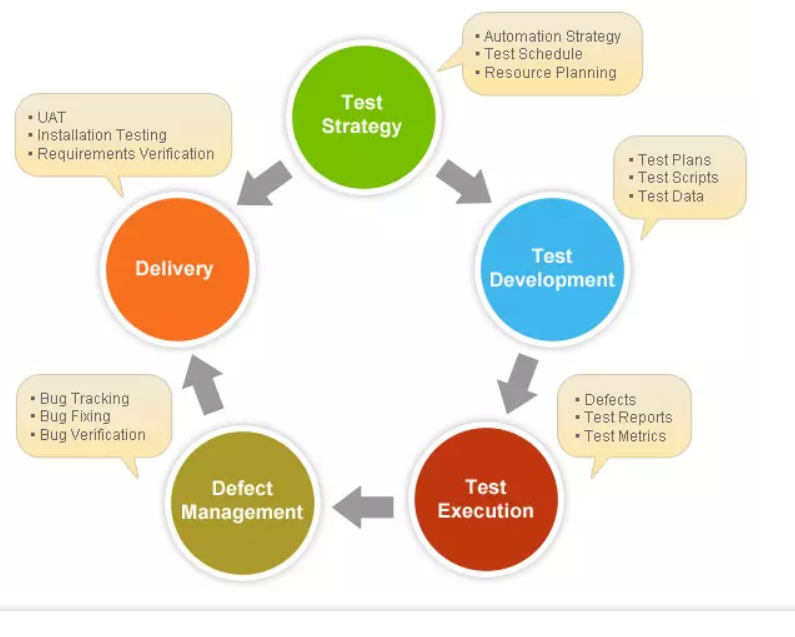
**Test Strategy (TS) là gì?**

Test Strategy (chiến lược test) là cách bạn xác định hướng tiếp cận test, cái bạn muốn đạt được, và làm như nào để bạn đạt được cái bạn muốn.

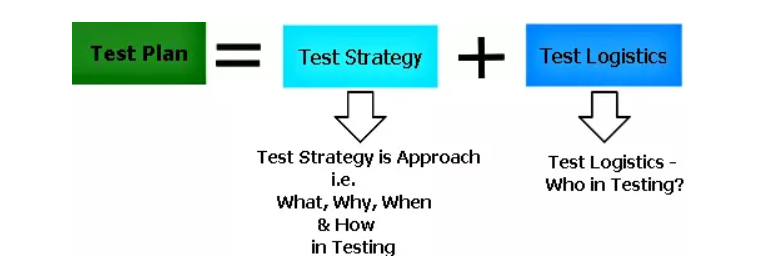
Trong file strategy, hãy đưa ra một kế hoạch tiếp cận đối tượng test một cách rõ ràng. Đây là một trong những tài liệu quan trọng nhất của QA. Viết TS hiệu quả là kĩ năng mà tester nào cũng cần phải có. Việc suy nghĩ về TS thấu đáo sẽ giúp phát hiện những requirement bị thiếu. Hoạt động trước khi test này giúp team xác định được phạm vi test và mức độ bao phủ của test. Nó giúp test manager nắm được tình trạng của dự án vào bất cứ thời điểm nào. Khả năng xảy ra thiếu hoạt động test nào là khá thấp vì đã có TS rồi.



Chưa có kế hoạch test mà đã chạy test case rồi sẽ kém hiệu quả. Có những team viết TS ra nhưng khi test lại không ngó lại. TS phải được thảo luận với toàn team để mọi người thống nhất trách nhiệm, cách tiếp cận trong việc test.

**Test Strategy vs Test Plan?**

Theo IEEE Standard 829-2008, TS là 1 phần của Test Plan. Test Plan gồm scope của dự án và mục tiêu kiểm thử. Đơn giản, nó sẽ gồm: test những gì, các tính năng được test, các tính năng không được test, đánh giá, lên lịch test, quản lý nguồn lực test. Trong khi đó, TS đưa ra hướng dẫn, cách tiếp cận test để có thể đạt được mục tiêu và việc thực thi các kiểu test được định nghĩa trong test plan. Nó gồm mục tiêu test, cách tiếp cận, môi trường test, các công cụ và chiến lược automation (nếu có), phân tích hiểm họa và kế hoạch đối phó với tình huống bất ngờ. Tổng kết, Test plan là cái bạn muốn đạt được, Test Strategy là kế hoạch để thực hiện mục tiêu bạn đặt ra.



**Test Strategy có gì?**

**1. Scope và Overview.** Tổng quan dự án với thông tin về người sẽ sử dụng tài liệu này. Hãy thêm các chi tiết về người review và approve TS. Xác định các hoạt động test và các giai đoạn test với timeline rõ ràng tương ứng với timeline dự án được định nghĩa trong test plan.

**2. Cách tiếp cận test.** Xác định tiến trình test, cấp độ test, vai trò và trách nhiệm của thành viên dự án. Với mỗi loại test (vd: unit, integration, system, regression, usability, load, performance, and security testing) hãy miêu tả lý do tại sao cách thức test đó lại cần được thực thi, đi kèm đó là thông tin khi nào test, ai test, trách nhiệm, cách tiếp cận test, chiến lược test automation và các công cụ test auto nếu có. Khi test, có nhiều hoạt động khác nhau thêm bug mới, sắp xếp độ ưu tiên của bug, assign bug cho ai, test lại bug, test hồi qui, UAT. Bạn phải định nghĩa chính xác các bước cần phải thực hiện đối với từng hoạt động. Với từng hoạt động, đưa các hình hoạt họa mô tả những thứ liên quan với từng hoạt động, bao gồm cả số tester và ai làm hoạt động đó. Ví dụ: Chu trình quản lý bug - bao gồm process log 1 bug mới sẽ bao gồm, log ở đâu, log bug mới như nào, trạng thái bug đặt là gì, ai sẽ phân loại bug, sau đó assign bug cho ai.

**3. Môi trường Test.** Môi trường test bao gồm thông tin về số môi trường và các cài đặt cần thiết cho từng loại môi trường. Ví dụ: 1 môi trường test A dành cho test chức năng, 1 môi trường B dành cho UAT. Cần phải xác định số user test trên mỗi môi trường, quyền truy cập của user, yêu cầu về phần cứng và phần mềm như hệ điều hành, bộ nhớ, dung lượng trống... Định nghĩa yêu cầu tạo test data cũng khá quan trọng. Trong TS, đưa các chỉ dẫn về việc tạo test data một cách rõ ràng. Cuối cùng, cần lập một kế hoạch back up và restore để đảm bảo không bị mất dữ liệu trong trường hợp gặp vấn đề trong khi code.

**4. Công cụ test.** Xác định các công cụ quản lý test và automation nếu có. Với các loại test hiệu năng như performance, load và security testing, miêu tả cách tiếp cận và các công cụ cần thiết để thực thi. Ghi chú luôn nếu công cụ đó là nguồn mở hay là phần mềm thương mại, ai có thể hỗ trợ sử dụng chúng.

**5. Kiểm soát việc release.** Việc release mà không có kế hoạch từ trước sẽ dẫn tới nhiều bất cập: sẽ có quá nhiều phiên bản phần mềm khác nhau và không thể test và quản lý hết được. Cần thiết lập kế hoạch/công cụ/qui trình quản lý các bản release này sẽ đảm bảo tất cả các bản build được kiểm tra trước khi release cho Khách hàng.

**6. Phân tích hiểm họa .** Liệt kê hết tất cả hiểm họa mà bạn có thể nghĩ ra. Cung cấp kế hoạch rõ ràng để đối ứng và giảm bớt độ thiệt hại nếu xảy ra với dự án.

**7. Xem xét và chấp thuận.** Khi tất cả những hoạt động được liệt kê trong TS rồi, nó cần được xem xét lại bởi các cá nhân có liên quan như PM, bộ phận dev, admin hệ thống. Nếu có bất kì sự chỉnh sửa nào với tài liệu này, nó cần được ghi vào ngay đầu tài liệu, kèm với tên người approve, thời gian chỉnh sửa, và ghi chú nếu có. Test Strategy là tài liệu sống, tức là nó luôn cần được xem xét liên tục và cập nhật.

**Tổng kết:** TS là sự phản ánh hoạt động của tester trong qui trình kiểm thử phần mềm. Chúng ta nên tham khảo tài liệu này khi test và theo sát kế hoạch cho tới khi release. Khi dự án sát ngày release với Khách hàng, ta thường ngó lơ những gì mà mình đã vạch ra trong TS. Nhưng ta cần thảo luận với dự án xem việc bỏ đi hoạt động test nào đó có gây ra hiểm họa hay vấn đề nào sau release không. Hầu hết các dự án theo mô hình Agile đều bỏ bỏ qua việc viết TS, vì họ tập trung vào thực thi test hơn là việc viết tài liệu. Nhưng việc đề ra một tài liệu TS đơn giản sẽ giúp đề ra kế hoạch rõ ràng, giảm thiểu các hiểm họa có thể xảy ra với dự án.

Tài liệu :<https://viblo.asia/p/7-buoc-de-tao-thanh-cong-mot-test-strategy-OeVKBwQyZkW>