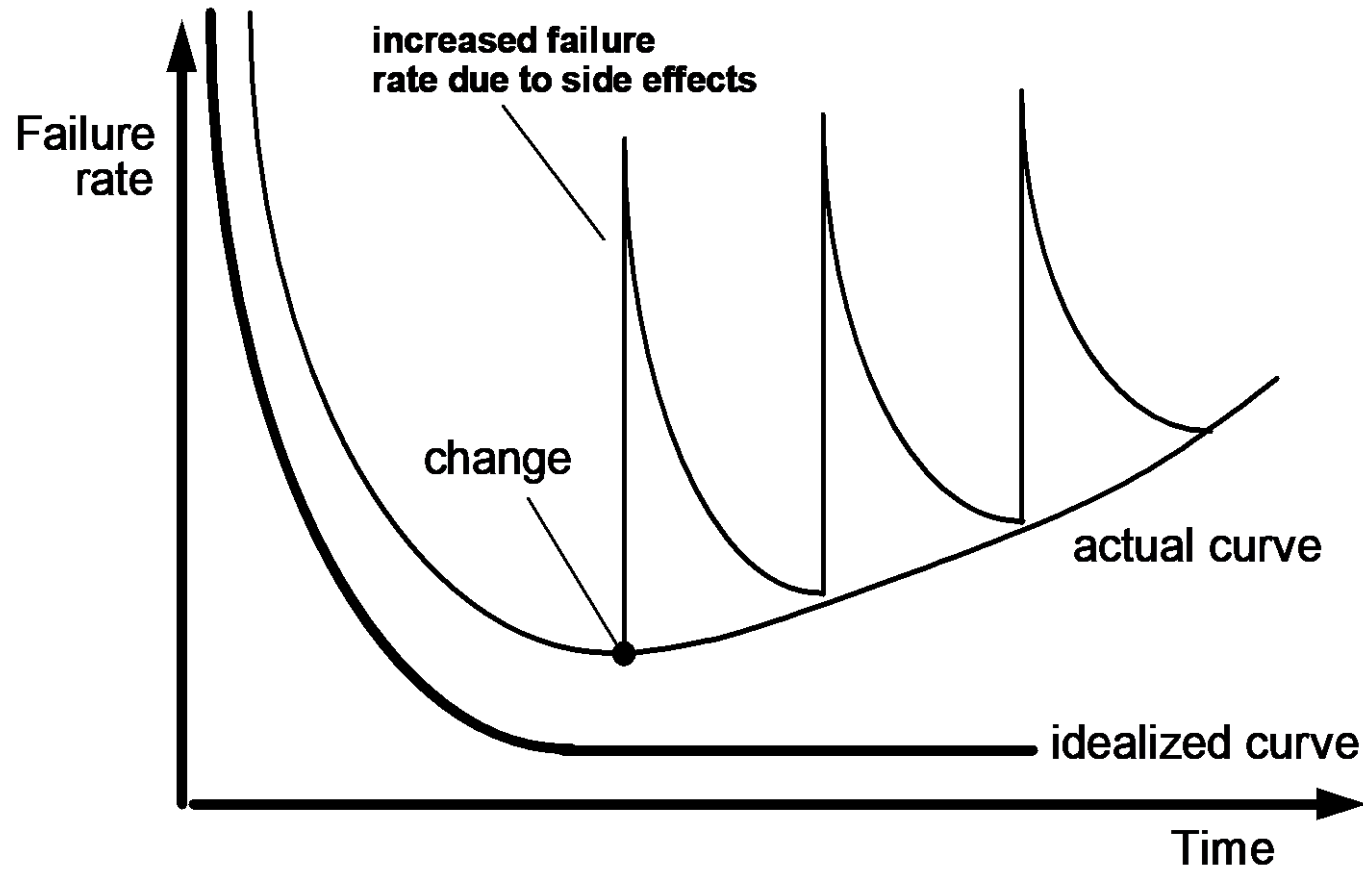
Phần 6. ABC

# Thế nào là phần mềm

- Phần mềm gồm: (1) **Các lệnh** (các chương trình máy tính) khi chạy sẽ cung cấp các tính năng, chức năng, và vận hành mong muốn; (2) **Cấu trúc dữ liệu** cho phép các chương trình xử lý thông tin trọn vẹn và (3) **Tài liệu** mô tả các hoạt động và sử dụng các chương trình.



# Sử dụng & Hỏng hóc



# 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 rộng khắp
- **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

# Phần mềm di sản

## *Tại sao cần thay đổi*

- Việc quản lý dự án đã có từ thời xưa: trong chiến tranh, xây dựng Kim tự tháp và các kỳ quan thế giới...
- 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 .
- **Tính sẵn dùng**: 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

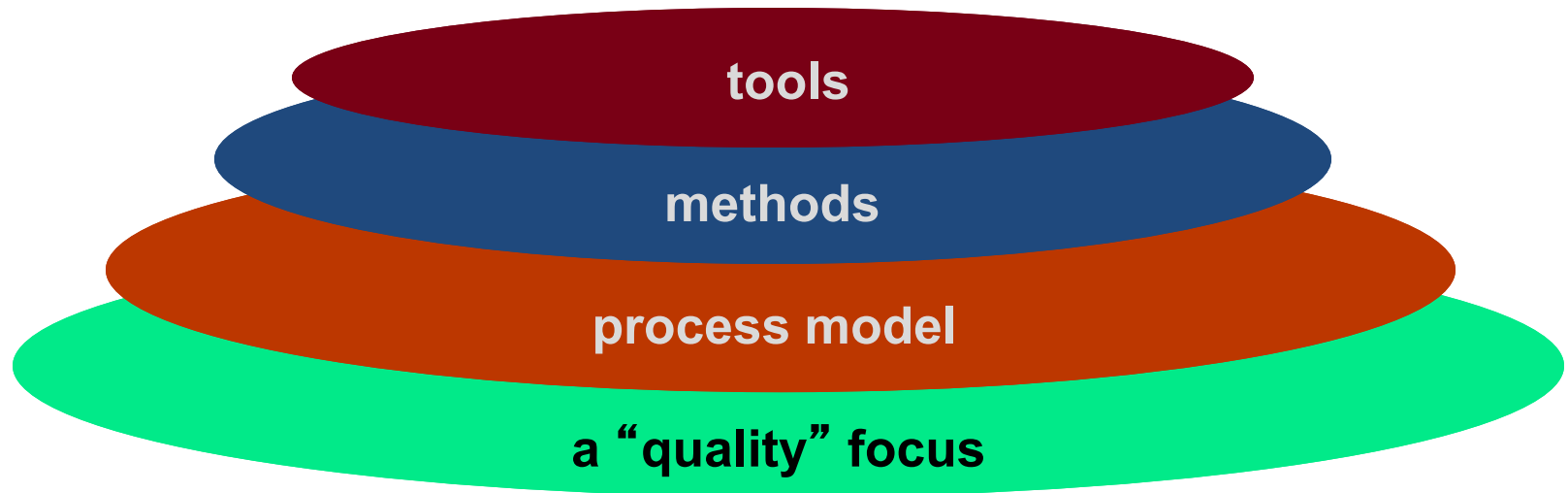
# Kĩ nghệ phần mềm

- Một số thực tế :
  - Cần nỗ lực hiểu các vấn đề trước khi một giải pháp phần mềm được phát triển
  - Thiết kế trở thành hoạt động then chốt
  - Phần mềm cần thể hiện có chất lượng cao
  - Phần mềm cần được bảo trì
- Định nghĩa khoa học về SE:
  - *[Software engineering is] the establishment and use of **sound engineering principles** in order to obtain **economically** software that is **reliable and works efficiently** on **real machines**.*

# Kỹ nghệ phần mềm

- Định nghĩa IEEE :
  - Kỹ nghệ phần mềm : (1) Việc áp dụng một cách *Tiếp cận định lượng, nguyên tắc, hệ thống* để *Phát triển, Vận hành và Bảo trì* phần mềm; có nghĩa là , Áp dụng kỹ nghệ vào phần mềm. (2) Các nghiên cứu về các Cách tiếp cận được nêu trong (1)
- The IEEE definition:
  - Software Engineering: (1) The application of a *systematic, disciplined, quantifiable approach* to the *development, operation, and maintenance* of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

# A Layered Technology



*Software Engineering*

# Khuôn khổ quy trình

- Khuôn khổ quy trình
  - Khuôn khổ hoạt động
    - Nhiệm vụ công việc
    - Sản phẩm công việc
    - Cột mốc & phân phối
    - Điểm kiểm tra bảo đảm chất lượng
- Umbrella Activities

# Khuôn khổ 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 mẫu quy trình

- 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

- Kết quả bàn giao có thể là ít hữu hình
- 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 kế hoạch cho giải pháp

- **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 ?

# Các nguyên tắc chung của Hoker

1. Có Lý do cho mọi thứ tồn tại
2. KISS (giữ đơn giản!)
3. Duy trì tầm nhìn
4. Bạn sản xuất gì, người khác tiêu thụ nấy
5. Hãy mở rộng cho tương lai
6. Lập kế hoạch hướng tới 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.



# Tài liệu tham khảo

- 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
- <http://bit.ly/2ihOLB4>