

### Đảm bảo chất lượng phần mềm Software Quality Assurance

Đỗ Thị Bích Ngọc PTIT/FIT/SE

dothibichngoc@gmail.com



Bài 3: Các khái niệm cơ bản



- Ta hãy xem xét môi trường phát triển và bảo trì phần mềm chuyên nghiệp
- Đặc điểm chính:
  - 1. Điều kiện hợp đồng
  - 2. Quan hệ Khách hàng nhà cung cấp
  - 3. Yêu cầu Teamwork
  - 4. Hợp tác và phối hợp với các đội khác
  - 5. tương tác (interface) với các hệ thống phần mềm khác
  - 6. Cần tiếp tục dự án ngay cả khi có sự thay đổi thành viên
  - 7. Cần tiếp tục bảo trì phần mềm trong thời gian dài
- Tất cả các hành động trên (là các hình thức khác nhau của SQA) rất quan trọng để công việc kỹ thuật phần mềm thành công



#### 1. Điều kiện hợp đồng

- Bao gồm yêu cầu chức năng, ngân sách dự án, thời gian biểu dự án.
- Điều kiện hợp đồng là lớn nhất! (so với các tham số khác)
- Cung cấp phần mềm đúng hạn, trong ngân sách, đáp ứng được yêu cầu chức năng tạo thành lực đẩy và các yếu tố pháp lý của hợp đồng.



- 2. Quan hệ khách hàng nhà cung cấp
  - Ta cần nhận thức là khách hàng điều khiển quá trình bằng nhiều cách – thay đổi yêu cầu, đánh giá, kiểm thử chấp nhận, chấp nhận triển khai, phê duyệt bàn giao
    - Mối quan hệ này phải tốt, nhưng có thể phức tạp kinh khủng!!



#### 3. Yêu cầu Teamwork

- Yêu cầu thời gian biểu các thành viên của đội làm việc cùng nhau
- Có nhiều thông tin chuyên môn thảo luận. Thường xuyên? Ở đâu? Khi nào?
- Hỗ trợ lẫn nhau và xem xét để nâng cao chất lượng sản phẩm



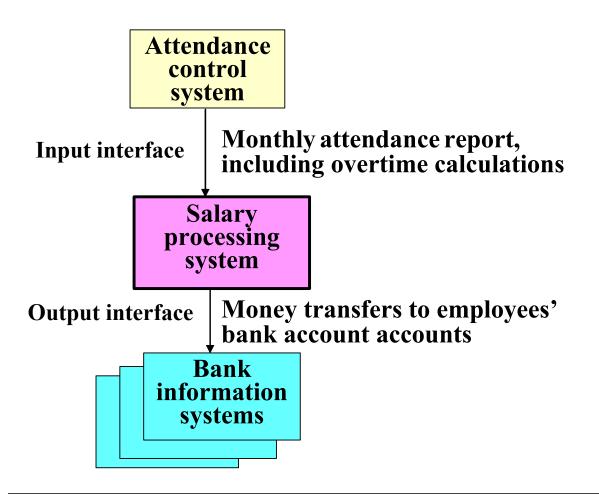
- 4. Hợp tác và phối hợp với các đội khác
  - Các tổ chức phần mềm lớn có nhiều đội và việc phát triển yêu cầu có hợp tác và phối hợp.
  - Chuyên gia ở đội khác?
    - Có thể mượn người đó?
  - Phát triển phần mềm có thể gia công từng phần.
  - Đội khác có thể phát triển phần mềm tương tự cho
     khách hàng và có thể hỗ trợ cực lớn
  - Xung đột



#### 5. Giao tiếp với các hệ thống khác

- Hệ thống đăng ký môn học có thể giao tiếp với hệ thống lập lịch lớp học và hệ thống thanh toán
- Thông thường, đầu ra của một hệ thống là đầu vào của hệ thống khác và ngược lại!
- Một hệ thống đăng ký môn học có thể cần 'interface' với một hệ thống thanh toán đã có với các định dạng file/cơ sở dữ liệu khác nhau.
- Đôi khi, đầu ra của một hệ thống là đầu vào của nhiều hệ thống.







- 6. Cần tiếp tục dự án ngay cả khi có sự thay đổi thành viên
  - Rất hay xảy ra, cần tính trước!
  - "The show must go on"
  - Fred Brooks: thêm người vào dự án trễ sẽ làm nó trễ hơn.
  - Fred Brooks: Giao tiếp với người mới
  - Fred Brooks: bắt người mới phải tăng tốc.
  - Thành viên của đội rời đi, được tuyển dụng, nghỉ phép, thuyên chuyển,...
  - Hiển nhiên là phiền toái, nhưng sự phát triển cần tiếp tục
  - Ta cần tính toán trước sự thay đổi này



- 7. Cần tiếp tục bảo trì phần mềm trong thời gian dài.
  - Phần mềm được phát triển để chạy nhiều năm.
    - Điều này ảnh hương tới công ty ở mức "đáy", không phải công ty phát triển phần mềm, trạng thái là "red"
  - Bảo trì là tốt và là nơi các công ty phát triển phần mềm kiếm tiền
  - Nhớ là, khi phát triển phần mềm, công ty ở trạng thái "red"
  - Cần tốn thời gian để triển khai và chuyển sang "black" và tạo doanh thu.



### Chất lượng phần mềm là gì?

### Phần mềm – định nghĩa của IEEE



#### Theo IEEE:

Phần mềm là: Chương trình máy tính, thủ tục, các tài liệu liên quan (nếu có) và dữ liệu liên quan

Định nghĩa của ISO (ISO 9000-3) danh sách 4 thành phần cần thiết để đảm bảo chất lượng của tiến trình phát triển phần mềm và bảo trì dài hạn:

- Chương trình máy tính (code)
- •Thủ tục
- •Tài liệu
- •Dữ liệu cần cho hoạt động của hệ thống phần mềm.



#### 1. Lỗi khi định nghĩa yêu cầu

- · Thường được xem như nguồn gốc của lỗi phần mềm
- Định nghĩa yêu cầu lỗi
  - Định nghĩa sai, ví dụ công thức sai
- Định nghĩa không đầy đủ
  - Yêu cầu không rõ ràng
- Thiếu yêu cầu
- Yêu cầu không cần thiết
  - · Nhiều dự án có những yêu cầu mà không bao giờ dùng đến
  - Ånh hưởng tới ngân sách, độ phức tạp, thời gian phát triển, ...



#### 2. Quan hệ Client-developer

- Hiểu sai các tài liệu yêu cầu
- · Hiểu sai tài liệu khi bị thay đổi
- Hiểu sai thay đổi (miệng) trong quá trình phát triển
- Không tham dự
  - Thông báo cho khách hàng về giải pháp của devlopers về những thay đổi yêu cầu và
  - Phản hồi của khách hàng về những câu hỏi của developer
- Đôi khi khách hàng trình bày như người dùng, lập trình viên trình bày theo tư duy khác hẳn.



#### 3. Sai phạm có chủ ý với yêu cầu phần mềm

- Developer tái sử dụng kết quả tương tự trước đó để tiết kiệm thời gian
- Thường tái sử dụng code cần chỉnh sửa những phần không cần hoặc không dùng được.
- developer(s) bỏ bớt chức năng do áp lực về thời gian/ngân sách.
- Developer chèn thêm "cải tiến" không được chấp nhận (cải tiến code, sắp xếp/tìm kiếm mới...) dẫn tới mất một vài tính năng



#### 4. Lỗi thiết kế logic

- Thuật toán sai.
  - · Công thức sai, bảng quyết định sai, toán tử/toán hạng sai...
- Định nghĩa tiến trình: các tiến trình trong hệ thống không phản ánh chính xác tiến trình nghiệp vụ
  - Note: không phải tất cả các lỗi đều là software errors.
  - Đây là lỗi thủ tục, và không phải là một phần của hệ thống...
- Lỗi khi định nghĩa điều kiện biên— nguồn lỗi phổ biến
  - Cần cẩn thận với các giá trị biên, ví dụ: 'no more than' "fewer than," "n times or more;" "the first time," ...



- 4. Lỗi thiết kế logic(tiếp)
  - Bổ sốt các trạng thái phần mềm
    - If rank is >= O1 and RPI is numeric, then.... Ta dễ bỏ sót hành động dựa trên trạng thái phần mềm
  - Bổ sốt các định nghĩa liên quan tới các phản ứng khi có hành động không hợp lệ trong phần mềm có code để phát hiện ra các hành động không hợp lệ nhưng không thiết kế các hành động đáp trả của phần mềm. Ví dụ: chuông cảnh báo,...



#### 5. Lỗi lập trình

- · Rất nhiều thứ liên quan tới lỗi coding.
  - Lỗi cú pháp (grammatical errors)
  - Lõi logic (program runs; results wrong)
  - Lỗi Run-time (crash during execution)



- 6. Không tuân thủ các hướng dẫn viết tài liệu và code
  - Không tuần thủ theo các khuôn mẫu templates (structure)
  - Không tuần thủ theo các chuẩn coding (attribute names...)
  - (Standards and Integration Branch)
    - Các chương trình khác phải chạy được trong môi trường!
    - Data Elements và Codes: AFM 300-4;
    - Tài liệu hướng dẫn và chỉ dẫn vận hành; AFDSDCM 300-8, ...
  - Đội SQA: kiểm thử không chỉ sự thực thi của phần mềm mà còn chuẩn coding, tài liệu hướng dẫn, thông báo được hiển thị, tài nguyên cần thiết, đặt tên tài nguyên (file names, program names,...)



#### 7. Thiếu sót của quá trình kiểm thử

- Là một phần của tiến trình phát triển nhưng thường xuyên bị cắt xén!
- Kế hoạch test không đầy đủ
  - Không test hết các phần của ứng dụng hoặc test qua loa!
- · Không phát hiện được lỗi tài liệu, báo cáo
- Không phát hiện được chính xác lỗi do mô tả mập mờ về lỗi đó
- Không đủ thời gian để sửa lỗi



#### 8. Lỗi giao diện người dùng và thủ tục

Các thủ tục chỉ dẫn cho người dùng cách thao tác cần thiết với từng bước của tiến trình. Chúng rất quan trọng với các phần mềm phức tạp đòi hỏi tiến trình gồm nhiều bước liên tiếp nhau, mỗi bước xử lý nhiều kiểu dữ liệu khác nhau và cho phép kiểm tra các kết quả trung gian

#### 9. Lỗi tài liệu

- Lỗi trong thiết kế tài liệu
- Lỗi trong tài liệu hướng dẫn sử dụng, online help
- Liệt kê những chức năng không tồn tại
  - Đã từng lập kế hoạch phát triển, nhưng hoãn và chưa kịp sửa tài liệu
- Thông báo lỗi vô nghĩa



#### 9 nguyên nhân gây ra lỗi phần mềm:

- 1. Lỗi khi định nghĩa yêu cầu
- 2. Quan hệ Client-developer tồi
- 3. Sai phạm có chủ ý với yêu cầu phần mềm
- 4. Lỗi thiết kế logic
- 5. Lỗi lập trình
- 6. Không tuân thủ các hướng dẫn viết tài liệu và code
- 7. Thiếu sót của quá trình kiểm thử
- 8. Lỗi giao diện người dùng và thủ tục
- 9. Lỗi tài liệu

Đặc tả (Specification): đặc tả lỗi, không đầy đủ, không nhất quán.

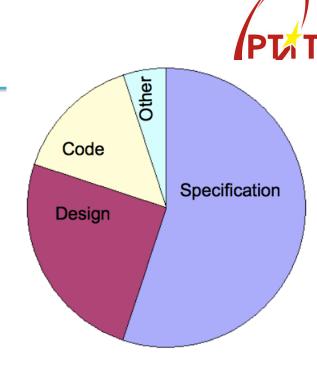
Thiết kế (Design): lỗi cơ bản trong thiết kế phần mềm.

Cài đặt (Code): lỗi lập trình, mã độc (malicious code).

#### Khác:

**Hệ thống hỗ trợ:** Ngôn ngữ lập trình nghèo nàn, trình biên dịch có lỗi...

Kiểm thử không đầy đủ: kiểm thử chưa xong, kiểm chứng nghèo nàn,...



Cited from R.
Patten's "Software testing", SAMS publishing





### Không có định nghĩa thống nhất cho "chất lượng phần mềm"

## Chất lượng Phần mềm – định nghĩa của la La Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần mềm – định nghĩa của la Chát lượng Phần lư

#### Chất lượng phần mềm là:

- Mức độ mà một hệ thống, thành phần, hay tiến trình đáp ứng được đặc tả yêu cầu by Philip Crosby
- Mức độ mà một hệ thống, thành phần, hay tiến trình đáp ứng được nhu cầu/mong muốn của khách hàng/người dùng.

by Joseph M. Juran

### Chất lượng Phần mềm – định nghĩa của la La

#### Software quality is:

(1) Mức độ mà một hệ thống, thành phần, hay tiến trình đáp ứng được đặc tả yêu cầu.

Nhấn mạnh vào đặc tả: nếu khách hàng nói rõ tất cả yêu cầu trong đặc tả VÀ nếu đặc tả được đáp ứng thì khách hàng sẽ hài lòng.

Đặc tả được soạn bởi khách hàng và đội phát triển. Do vậy, lỗi đặc tả sẽ không bị coi và không ảnh hưởng tới chất lượng phần mềm. Đây là điều ta cần xem xét.

### 



Chất lượng phần mềm là: (Joseph Juran)

Mức độ mà một hệ thống, thành phần, hay tiến trình đáp ứng được nhu cầu/mong muốn của khách hàng/người dùng.

Ở đây, nhấn mạnh vào việc đáp ứng các yêu cầu của khách hàng. Do vậy, đặc tả có thể phải chỉnh sửa

Tuy nhiên, khách hàng phải có khả năng đưa ra một bản đặc tả chính xác và đầy đủ.

Trên thực tế, nhiều vấn đề lớn có thể được phát hiện ra quá muộn. Khách hàng lại không hài lòng!

### Chất lượng Phần mềm – định nghĩa củ**p**y Roger Pressman (dùng trong môn này)

•Sự đáp ứng các yêu cầu chức năng, hiệu năng, các chuẩn (đặc tả) được phát triển, các đặc trưng mong muốn từ mọi phần mềm chuyên nghiệp (ngầm định).

### Đảm bảo chất lượng Phần mềm



#### Đảm bảo chất lượng phần mềm là:

Một cách hệ thống, tập hợp các hành động được lập kế hoạch cần thiết để đem lại sự tin cậy là tiến trình phát triển phần mềm/tiến trình bảo trì tuân thủ theo các yêu cầu kỹ thuật về chức năng đã đưa ra cũng như yêu cầu quản lý về giữ lịch trình và hoạt động trong phạm vi ngân sách

## Đảm bảo chất lượng Phần mềm – mục ti**ết**

- (1) Đảm bảo một mức độ tin cậy chấp nhận được là phần mềm sẽ tuân thủ các yêu cầu kỹ thuật về chức năng.
- (2) Đảm bảo một mức độ tin cậy chấp nhận được là phần mềm sẽ tuân thủ các yêu cầu quản lý về thời gian và tài chính.
- (3) Khởi đầu và quản lý các hoạt động để phát triển phần mềm và các hoạt động SQA được cải thiện và đạt hiệu quả cao hơn.