* Software Quality Concepts and Culture
* Software Quality Standards
* Inspections and reviews
* Testing, Verification and Validation

Tôi nhận được một email mà người gửi viết: “Khác biệt giữa trắc nghiệm - Verification và kiểm nghiệm Validation (V&V) là gì và có bao nhiêu kĩ thuật V&V?"  
  
Câu trả lời của tôi: Nhiều sinh viên lẫn lộn về thuật ngữ trắc nghiệm và kiểm nghiệm bởi vì chúng thường được dùng đổi lẫn cho nhau trong một số sách giáo khoa. Tuy nhiên, có khác biệt về nghĩa của chúng. Theo Bảng từ chuẩn IEEE về thuật ngữ kĩ nghệ phần mềm, trắc nghiệm được định nghĩa là "Qui trình đánh giá hệ thống hay cấu phần để xác định liệu sản phẩm của pha phát triển đã nêu có thoả mãn các điều kiện được áp đặt lúc bắt đầu pha đó không." Kiểm nghiệm được định nghĩa là "Qui trình đánh giá một hệ thống hay cấu phần trong hay cuối qui trình phát triển để xác định liệu nó có thoả mãn các yêu cầu đặc biệt không." Về căn bản, trắc nghiệm chứng tỏ liệu cái ra của pha có tuân thủ theo cái vào của pha không, tuy nhiên nó không phát hiện lỗi nếu cái vào là không đúng. Bởi vì phụ thuộc một mình vào trắc nghiệm là KHÔNG đủ, cho nên kiểm nghiệm là cần để kiểm tra các vấn đề với đặc tả yêu cầu để chứng minh rằng hệ thống làm việc đúng tương ứng.  
  
Có vài kĩ thuật trắc nghiệm nhưng phần lớn rơi vào hai khu vực chính: Kiểm thử động và kiểm thử tĩnh.  
• Kiểm thử động bao gồm việc thực hiện hệ thống hay cấu phần. Về căn bản, một số các trường hợp kiểm thử được chọn ra, tại đó từng trường hợp kiểm thử đều có chứa dữ liệu kiểm thử. Những trường hợp kiểm thử này được dùng để xác định kết quả kiểm thử ra. Kiểm thử động có thể được phân chia thêm thành ba loại - kiểm thử chức năng, kiểm thử cấu trúc, và kiểm thử ngẫu nhiên.  
• Kiểm thử chức năng bao gồm nhận diện và kiểm thử tất cả các chức năng của hệ thống như đã được xác định trong yêu cầu. Dạng này của kiểm thử là ví dụ về kiểm thử hộp đen vì nó không bao gồm tri thức về thực hiện hệ thống.  
• Kiểm thử cấu trúc bao gồm kiểm thử có tri thức đầy đủ về thực hiện hệ thống (kiểm thử hộp trắng). Nó dùng thông tin từ cấu trúc nội bộ của hệ thống để làm ra kiểm thử để kiểm vận hành của từng cấu phần riêng lẻ. Kiểm thử chức năng và cấu trúc cả hai đều chứa các trường hợp kiểm thử mà sẽ kiểm đặc trưng đặc thù của hệ thống.  
• Kiểm thử ngẫu nhiên - Kiểm thử chọn tự do các trường hợp kiểm thử trong tập mọi trường hợp kiểm thử có thể có. Việc dùng cái vào được xác định ngẫu nhiên có thể phát hiện ra lỗi không được các kĩ thuật kiểm thử hệ thống khác phát hiện ra.  
  
• Kiểm thử tĩnh là kiểm thử không chứa việc thực hiện hệ thống hay cấu phần. Một số có thể được thực hiện một cách thủ công trong khi các kiểm thử khác được tự động hoá. Kiểm thử tĩnh có thể được phân chia thêm thành các kĩ thuật phân tích tính nhất quán và kĩ thuật đo tính chất chương trình.  
• Kĩ thuật về tính nhất quán - Các kĩ thuật được dùng để đảm bảo tính chất chương trình như đúng cú pháp, tương ứng đúng tham biến giữa các thủ thục, đúng định kiểu, và dịch đúng yêu cầu và đặc tả.  
• Kĩ thuật đo - Kĩ thuật đo các tính chất như việc sinh lỗi, tính hiểu được, và có cấu trúc tốt.  
Có một vài kĩ thuật kiểm nghiệm như phương pháp hình thức, cách tiêm lỗi (phần cứng và phần mềm), phân tích rủi ro và phân tích phụ thuộc. Kiểm nghiệm thường xảy ra ở cuối chu kì phát triển, và nhìn vào hệ thống đầy đủ để trắc nghiệm, hội tụ vào các hệ con nhỏ hơn.  
  
• Phương pháp hình thức - Phương pháp hình thức dùng các kĩ thuật toán học và logic để diễn đạt, nghiên cứu, và phân tích đặc tả, thiết kế, tài liệu, và hành vi của cả phần cứng và phần mềm.  
• Tiêm lỗi - Tiêm lỗi là việc kích hoạt có chủ định các lỗi hoặc bởi phương tiện phần cứng hay phần mềm để quan sát vận hành hệ thống trong điều kiện có lỗi.  
• Phân tích tính phụ thuộc - Phân tích tính phụ thuộc bao gồm nhận diện những nguy cơ và rồi đề đạt giải pháp làm giảm rủi ro của nguy cơ xuất hiện.  
• Phân tích rủi ro – qui trình nhận diện các hậu quả có thể của từng nguy cơ và xác suất xuất hiện của chúng.

* Statistical Quality Assurance
* **Software Quality Processes- bảo đảm chất lượng PM**

Tại một số công ty ở Việt Nam, SQA được xem như việc kiểm lỗi (test) khi sản phẩm đã định hình. Số khác lại coi SQA là việc kiểm tra đầu ra trung gian của sản phẩm, để sản phẩm đạt được sự nhất quán trong quá trình thực hiện. Dù ở khía cạnh nào thì SQA cũng là kiểm tra trực tiếp sản phẩm

Khi dự án phần mềm trở nên lớn hơn và phức tạp hơn, vai trò của Đảm bảo chất lượng phần mềm - Software Quality Assurance (SQA) trở nên gay gắt hơn. Ngay cả ngày nay, các phương pháp để đảm bảo chất lượng phần mềm vẫn thường không được nhiều người quản lí dự án hiểu rõ.   
**Đảm bảo chất lượng phần mềm yêu cầu rằng tri thức và kỉ luật kĩ nghệ phải được áp dụng trong MỌI pha của vòng đời phát triển, KHÔNG phải là những pha cuối cùng của kiểm thử hay đưa ra như nhiều người vẫn hiểu lầm.**   
Người kĩ sư đảm bảo chất lượng phần mềm được yêu cầu có nhiều năm phát triển phần mềm và tri thức miền đủ để đánh giá tính đầy đủ và tính đúng đắn của yêu cầu hệ thống, và họ phải có khả năng xác định liệu thiết kế có tổ hợp mọi yêu cầu một cách chính xác không. Cuối cùng, người kĩ sư SQA chịu trách nhiệm về quản lí thông báo liệu sản phẩm phần mềm có tin cậy không và có đáp ứng chuẩn chất lượng không. Với loại công việc này, người kĩ sư SQA phải là người có kinh nghiệm nhất trong tổ chức. Họ phải làm việc như người phát triển phần mềm trong nhiều năm và đi lên người lãnh đạo kĩ thuật hay kiến trúc sư và thực hiện công việc này trong nhiều năm trước khi trở thành kĩ sư SQA.   
  
Cuốn “Sổ tay của Đảm bảo chất lượng phần mềm,” định nghĩa SQA là: "Tập các hoạt động có hệ thống cung cấp bằng chứng về khả năng của qui trình phần mềm tạo ra sản phẩm phần mềm khớp với việc sử dụng. Do đó hội tụ của SQA là giám sát liên tục trong toàn thể vòng đời phát triển phần mềm để đảm bảo chất lượng của sản phẩm được chuyển giao. Điều này yêu cầu giám sát cả qui trình và sản phẩm. Trong đảm bảo qui trình, SQA cung cấp việc quản lí với phản hồi khách quan liên quan tới tuân thủ các kế hoạch, thủ tục, chuẩn và phân tích đã được chấp thuận. Các hoạt động đảm bảo sản phẩm hội tụ vào mức độ thay đổi của chất lượng sản phẩm bên trong từng pha của vòng đời, như yêu cầu, thiết kế, viết mã và kế hoạch kiểm thử. Mục tiêu là nhận diện và khử bỏ khiếm khuyết trong toàn bộ vòng đời sớm nhất có thể được, do vậy giảm chi phí kiểm thử và bảo trì.  
  
Viện các kĩ sư điện và điện tử (IEEE) định nghĩa chất lượng là "mức độ mà hệ thống, cấu phần, hay qui trình đáp ứng cho các yêu cầu xác định, và nhu cầu hay mong đợi của khách hàng hay người dùng." Trong khi định nghĩa này dường như rõ ràng và không mơ hồ, nhiều người quản lí phần mềm vẫn phàn nàn rằng chất lượng là "khó định nghĩa, không thể đo được, khó nhận ra” và do đó bỏ qua nó. Sau đây là định nghĩa chi tiết khác về chất lượng phần mềm như được định nghĩa trong “Sổ tay của Đảm bảo chất lượng phần mềm” chuẩn.   
  
Tính đúng đắn: mức độ mà dự án hoàn thành các đặc tả của nó.   
Tính hiệu quả: dùng tài nguyên trong thực hiện và lưu giữ.   
Tính linh hoạt: dễ làm thay đổi được yêu cầu do thay đổi trong môi trường vận hành.   
Tính toàn vẹn: bảo vệ dự án khỏi truy nhập không được phép.   
Tính liên tác: nỗ lực được yêu cầu để tích hợp hệ thống vào hệ thống khác.   
Tính bảo trì: nỗ lực được yêu cầu để định vị và sửa lỗi trong dự án trong môi trường vận hành của nó.   
Tính khả chuyển: nỗ lực được yêu cầu để truyền dự án từ môi trường này sang môi trường khác.   
Tính tin cậy: khả năng không hỏng.   
Tính tái dụng: dễ dùng lại phần mềm trong hoàn cảnh khác.   
Tính kiểm thử được: dễ dàng kiểm thử dự án để đảm bảo rằng nó không lỗi và đáp ứng đặc tả.   
Tính khả dụng: dễ dùng phần mềm.   
(Efficiency: use of resources execution and storage.   
Flexibility: ease of making changes required by changes in the operating environment.   
Integrity: protection of the project from unauthorized access.   
Interoperability: effort required to integrate the system to another system.   
Maintainability: effort required to locate and fix a fault in the project within its operating environment.   
Portability: effort required to transfer a project from one environment to another.   
Reliability: ability not to fail.   
Reusability: ease of re-using software in a different context.   
Testability: ease of testing the project to ensure that it is error-free and meets its specification.   
Usability: ease of use of the software.)

Tất nhiên, trong một thế giới hoàn hảo tất cả những tiêu chí này sẽ được đáp ứng, nhưng trong thực tế việc bù trừ là một phần của mọi dự án phát triển. Thường phần mềm hiệu quả nhất lại không khả chuyển, vì tính khả chuyển sẽ yêu cầu mã phụ thêm, làm giảm tính hiệu quả. Tính khả dụng là chủ quan và thay đổi tuỳ theo kinh nghiệm của người dùng. Khi dùng các tiêu chí này để xác định mục tiêu đảm bảo của hệ thống phần mềm, mục đích và việc dùng hệ thống phải được tính tới. Trong thế giới thực của phát triển phần mềm, tiêu chí về chất lượng được nhận diện và áp dụng cho mức độ khác biệt xem như kết quả của các quyết định bù trừ.   
  
Với toàn cầu hoá, khi nhiều công ti làm kinh doanh qua các biên giới quốc gia, yêu cầu về chất lượng sản phẩm đang trở nên quan trọng hơn. Thực tế đã chứng minh rằng việc có SQA là đảm bảo rằng có kỉ luật và kiểm soát trong qui trình phát triển phần mềm thông qua đánh giá độc lập do đó SQA sẽ xác định liệu một sản phẩm sẽ được chấp nhận ở chỗ nào đó hay không. Có hai mô hình phổ biến để kiểm điểm và đảm bảo chất lượng phần mềm: ISO 9000 và CMMI. Tổ chức tiêu chuẩn quốc tế (ISO 9000) cung cấp một cách để thu được việc uỷ nhiệm bên ngoài cho hệ thống quản lí chất lượng. Nhiều công ti đã dùng ứng dụng của ISO cho phần mềm, nhưng vấn đề là ở chỗ nó hội tụ phần lớn vào thủ tục thay vì qui trình. Mô hình kia là Tích hợp mô hình trưởng thành năng lực (CMMI) của Viện kĩ sư phần mềm hội tụ trên cơ sở rằng chất lượng của sản phẩm phần mềm chủ yếu được xác định bởi chất lượng của qui trình phát triển và bảo trì phần mềm được dùng để xây dựng nó.  
  
Đảm bảo chất lượng là mấu chốt cho mọi doanh nghiệp tương lai. Có SQA có kinh nghiệm là bản chất cho doanh nghiệp nhưng ngay cả ngày nay, nhiều công ti phần mềm hiếm khi đầu tư đủ ngân quĩ để thực hiện công việc SQA. Một số người tin họ có thể tránh được nó nhiều nhất có thể được. Thái độ “cắt giảm chi phí” và có sản phẩm chất lượng kém là không thể chấp nhận được trong thế giới cạnh tranh cao. Nhiều công ti sẽ KHÔNG sống sót lâu được vì nhiều khách hàng đang đòi hỏi sản phẩm chất lượng tốt hơn với an toàn và tin cậy tốt nhất.

* **Process Assurance- bảo đảm chất lượng quy trình**

Còn PQA thì có 2 việc chính:

**Một** là xây dựng hệ thống quy trình cho doanh nghiệp (bằng cách ứng dụng những quy trình quản lý sẵn có như ISO hay CMM hoặc dựa trên đó xây dựng quy trình chuẩn cho doanh nghiệp).

Doanh nghiệp phần mềm nào cũng hiểu rằng xây dựng được một quy trình quản lý chất lượng tốt chính là cách làm tăng lợi nhuận hơn gấp nhiều lần.   
  
Trong chi phí về chất lượng sản phẩm, ***có 3 loại chi phí con:***   
-chi phí chặn lỗi (prevention cost);   
-chi phí kiểm soát lỗi (control cost)  
-chi phí xảy ra khi có lỗi (failure

Theo một số tài liệu về quy trình sản xuất phần mềm (như CMM), 1 đồng bỏ ra cho khâu thứ nhất sẽ làm giảm 10 đồng cho khâu thứ 3, và 1 đồng cho khâu thứ 2 sẽ giảm 3 đồng cho khâu thứ 3. Muốn làm được việc ấy, điều hiển nhiên phải có người đưa quy trình ấy vào thực tế. Nhưng để tuyển được một PQA không phải là điều đơn giản.

**Hai** là thực hiện việc giám sát, kiểm tra việc thực hiện quy trình của từng bộ phận, từng dự án (DA), từ đó tổng hợp thông tin để đưa ra những cải tiến cho quy trình hoạt động tốt.

Với công việc đúng nghĩa như trên thì PQA đang là “của hiếm” trong các công ty phần mềm và cũng là đại diện nghề QA chúng tôi muốn đề cập. Doanh nghiệp phần mềm nào cũng hiểu rằng xây dựng được một quy trình quản lý chất lượng tốt chính là cách làm tăng lợi nhuận hơn gấp nhiều lần. Trong chi phí về chất lượng sản phẩm, có 3 loại chi phí con: chi phí chặn lỗi (prevention cost); chi phí kiểm soát lỗi (control cost) và chi phí xảy ra khi có lỗi (failure). Theo một số tài liệu về quy trình sản xuất PM (như CMM), 1 đồng bỏ ra cho khâu thứ nhất sẽ làm giảm 10 đồng cho khâu thứ 3, và 1 đồng cho khâu thứ 2 sẽ làm giảm 3 đồng cho khâu thứ 3. Muốn làm được việc ấy, điều hiển nhiên phải có người đưa quy trình ấy vào thực tế. Nhưng để tuyển được một PQA không phải là điều đơn giản.

nhìn chung PQA không đòi hỏi phải có kiến thức sâu về CNTT như SQA vì không liên quan đến sản phẩm chính của DA PM. Tuy nhiên, kiến thức của người làm PQA phải rộng và có độ sâu nhất định để hiểu hết công việc trong phạm vi hệ thống mình đang quản lý, đặc biệt là khả năng tổ chức, tư duy logic và có hệ thống. Ngoài ra, kỹ năng giao tiếp và khai thác thông tin của nhân viên PQA cũng quan trọng. Trong công ty có nhiều người, mỗi người giỏi ở một lĩnh vực khác nhau. PQA cần biết khai thác để tận dụng những thông tin đó cho việc xây dựng hệ thống quy trình.  
“Hiểu biết về quy trình ở mức sâu; Khả năng trình bày, thuyết phục tốt; Kỹ năng lắng nghe, để thấy cái gì cần điều chỉnh thì điều chỉnh và cũng để thuyết phục tốt hơn; Kỹ năng đo lường và phân tích số liệu; Kỹ năng làm PM (kỹ năng này sẽ được ưu tiên cao khi tuyển dụng)”

Nói như vậy, không có nghĩa là một sinh viên CNTT học tốt ra trường có thể làm PQA. Trong trường, sinh viên cũng được dạy về các quy trình làm PM nhưng là học kiểu “cưỡi ngựa xem hoa”, chỉ biết làm phần mềm phải theo những bước nào. Nhưng giữa các bước đó có liên quan với nhau như thế nào, cần kiểm tra ra sao thì họ chưa hình dung được. Vì vậy, hiện nay hầu hết các công ty đều có chương trình đào tạo riêng cho vị trí này sau thời gian tuyển dụng.

“Thường sau 1 tháng, một lập trình viên mới có thể bắt nhịp được với công việc. Còn với vị trí PQA, thời gian làm quen ít nhất phải hai tháng. Bởi vì để đào tạo họ hiểu biết về hệ thống, tư duy về hệ thống và hiểu biết về mô hình quản lý chất lượng thì không thể là trong một thời gian ngắn. Thông thường họ cũng phải trải qua vài DA làm việc test lỗi với mức độ khó dần để hình dung được quy trình, nắm bắt được yêu cầu của DA”.  
Khó khăn như vậy nhưng đây lại là nghề thu hút được giới nữ. Qua khảo sát, phần lớn PQA tại các công ty phần mềm hiện nay là nữ. Điều này có lẽ là do nghề đòi hỏi tính kiên nhẫn và mềm dẻo trong giao tiếp, đồng thời không phải chạy theo deadline các dự án như các lập trình viên. Mức thu nhập của nghề cũng không hề kém các vị trí khác trong ngành công nghiệp phần mềm. “QA là một bộ phận quan trọng trong công ty, và chúng tôi thích điều đó”, chị Minh nói về niềm yêu thích công việc của mình.

“Hiện nay, đa số các mô hình đào tạo CNTT tại Việt Nam chỉ tập trung đào tạo nghề lập trình viên và quản trị mạng. Tuy nhiên, theo chúng tôi được biết, một số bạn sau thời gian làm lập trình viên chuyển qua vị trí QA vẫn đảm đương tốt công việc. Điều đó chứng tỏ rằng cơ hội với nghề QA không khó với tới cho các sinh viên CNTT”.

* Product Assurance
* Software Metrics
* Process Metrics

<http://testervn.com/forums/forumdisplay.php?s=e15782c0af7f77e0b54257fea274164d&f=251>

<http://coltech.vnu.edu.vn/~hungpn/class/ASE/docs/7_Testing.pdf>