Bài thu hoạch Công nghệ phần mềm

21020055 - Trần Thùy Dung

April 26

1 Kiểm thử

Kiểm thử là một phần của quá trình **Verification and Validation (V&V)**. Đó là **hai chu trình độc lập** được kết hợp với nhau để kiểm tra một sản phẩm (dịch vụ, hệ thống) có đáp ứng đặc tả yêu cầu hay không.

- **Verification.** là quá trình tìm lỗi implementing so bới **thiết kế**. Nói cách khác, đây là quá trình xác định đội ngũ có **làm sản phẩm đúng hay không**.
  - **In:** $\mathbb{D} \to \mathbb{I}$
  - **Out:** $\mathbb{I} \vdash \mathbb{D}$

    Nếu `fail`: Kèm theo counter-eg.
- **Validation.** là quá trình kiểm tra sản phẩm liệu có đúng với **đặc tả yêu cầu**. Nói cách khác, đây là quá trình xác định đội ngũ có **làm đúng sản phẩm hay không**.
  - **In:** $\mathbb{SRS} \to \mathbb{I}$
  - **Out:** $\mathbb{I} \vdash \mathbb{SRS}$

Nếu quá trình verification **không** diễn ra trước validation, sẽ không thể xác định được ngọn nguồn của lỗi. Cụ thể, nếu như validation diễn ra trước, khi thất bại, nguyên nhân không rõ là do lập trình sai thiết kế hay thiết kế sai so với yêu cầu. Trái lại, nếu verification diễn ra trước, khi thất bại, nguyên nhân đã hoàn toàn xác định là do lập trình sai thiết kế.

Có hai phương thức để kiểm tra kết quả có đúng yêu cầu đề ra hay không:

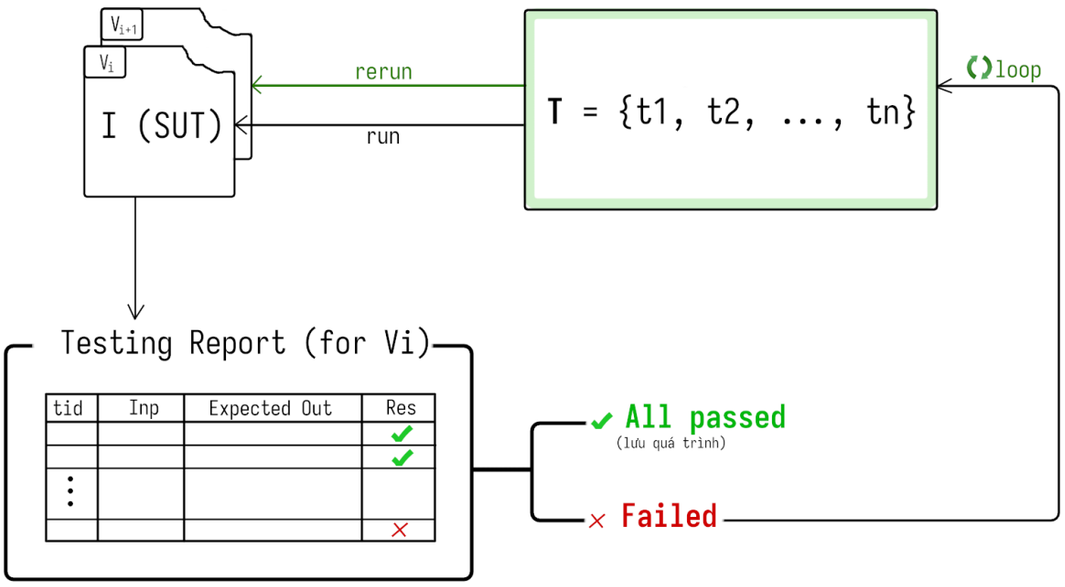
- **Static V&V.** Kiểm thử mà không cần execute code.
- **Dynamic V&V.** Kiểm thử có execute code. Đây cũng chính là **Software Testing**.

Mô tả quá trình kiểm thử

Giả sử chúng ta đã implement sản phẩm tại $\texttt{version}_\texttt{i}$ là V_i, có bộ test $\mathbb{T} = \{t_1, t_2, ..., t_n\}$. Sau khi chạy xong bộ test, một bản báo cáo kết quả cho V_i được sinh ra, thể hiện rõ tình trạng kiểm thử.

- Nếu bộ test `all passed` *sau một quá trình kiểm thử*, quá trình đó phải được ghi lại rõ ràng.

  Chất lượng sản phẩm phụ thuộc vào chất lượng của bộ test (bộ test càng mạnh, chất lượng càng cao). Dường như không thể có chuyện chạy bộ test `all passed` ngay lần đầu, nguyên do chỉ có thể do test yếu.
- Nếu `failed`, quá trình kiểm thử tiếp tục với `version` tiếp theo là V_{i+1} và bộ test $\mathbb{T}$, cho đến khi nào `pass` mới thôi.



Hình 1: Testing

2 Về Tokyo Techies

Anh Đỗ Bá Đức

Anh Đỗ Bá Đức, để trở thành CEO cũng như Founder của Tokyo Techies như bây giờ, đã trải qua nhiều giai đoạn, từ ban đầu có nền tảng là văn học, sau đó có một khoảng thời gian làm việc tại Softbank, và rồi anh quyết định chuyển sang dạy học với tầm nhìn xa. Đến năm 30 tuổi, anh bắt đầu `hacking`.

Trong bối cảnh hiện nay với xu thế là **chuyển đổi số**, nói cách khác, mọi tổ chức đều muốn công nghệ hóa ý tưởng của mình, anh Đức đã thành lập Tokyo Techies với trọng tâm là one-stop IT consultance và **giải quyết các bài toán chuyển đổi số cho các doanh nghiệp**. Kỳ vọng của các công ty là lời giải cho bài toán lớn phục vụ hàng triệu người (và nhiều hơn thế), tuy nhiên trên thực tế, để có được lượng lớn kỹ sư phần mềm không dễ dàng. Nhìn nhận được điều này, Tokyo Techies đóng vai trò là công ty phần mềm, đưa nhu cầu này thành hoàn toàn có thể. Bởi tầm nhìn lâu dài, anh Đức mong muốn công ty trở thành một công ty toàn cầu, với tư duy toàn cầu.

Về miền, khách hàng của Tokyo Techies bao phủ nhiều miền khác nhau, từ bán lẻ, giáo dục, vận chuyển, y tế, ... Còn về công nghệ, Tokyo Techies cũng bao phủ tương đối nhiều mảng và dịch vụ, mà cốt lõi nhất là **lĩnh vực bảo mật**.

Anh chia sẻ, kiến thức công nghệ chỉ đóng 20% trong con đường sự nghiệp sau này. Nói cách khác, công nghệ chỉ là bare minimum để mỗi chúng ta gia nhập thị trường mà thôi. Vấn đề là, phải làm sao để vận dụng kỹ năng công nghệ để giải quyết các vấn đề thực tế, từ đó đem lại giá trị.