CÔNG NGHỆ PHẦN MỀM

SOFTWARE ENGINEERING

TS. Võ Đức Hoàng

Khoa Công nghệ thông tin

Trường Đại học Bách khoa - Đại học Đà Nẵng

Mục đích

- Hiểu và nắm bắt
  - Khái niệm công nghệ phần mềm
  - Các mô hình phát triển phần mềm
  - Các hoạt động phát triển phần mềm
  - Các kỹ thuật và phương pháp cơ bản trong phát triển phần mềm
- Áp dụng công nghệ phần mềm trong phát triển phần mềm

Đánh giá kết quả

- Giữa kỳ (20%)
  - Thuyết trình
- Bài tập (20%)
  - Thuyết trình
  - Chuyên cần
- Cuối kỳ (60%)
  - Trắc nghiệm

Nội dung

- Chương 1: Giới thiệu Công nghệ phần mềm
- Chương 2: Các mô hình phát triển phần mềm
- Chương 3: Phân tích và đặc tả yêu cầu
- Chương 4: Các kỹ thuật đặc tả
- Chương 5: Thiết kế
- Chương 6: Lập trình và ngôn ngữ lập trình
- Chương 7: Kiểm thử
- Chương 8: Quản trị dự án

Tài liệu học tập

- ***Giáo trình chính:***
  - Lê Đức Trung, Công nghệ phần mềm, NXB KHKT, 2001.
  - Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2016
- ***Tài liệu tham khảo:***
  - Ronald Leach, Introduction to Software Engineering, CRC Press, 1999.
  - Roger S. Pressman, Software Engineering : a practitioner's approach, 7th Edition, The McGraw-Hill Companies, Inc

Tổng quan về Công nghệ phần mềm (1)

Nội dung

- Lịch sử phát triển phần mềm và khủng hoảng phần mềm
- Phân loại phần mềm
- Chất lượng phần mềm
- Đặc trưng của phần mềm
- Công nghệ phần mềm
  - Khái niệm
  - Mục đích
  - Nguyên tắc
  - Đặc trưng

Lịch sử phát triển phần mềm

- 1946, máy tính điện tử ra đời
- 1950, máy tính được thương mại hóa
  - Phần mềm bắt đầu được phát triển
- Những năm 1960
  - Những thất bại về phát triển phần mềm
  - Sản phẩm phần mềm phức tạp
  - Nhiều lỗi
  - Tổ chức sản xuất: giá thành, tiến độ, ...
- Người ta nói đến “Khủng hoảng phần mềm”

Lịch sử phát triển phần mềm

- Từ thủ công đến công nghệ

- Chương trình nhỏ
  - không chuyên nghiệp
  - 1 người làm
  - Người sử dụng = người phát triển
  - 1 sản phẩm = mã nguồn
  - Tiến trình phát triển đơn giản

- Dự án lớn
  - chuyên nghiệp
  - Nhiều người làm
  - khách hàng & nhà cung cấp
  - Nhiều sản phẩm
  - Tiến trình phát triển phức tạp

- Năm 1968, hội thảo đầu tiên về “Công nghệ phần mềm”

Khủng hoảng phần mềm

- Về mặt sản phẩm
  - Chất lượng sản phẩm phần mềm
    - Không đáp ứng yêu cầu thực tế
    - Khó sử dụng
    - Không tin cậy
    - Khó bảo trì

=> Khách hàng không hài lòng

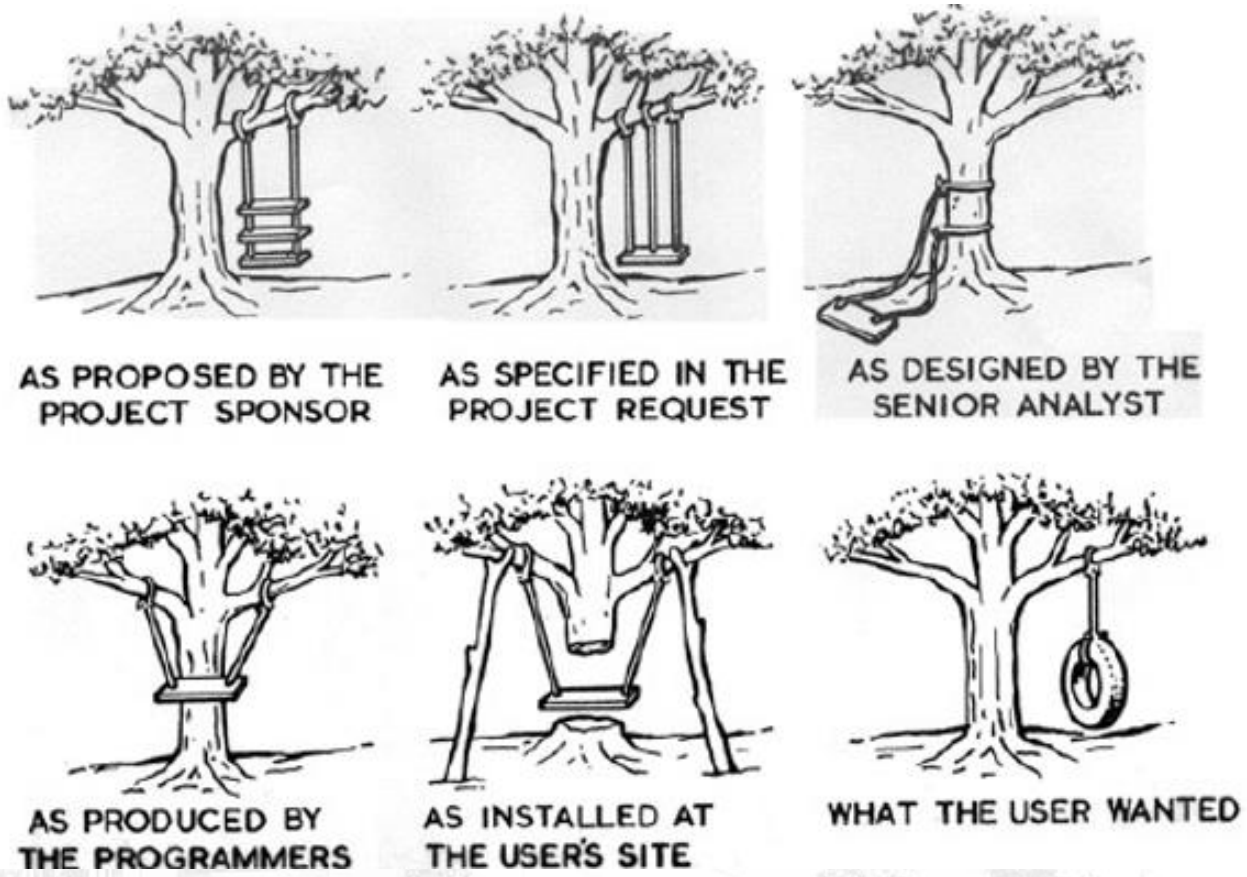
Khủng hoảng phần mềm

- Về mặt quản lý
  - Kế hoạch
    - Không đánh giá đúng giá thành
    - Không đúng tiến độ
    - Chi phí phát triển/chi phí bảo trì
  - Về mặt pháp lý
    - Hợp đồng không rõ ràng, không chặt chẽ
  - Nhân lực
    - Đào tạo
    - Giao tiếp
  - Thiếu tiêu chuẩn đánh giá sản phẩm
  - Thiếu qui trình quản lý

Khủng hoảng phần mềm

- Điều tra của General Acounting Office (1982) trên nhiều dự án với tổng vốn đầu tư $68.000.000
  - Không giao sản phẩm: 29%
  - Không được sử dụng: 47%
  - Bỏ cuộc: 19%
  - Được sử dụng sau khi đã chỉnh sửa: 3%
  - Tốt: 2%

Khủng hoảng phần mềm



Khó khăn và thách thức

- Phát triển phần mềm thực tế là lĩnh vực đầy khó khăn, thách thức (rủi ro cao):
- Nhiều dự án lớn thất bại
- Chi phí phát triển cao
- Không đạt được mục đích

(không được đưa vào sử dụng, hiệu quả thấp)

Ví dụ: các dự án thất bại

- OS 360
  - > 1M dòng lệnh
  - Từ 1963->1966 (gấp đôi dự kiến)
  - 5000 nhân công
  - 200 M USD

=> (IBM 370) 7M dòng lệnh

Ví dụ: tổn thất do lỗi phần mềm

- 1978: vệ tinh phóng lên sao Kim bị hỏng do lỗi của phần mềm
  - Lỗi của câu lệnh FOR (Fortran)
- 1996: vệ tinh Ariane 5 hỏng do lỗi phần mềm gây thiệt hại 500M$
  - Lỗi phép toán số thực (Ada)

Bản chất của vấn đề

- Tính phức tạp là bản chất của phần mềm
- Yêu cầu sử dụng phần mềm không ngừng thay đổi
- Sự tiến bộ nhanh của phần mềm và phần cứng (hạ tầng -phần nền): thay đổi

=> Yêu cầu tiến hóa phần mềm là tất yếu

Tiến hóa của phần mềm

- Chi phí cho phần mềm cao
- Phần mềm trở thành ngành công nghiệp khổng lồ
  - Phí phát triển OS 360 (1963~1966) : **200M$**
  - Chi phí phần mềm năm 1985 : **70B$**
  - Chi phí cho phần mềm năm 2000 : **770B$**

  (mức tăng 12%/năm)
- Năng suất lập trình vẫn thấp
  - Phát triển phần mềm mang tính thủ công
  - Giá thành cao

Thách thức đối với phần mềm

- Khả năng xây dựng phần mềm không đáp ứng kịp nhu cầu tăng nhanh (có internet, mọi lĩnh vực xã hội)
- Qui mô và độ phức tạp ngày càng tăng khiến chi phí phát triển, bảo trì càng ngày càng tốn kém
- Sự tinh vi của phần cứng vượt xa khả năng tạo ra phần mềm để khai thác nó

=> **Cần có những phương pháp, công cụ hiện đại để phát triển phần mềm**

Khó khăn của phát triển phần mềm

- "***Khó sản xuất được phần mềm*** **có chất lượng theo đúng lịch trình & kinh phí cho trước"**
- **Phần mềm là phần tử logic: không kiểm soát được theo phương pháp thông thường**
  - **trong sản xuất** (*rủi ro, tính thủ công*)
  - **trong bảo trì** (*lớn, phức tạp, thay đổi nhanh*)
  - **trong kiểm soát chất lượng** (*làm thủ công, nhiều người, nhiều công đoạn*)

Định nghĩa về phần mềm

- Phần mềm bao gồm:
  - **Chương trình máy tính**
    - Phần vận hành được: mã máy
    - Không vận hành: mã nguồn
  - **Các cấu trúc dữ liệu**
    - Cấu trúc làm việc (bộ nhớ trong)
    - Cấu trúc lưu trữ
  - **Tài liệu sử dụng**
    - Hướng dẫn sử dụng
    - Tham khảo kỹ thuật
    - Tài liệu đặc tả, phân tích, thiết kế, kiểm thử

Phân loại phần mềm (*trên mục đích sử dụng*)

- **Phần mềm hệ thống**
  - Tập các chương trình phục vụ cho các chương trình
  - Tương tác trực tiếp với phần cứng
  - Phục vụ nhiều người dùng
- **Phần mềm thời gian thực**
  - Thu thập, xử lí các dữ kiện thế giới thực
  - Đáp ứng yêu cầu chặt chẽ về thời gian
    - kiểm soát, điều khiển
    - điều phối
    - thu thập dữ liệu
    - phân tích dữ liệu

Phân loại phần mềm (tt)

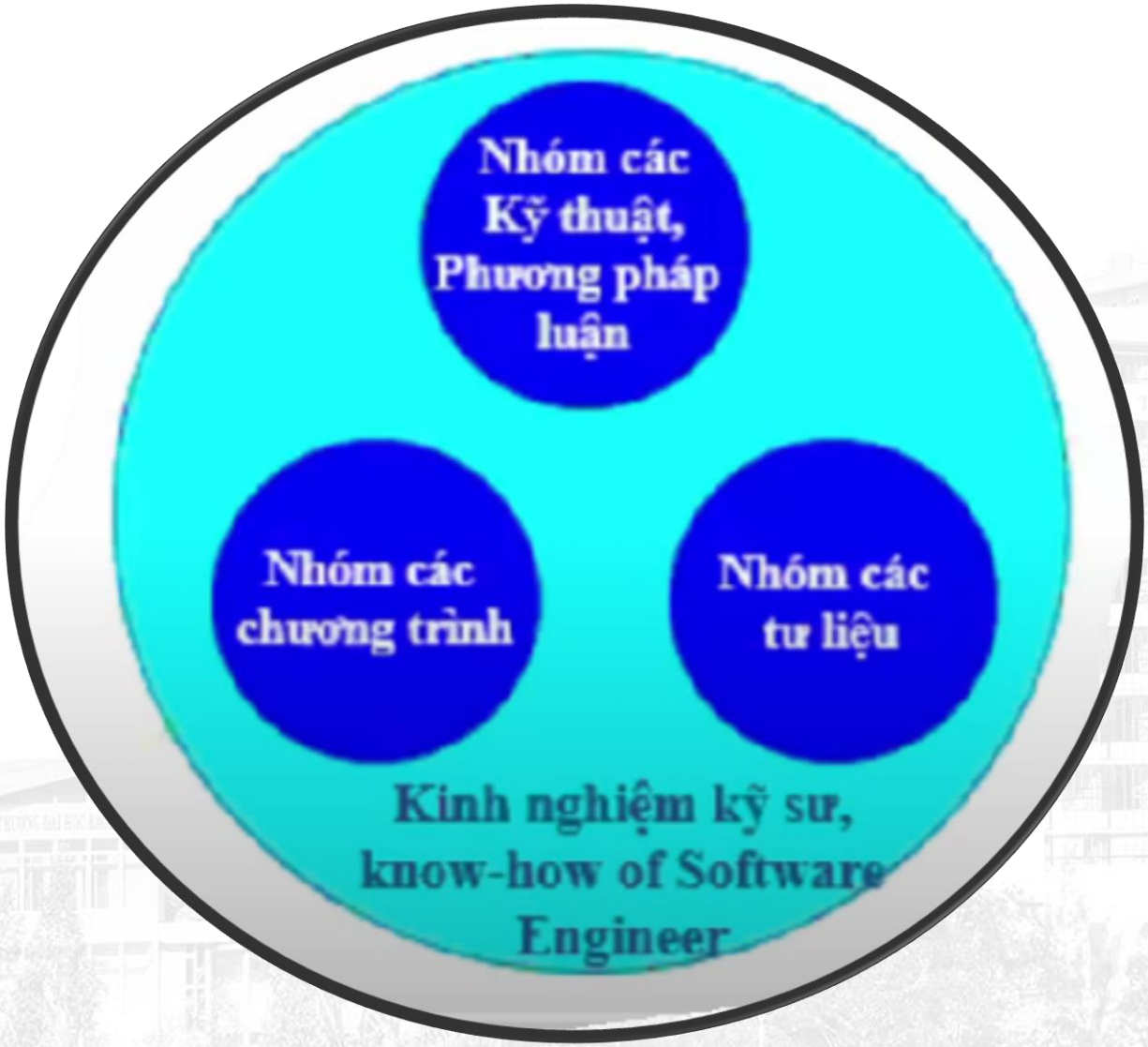
- **Phần mềm nghiệp vụ**
  - Xử lí các thông tin nghiệp vụ, thường gắn với CSDL
  - Xử lí các giao tác (mạng máy tính bán hàng...)
  - Lĩnh vực ứng dụng rất lớn
- **Phần mềm khoa học kỹ thuật**
  - Đặc trưng bởi thuật toán (tính toán vật lí, mô phỏng)
  - Đòi hỏi năng lực tính toán cao
- **Phần mềm nhúng (embeded software)**
  - Chỉ đọc ra khi thiết bị khởi động,
  - Thực hiện chức năng hạn chế (điều khiển sản phẩm)
  - Là sự kết hợp giữa hệ thống và thời gian thực

Phân loại phần mềm (tt)

- **Phần mềm máy tính cá nhân**
  - Các bài toán nghiệp vụ nhỏ(ứng dụng văn phòng)
  - Giao diện đồ họa phát triển
  - Có nhu cầu rất cao
- **Phần mềm trí tuệ nhân tạo**
  - Dùng các thuật toán phi số (logic): suy luận, tìm kiếm
  - Hệ chuyên gia, nhận dạng, trò chơi...
- **Phần mềm công cụ cho Công nghệ phần mềm**
  - Compiler, các công cụ CASE...

Phân biệt SOFTWARE và PROGRAMS

SOFTWARE


Nhóm các Kỹ thuật, Phương pháp luận
Nhóm các chương trình
Nhóm các tư liệu
Kinh nghiệm kỹ sư, know-how of Software Engineer

Phân loại phần mềm *(trên quan điểm của người phát triển)*

- Sản phẩm đặt hàng
  - Sản xuất theo đơn đặt hàng (HTTT quản lý...)
  - Đơn chiếc, yêu cầu đặc thù
- Sản phẩm chung
  - Bán rộng rãi
  - Thỏa mãn yêu cầu chung của số lớn người dùng

=> Mỗi loại có cách thức tiếp cận riêng, nhất là ở các bước phân tích, bảo trì

Chất lượng phần mềm

- Tính đúng đắn (correctness)
  - Thực hiện đúng các đặc tả về chức năng (functional specification)
  - Thuật toán đúng
- Tính tin cậy (reliability)
  - Đáp ứng được những yêu cầu đặt ra
- Tính bền vững (robustness)
  - Hoạt động tốt trong những điều kiện sử dụng khác nhau

Chất lượng phần mềm

- Tính hiệu quả (efficiency)
  - Sử dụng hiệu quả các nguồn tài nguyên (bộ nhớ, CPU...)
- Tính thân thiện (user friendlyness)
  - Dễ sử dụng
- Tính dễ kiểm tra (verifiability)
  - Dễ kiểm tra chất lượng

Chất lượng phần mềm

- Tính dễ bảo trì (maintainability)
  - Dễ xác định và sửa lỗi
  - Dễ tạo phiên bản mới khi có sự mở rộng
- Tính tái sử dụng (reusability)
  - Dễ tái sử dụng trong các phần mềm mới
- Tính khả chuyển (portability)
  - Dễ sử dụng trong môi trường mới

Chất lượng phần mềm

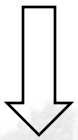
- Tính dễ hiểu (understandability)
  - Dễ hiểu đối với người sử dụng cũng như với người phát triển
- Tính hợp tác (interoperability)
  - Dễ hợp tác với các phần mềm khác
- Sản xuất hiệu quả (productivity)
  - Tiến trình sản xuất phần mềm phải hiệu quả
- Khả năng giao sản phẩm đúng hạn (timeliness)
  - Giao sản phẩm theo từng gói

Chất lượng phần mềm

- Sự thỏa hiệp giữa các tiêu chí chất lượng
  - Tính thân thiện / tính bền vững
  - Tính khả chuyển / tính hiệu quả

Đặc trưng của phần mềm

- Phát triển phần mềm khác chế tạo phần cứng
  - Sản xuất mang tính thủ công (sáng tạo, không theo khuôn mẫu)
  - Khó kiểm soát chất lượng ở các bước trung gian
  - Khó dự đoán trước về hiệu năng



áp dụng các phương pháp tiên tiến, tự động những khâu cụ thể

Đặc trưng của phần mềm

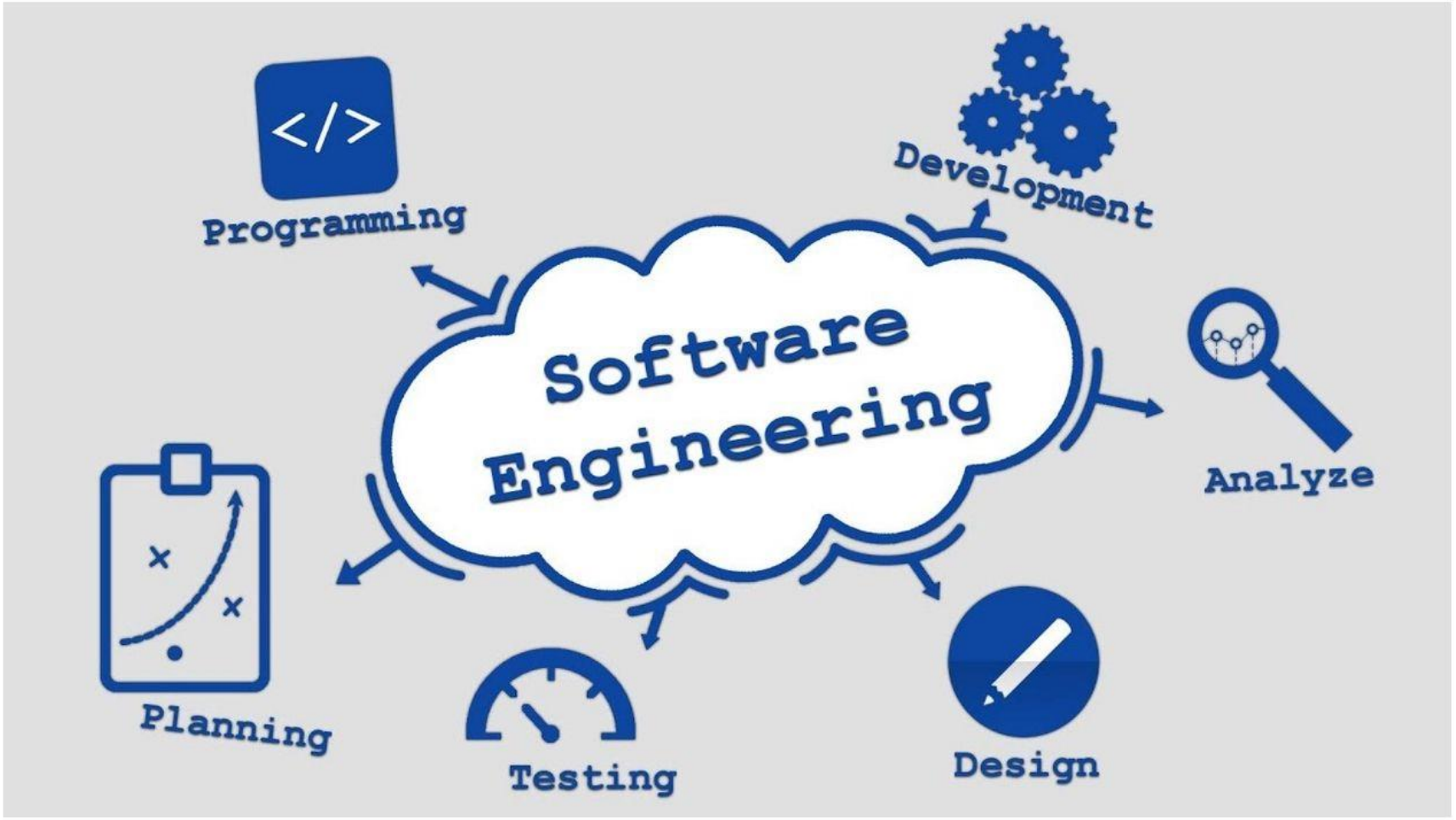
- Phần mềm thoái hóa theo thời gian
  - Môi trường sử dụng, nhu cầu thay đổi
  - Lỗi sinh ra tăng do nâng cấp
- Phần mềm không được lắp ráp theo mẫu
  - Không có danh mục chi tiết cho trước
  - Được đặt hàng theo từng yêu cầu riêng

"***bảo trì phần mềm phức tạp hơn hẳn so với bảo trì phần cứng***"

Tóm tắt

- Phần mềm có tầm quan trọng đặc biệt: tạo nên sự khác biệt của tổ chức, của hệ thống
  - Phát triển phần mềm là công việc phức tạp, rủi ro
  - là phần tử logic, không có độ đo trực quan, khó kiểm soát chất lượng khi phát triển
  - không được định hình trước, khó dự đoán hiệu năng khi chưa có sản phẩm
  - mang tính thủ công, phụ thuộc vào con người
  - bị ảnh hưởng lớn từ môi trường->nhiều rủi ro
- Cần áp dụng các phương pháp tiên tiến (kỹ nghệ phần mềm)

Công nghệ phần mềm



Khái niệm

- Công nghệ phần mềm (Software Engineering-SE)
  - Nghiên cứu và phát triển các phương pháp, kĩ thuật và công cụ nhằm xây dựng các phần mềm một cách kinh tế, có độ tin cậy cao và hoạt động hiệu quả
  - Thiết kế, xây dựng và bảo trì các phần mềm **phức tạp**, **bền vững** và **chất lượng**.

Các khái niệm

- Bauer [1969]: SE là việc thiết lập và sử dụng các nguyên lý công nghệ đúng đắn để thu được phần mềm 1 cách kinh tế vừa tin cậy vừa làm việc hiệu quả trên các máy thực
- Parnas [1987]: SE là việc xây dựng phần mềm nhiều phiên bản bởi nhiều người
- Sommerville [1995]: SE là một nguyên lý kỹ nghệ liên quan đến tất cả các mặt (lý thuyết, phương pháp và công cụ) của sản phần mềm

Mục đích

- Mục đích
  - Áp dụng thực tế
    - Các kiến thức khoa học
    - Các nguyên tắc kinh tế
    - Các nguyên tắc quản lý
    - Các kỹ thuật và công cụ thích hợp
  - Để sản xuất và bảo trì các phần mềm nhằm bảo đảm 4 yêu cầu (FQCD):
    - Phần mềm tạo ra phải đáp ứng được yêu cầu người sử dụng
    - Phần mềm phải đạt được các tiêu chuẩn về chất lượng
    - Giá thành phải nằm trong giới hạn đặt ra
    - Tiến độ xây dựng phần mềm phải đảm bảo

Nguyên tắc

- Các nguyên tắc cơ bản
  - Chặt chẽ (rigor and formality)
  - Chia nhỏ (separation of concerns)
  - Mô-đun hóa (modularity)
  - Trừu tượng (abstraction)
  - Phòng ngừa sự thay đổi (anticipation of change)
  - Tổng quát hóa (generality)
  - Giải quyết từng bước (incrementality)

Nguyên tắc

- Chặt chẽ (rigor and formality)
  - Sử dụng mô hình lý thuyết và toán học
  - áp dụng cho tất cả các bước, tất cả các sản phẩm
  - Ví dụ
    - “chọn z là giá trị lớn nhất của x và y”
    - z = max(x, y)

Nguyên tắc

- Chia nhỏ (separation of concerns)
  - Làm chủ độ phức tạp
    - Chỉ tập trung một lĩnh vực cùng một lúc
  - Chia vấn đề thành các phần nhỏ hơn
    - Giải quyết một phần nhỏ sẽ đơn giản hơn
    - “chia để trị” (divide and conquer)
  - Có thể chia nhỏ theo
    - thời gian: lập kế hoạch
    - khái niệm: giao diện / thuật toán
    - Xử lý: chia các xử lý con

Nguyên tắc

- Mô-đun hóa (modularity)
  - Chia nhỏ độ phức tạp
    - Dễ hiểu
    - Dễ quản lý các hệ thống phức tạp
  - Quan hệ mật thiết với nguyên tắc “chia nhỏ”
  - Các phương pháp mô-đun hóa
    - Chiến lược từ trên xuống (top-down)
    - Chiến lược từ dưới lên (bottom-up)
  - Chất lượng của mô-đun hóa
    - liên kết lỏng lẻo (low coupling)
    - Kết cố cao (high cohesion)

Nguyên tắc

- Trừu tượng (abstraction)
  - Loại bỏ những gì không quan trọng
  - Chỉ xem xét các yếu tố quan trọng
  - Sử dụng các mô hình
    - mô hình cho người sủ dụng
    - mô hình cho người phát triển
  - Ví dụ
    - Ngôn ngữ lập trình/cấu truc phần cứng
    - Xây dựng tài liệu
    - Đặc tả bởi điều kiện trước/sau

Nguyên tắc

- Phòng ngừa sự thay đổi (anticipation of change)
  - Phần mềm là sản phẩm thường xuyên thay đổi
  - Dự báo các yếu tố có thể thay đổi
    - ảnh hưởng có thể
  - các thay đổi thường gặp
    - Trong đặc tả yêu cầu
    - Trong ngữ cảnh sử dụng
    - Khả năng về công nghệ

Nguyên tắc

- Tổng quát hóa (generality)
  - xem xét vấn đề trong ngữ cảnh tổng quát
  - Giải quyết vấn đề lớn hơn
  - Mục đích
    - Tái sử dụng dễ dàng
    - Có thể sử dụng các công cụ có sẵn
      - Sử dụng design patterns
    - Chi phí có thể tăng cao

Nguyên tắc

- Giải quyết từng bước (incrementality)
  - Nguyên tắc
    - xác định một phần (tập con)
    - Phát triển
    - Đánh giá
    - Bắt đầu lại
  - Áp dụng cho
    - Phát triển một sản phẩm
      - Một đặc tả/một kiến trúc/…
    - Mô hình phát triển
      - Mô hình lặp

Các đặc trưng cơ bản

- Là một quá trình kỹ nghệ gồm ba mặt:
  - Thủ tục (procedures)
  - Phương pháp (methods)
  - Công cụ (tools)
- Nhằm tạo ra phần mềm hiệu quả, với các giới hạn cho trước

Các đặc trưng cơ bản

- Các thủ tục (procedures)
  - qui trình quản lý:
    - Xác định trình tự thực hiện các công việc
    - Xác định các tài liệu, sản phẩm cần bàn giao, và cách thức thực hiện
    - Định các mốc thời gian (millestones) và sản phẩm bàn giao

Các đặc trưng cơ bản

- Các phương pháp (methods)
  - Cách làm cụ thể để xây dựng phần mềm
  - Mỗi công đoạn làm phần mềm có các phương pháp riêng
    - Phương pháp phân tích (xác định, đặc tả)
    - Phương pháp thiết kế (mô hỡnh, thuật toán, dữ liệu...)
    - Phương pháp lập trình (hướng đối tượng)
    - Phương pháp kiểm thử (chức năng, cấu trúc)

Các đặc trưng cơ bản

- Các phương pháp (methods)
  - Phương pháp hướng cấu trúc thường bao gồm:
    - Mô hình về hệ thống: thường mô tả bằng đồ thị
    - Các ký pháp: giúp mô tả các mô hỡnh
    - Các quy tắc: các ràng buộc đặt lên mô hỡnh
    - Các đề xuất: các lời khuyên cho thiết kế tốt
    - Hướng dẫn về tiến trình: các hoạt động cần thực hiện

Các đặc trưng cơ bản

- Các công cụ - tools
  - Cung cấp sự trợ giúp tự động / bán tự động cho từng phương pháp
  - Computer Aided Software Engineering –CASEcác công cụ phân mềm được chuẩn hóa để trợ giúpcác công đoạn khác nhau trong quá trỡnh phát triển
  - Ví dụ:
    - compiler, debugger
    - công cụ sinh giao diện (C Builder,...)
    - hỗ trợ phân tích, thiết kế (Rwin,Modeler(Oracle Designer, Rational Rose,...)