**Phùng Tuấn Minh Stt 18**

**Câu 1:**

**Kiểm thử phần mềm là gì**

**Kiểm thử phần mềm** là một phương pháp để kiểm tra xem sản phẩm phần mềm thực tế có phù hợp với yêu cầu mong đợi hay không và để đảm bảo rằng sản phẩm phần mềm không có [lỗi](https://www.guru99.com/defect-management-process.html) . Nó liên quan đến việc thực thi các thành phần phần mềm/hệ thống bằng cách sử dụng các công cụ thủ công hoặc tự động để đánh giá một hoặc nhiều thuộc tính quan tâm. Mục đích của kiểm thử phần mềm là xác định các lỗi, lỗ hổng hoặc các yêu cầu còn thiếu so với yêu cầu thực tế.

Một số thích nói định nghĩa kiểm thử phần mềm là Kiểm thử [hộp trắng](https://www.guru99.com/white-box-testing.html) và [Hộp đen](https://www.guru99.com/black-box-testing.html) . Nói một cách đơn giản, Kiểm thử phần mềm có nghĩa là Xác minh ứng dụng đang được kiểm thử (AUT). Khóa học Kiểm thử phần mềm này giới thiệu phần mềm kiểm thử cho khán giả và chứng minh tầm quan trọng của kiểm thử phần mềm.

Tại sao kiểm thử phần mềm lại quan trọng?

Kiểm thử phần mềm rất quan trọng vì nếu có bất kỳ lỗi hoặc lỗi nào trong phần mềm, nó có thể được xác định sớm và có thể được giải quyết trước khi chuyển giao sản phẩm phần mềm. Sản phẩm phần mềm được kiểm tra đúng cách đảm bảo độ tin cậy, bảo mật và hiệu suất cao, giúp tiết kiệm thời gian, hiệu quả chi phí và sự hài lòng của khách hàng.

Sự cần thiết của Kiểm thử là gì?

Kiểm tra là quan trọng vì lỗi phần mềm có thể tốn kém hoặc thậm chí nguy hiểm. Lỗi phần mềm có khả năng gây ra tổn thất về tiền bạc và con người, và lịch sử có rất nhiều ví dụ như vậy.

* Vào tháng 4 năm 2015, thiết bị đầu cuối của Bloomberg ở London đã gặp sự cố do trục trặc phần mềm, ảnh hưởng đến hơn 300.000 nhà giao dịch trên thị trường tài chính. Nó buộc chính phủ phải hoãn việc bán khoản nợ 3 tỷ bảng Anh.
* Xe Nissan triệu hồi hơn 1 triệu xe trên thị trường do lỗi phần mềm cảm biến cảm biến túi khí. Đã có báo cáo về hai vụ tai nạn do lỗi phần mềm này.

Lợi ích của Kiểm thử phần mềm là gì?

Dưới đây là những lợi ích của việc sử dụng kiểm thử phần mềm:

* Hiệu quả về chi phí: Đây là một trong những lợi thế quan trọng của kiểm thử phần mềm. Thử nghiệm bất kỳ dự án CNTT nào đúng hạn sẽ giúp bạn tiết kiệm tiền trong thời gian dài. Trong trường hợp nếu các lỗi bắt gặp trong giai đoạn kiểm thử phần mềm trước đó, thì sẽ tốn ít chi phí hơn để sửa.
* Bảo mật: Đây là lợi ích dễ bị tổn thương và nhạy cảm nhất của kiểm thử phần mềm. Mọi người đang tìm kiếm các sản phẩm đáng tin cậy. Nó giúp loại bỏ rủi ro và các vấn đề sớm hơn.
* Chất lượng sản phẩm: Là yêu cầu tất yếu của bất kỳ sản phẩm phần mềm nào. Kiểm tra đảm bảo một sản phẩm chất lượng được cung cấp cho khách hàng.
* Sự hài lòng của khách hàng: Mục đích chính của bất kỳ sản phẩm nào là mang lại sự hài lòng cho khách hàng của họ. Kiểm tra UI/UX đảm bảo trải nghiệm người dùng tốt nhất.

## Kiểm thử trong Kỹ thuật phần mềm

Theo ANSI/IEEE 1059, **Kiểm thử trong Công nghệ phần mềm** là một quá trình đánh giá một sản phẩm phần mềm để tìm xem liệu sản phẩm phần mềm hiện tại có đáp ứng các điều kiện được yêu cầu hay không. Quá trình thử nghiệm liên quan đến việc đánh giá các tính năng của sản phẩm phần mềm cho các yêu cầu về bất kỳ yêu cầu nào còn thiếu, lỗi hoặc lỗi, bảo mật, độ tin cậy và hiệu suất.

Các loại kiểm thử phần mềm

Dưới đây là các loại kiểm thử phần mềm:

Thông thường Kiểm thử được phân thành ba loại.

* Thử nghiệm chức năng
* Kiểm thử phi chức năng hoặc [kiểm thử hiệu suất](https://www.guru99.com/performance-testing.html)
* Bảo trì (Hồi quy và Bảo trì)

## Chiến lược kiểm thử trong công nghệ phần mềm

Dưới đây là các chiến lược quan trọng trong công nghệ phần mềm:

**Kiểm tra đơn vị:** Cách tiếp cận cơ bản kiểm tra phần mềm này được lập trình viên tuân theo để kiểm tra đơn vị của chương trình. Nó giúp các nhà phát triển biết liệu từng đơn vị mã có hoạt động bình thường hay không.

**Kiểm thử tích hợp:** Nó tập trung vào việc xây dựng và thiết kế phần mềm. Bạn cần xem các thiết bị tích hợp có hoạt động tốt không lỗi hay không.

**Kiểm tra hệ thống:** Trong phương pháp này, phần mềm của bạn được biên dịch tổng thể và sau đó được kiểm tra tổng thể. Chiến lược thử nghiệm này kiểm tra chức năng, bảo mật, tính di động, trong số những thứ khác.

## Kiểm tra chương trình

**Kiểm thử chương trình** trong kiểm thử phần mềm là một phương pháp thực thi một chương trình phần mềm thực tế với mục đích kiểm tra hành vi của chương trình và tìm lỗi. Chương trình phần mềm được thực thi với dữ liệu trường hợp thử nghiệm để phân tích hành vi hoặc phản hồi của chương trình đối với dữ liệu thử nghiệm. Một chương trình thử nghiệm tốt là một chương trình có cơ hội tìm ra lỗi cao.

Câu 2.2(18/4)

**7 nguyên tắc kiểm thử phần mềm**

1. Thử nghiệm cho thấy sự hiện diện của lỗi
2. Thử nghiệm toàn diện là không thể
3. thử nghiệm sớm
4. phân cụm lỗi
5. Nghịch lý thuốc trừ sâu
6. Thử nghiệm phụ thuộc vào ngữ cảnh
7. Không có sai lầm ngụy biện

## 1) Không thể thử nghiệm toàn diện

Đúng! Thử nghiệm toàn diện là không thể. Thay vào đó, chúng tôi cần số lượng thử nghiệm tối ưu dựa trên đánh giá rủi ro của ứng dụng.

Và câu hỏi triệu đô là, làm thế nào để bạn xác định rủi ro này?

Để trả lời điều này chúng ta hãy làm một bài tập

Theo bạn, Hoạt động nào có nhiều khả năng khiến Hệ điều hành của bạn bị lỗi?

Tôi chắc rằng hầu hết các bạn sẽ đoán được, Mở 10 ứng dụng khác nhau cùng một lúc.

Vì vậy, nếu bạn đang thử nghiệm Hệ điều hành này, bạn sẽ nhận ra rằng các lỗi có thể được tìm thấy trong hoạt động đa tác vụ và cần được kiểm tra kỹ lưỡng, điều này đưa chúng ta đến nguyên tắc tiếp theo là Phân cụm [lỗi](https://www.guru99.com/defect-management-process.html)

## 2) Phân cụm lỗi

Phân cụm lỗi cho biết một số lượng nhỏ các mô-đun chứa hầu hết các lỗi được phát hiện. Đây là ứng dụng của Nguyên tắc Pareto để kiểm thử phần mềm: khoảng 80% vấn đề được tìm thấy trong 20% ​​mô-đun.

Bằng kinh nghiệm, bạn có thể xác định các mô-đun rủi ro như vậy. Nhưng cách tiếp cận này có vấn đề riêng của nó

Nếu các bài kiểm tra giống nhau được lặp đi lặp lại nhiều lần, cuối cùng các trường hợp kiểm tra tương tự sẽ không còn tìm thấy lỗi mới.

## 3) Nghịch lý thuốc trừ sâu

Việc sử dụng lặp đi lặp lại cùng một hỗn hợp thuốc trừ sâu để diệt trừ côn trùng trong quá trình canh tác, theo thời gian sẽ dẫn đến việc côn trùng phát triển khả năng kháng thuốc, do đó thuốc trừ sâu sẽ mất tác dụng đối với côn trùng. Điều tương tự cũng áp dụng cho kiểm thử phần mềm. Nếu cùng một tập hợp các kiểm thử lặp đi lặp lại được tiến hành, phương pháp này sẽ vô dụng để phát hiện ra các lỗi mới.

Để khắc phục điều này, các trường hợp thử nghiệm cần được xem xét và sửa đổi thường xuyên, thêm các trường hợp thử nghiệm mới & khác nhau để giúp tìm ra nhiều lỗi hơn.

Người kiểm thử không thể đơn giản phụ thuộc vào các kỹ thuật kiểm thử hiện có. Anh ta phải liên tục tìm cách cải thiện các phương pháp hiện có để làm cho việc kiểm tra hiệu quả hơn. Nhưng ngay cả sau tất cả mồ hôi và công sức thử nghiệm này, bạn không bao giờ có thể khẳng định rằng sản phẩm của mình không có lỗi. Để hiểu rõ điểm này, chúng ta hãy xem video này về buổi ra mắt công khai của Windows 98

Bạn nghĩ rằng một công ty như MICROSOFT sẽ không kiểm tra kỹ lưỡng hệ điều hành của họ và sẽ mạo hiểm danh tiếng của họ chỉ để thấy hệ điều hành của họ gặp sự cố khi ra mắt công chúng!

## 4) Thử nghiệm cho thấy sự hiện diện của lỗi

Do đó, nguyên tắc kiểm thử nói rằng – Kiểm thử nói về sự hiện diện của lỗi chứ không nói về việc không có lỗi. tức là Kiểm thử phần mềm làm giảm xác suất lỗi chưa được phát hiện còn lại trong phần mềm nhưng ngay cả khi không tìm thấy lỗi nào, đó không phải là bằng chứng về tính đúng đắn.

Nhưng điều gì sẽ xảy ra nếu bạn làm việc chăm chỉ hơn, thực hiện mọi biện pháp phòng ngừa và làm cho sản phẩm phần mềm của bạn không có lỗi 99%. Và phần mềm không đáp ứng được nhu cầu & yêu cầu của khách hàng.

Điều này dẫn chúng ta đến nguyên tắc tiếp theo, nguyên tắc này khẳng định rằng- Không có Lỗi

## 5) Không có lỗi – ngụy biện

Có thể phần mềm không có lỗi 99% vẫn không sử dụng được. Đây có thể là trường hợp nếu hệ thống được kiểm tra kỹ lưỡng cho yêu cầu sai. Kiểm thử phần mềm không chỉ đơn thuần là tìm lỗi mà còn để kiểm tra xem phần mềm có đáp ứng được nhu cầu kinh doanh hay không. Việc không có Lỗi là Sai lầm, tức là Việc tìm và sửa các lỗi không giúp ích gì nếu bản dựng hệ thống không sử dụng được và không đáp ứng nhu cầu & yêu cầu của người dùng.

Để giải quyết vấn đề này, nguyên tắc thử nghiệm tiếp theo nói rằng Thử nghiệm sớm

## 6) Kiểm tra sớm

Kiểm thử sớm – Kiểm thử nên bắt đầu càng sớm càng tốt trong Vòng đời phát triển phần mềm. Vì vậy, bất kỳ lỗi nào trong giai đoạn yêu cầu hoặc thiết kế đều được nắm bắt trong giai đoạn đầu. Sửa lỗi trong giai đoạn đầu của quá trình thử nghiệm sẽ rẻ hơn nhiều. Nhưng một người nên bắt đầu thử nghiệm sớm như thế nào? Bạn nên bắt đầu tìm lỗi ngay khi các yêu cầu được xác định. Thông tin thêm về nguyên tắc này trong phần hướng dẫn đào tạo sau.

## 7) Thử nghiệm phụ thuộc vào ngữ cảnh

Thử nghiệm phụ thuộc vào ngữ cảnh, về cơ bản có nghĩa là cách bạn thử nghiệm một trang web thương mại điện tử sẽ khác với cách bạn thử nghiệm một ứng dụng thương mại bán chạy. Tất cả các phần mềm được phát triển không giống nhau. Bạn có thể sử dụng một cách tiếp cận, phương pháp, kỹ thuật và loại thử nghiệm khác nhau tùy thuộc vào loại ứng dụng. Ví dụ: thử nghiệm, bất kỳ hệ thống POS nào tại cửa hàng bán lẻ sẽ khác với thử nghiệm máy ATM.

Câu 3.3(18/5)

## Scrum trong kiểm thử phần mềm

**Scrum trong Kiểm thử phần mềm** là một phương pháp để xây dựng các ứng dụng phần mềm phức tạp. Nó cung cấp các giải pháp dễ dàng để thực hiện các tác vụ phức tạp. Scrum giúp nhóm phát triển tập trung vào tất cả các khía cạnh của quá trình phát triển sản phẩm phần mềm như chất lượng, hiệu suất, khả năng sử dụng, v.v. Nó cung cấp tính minh bạch, kiểm tra và thích ứng trong quá trình phát triển phần mềm để tránh sự phức tạp.

## Kiểm thử Scrum

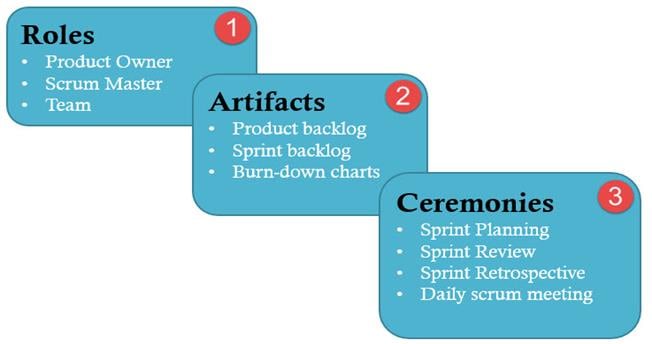
**Scrum Testing** là một thử nghiệm được thực hiện theo phương pháp scrum để xác minh các yêu cầu ứng dụng phần mềm được đáp ứng. Nó liên quan đến việc kiểm tra các tham số phi chức năng như bảo mật, khả năng sử dụng, hiệu suất, v.v. Người kiểm tra không có vai trò tích cực trong quy trình nên nó thường được thực hiện bởi các nhà phát triển với Kiểm tra đơn vị. Đôi khi cần có các nhóm kiểm thử chuyên dụng tùy thuộc vào tính chất và độ phức tạp của dự án.

## Các tính năng chính của phương pháp Scrum

Sau đây là các tính năng chính của Scrum-

* Scrum có một lịch trình cố định ngắn cho các chu kỳ phát hành với phạm vi có thể điều chỉnh được gọi là **chạy nước rút** để giải quyết các nhu cầu phát triển thay đổi nhanh chóng. Mỗi bản phát hành có thể có nhiều lần chạy nước rút. Mỗi Dự án Scrum có thể có nhiều Chu kỳ Phát hành.
* Một chuỗi **các cuộc họp, sự kiện và sự kiện lặp đi lặp lại**
* Thực hành thử nghiệm và thực hiện các yêu cầu mới, được gọi là **câu chuyện** , để đảm bảo một số công việc được phát hành sau mỗi lần chạy nước rút

Scrum dựa trên 3 Trụ cột sau-



Hãy xem xét từng cái một

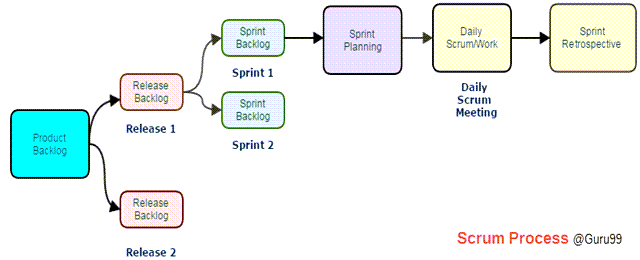
### 1. Vai trò trong Scrum

Có ba vai trò chính trong Thử nghiệm Scrum - Chủ sở hữu sản phẩm, Scrum Master và Nhóm phát triển. Hãy nghiên cứu chúng một cách chi tiết

NoSQL Database Tutorial02:00Important Linux Commands for Beginners Linux Tutorial15:03

| **Chủ sở hữu sản phẩm** | **Đội sản xuất** | **Đội** |
| --- | --- | --- |
| Anh ấy / Cô ấy xác định các tính năng của sản phẩm. | Anh ấy / Cô ấy quản lý nhóm và chăm sóc năng suất của nhóm | Team thường khoảng 5-9 thành viên |
| Product Owner quyết định ngày phát hành và các tính năng tương ứng | Anh ấy / Cô ấy duy trì danh sách chặn và loại bỏ các rào cản trong quá trình phát triển | Nó bao gồm các nhà phát triển, nhà thiết kế và đôi khi là người thử nghiệm, v.v. |
| Họ ưu tiên các tính năng theo giá trị thị trường và lợi nhuận của sản phẩm | Anh ấy / Cô ấy phối hợp với tất cả các vai trò và chức năng | Nhóm tự tổ chức và lên lịch làm việc |
| Anh ấy / Cô ấy chịu trách nhiệm về lợi nhuận của sản phẩm | Anh ấy / Cô ấy bảo vệ đội khỏi sự can thiệp từ bên ngoài | Có quyền làm mọi thứ trong ranh giới của dự án để đáp ứng mục tiêu nước rút |
| Anh ấy / Cô ấy có thể chấp nhận hoặc từ chối kết quả hạng mục công việc | Mời tham dự cuộc họp Scrum hàng ngày, đánh giá Sprint và lập kế hoạch | Tích cực tham gia các nghi lễ hàng ngày |
|  |  |  |

### 2. Tạo tác Scrum



Một quy trình scrum bao gồm

* **Câu chuyện của người dùng:** Chúng là một lời giải thích ngắn gọn về các chức năng của hệ thống đang được thử nghiệm. Ví dụ đối với Nhà cung cấp bảo hiểm là – “Có thể thanh toán phí bảo hiểm bằng hệ thống trực tuyến.”
* **Product Backlog:** Đây là tập hợp các câu chuyện của người dùng được ghi lại cho một sản phẩm scrum. **Product Owner chuẩn bị** và duy trì product backlog. Nó được ưu tiên bởi chủ sở hữu sản phẩm và bất kỳ ai cũng có thể thêm vào nó với sự chấp thuận của chủ sở hữu sản phẩm.
* **Release Backlog:** Một lần phát hành là một khung thời gian trong đó số lần lặp lại được hoàn thành. **Chủ sở hữu sản phẩm phối hợp** với chủ scrum để quyết định những câu chuyện nào sẽ được nhắm mục tiêu để phát hành. Các câu chuyện trong bản phát hành tồn đọng được nhắm mục tiêu hoàn thành trong một bản phát hành.
* **Sprints:** Đó là một khoảng thời gian nhất định để hoàn thành các câu chuyện của người dùng, do chủ sở hữu sản phẩm và nhóm nhà phát triển quyết định, thường là 2-4 tuần.
* **Sprint Backlog:** Đó là một tập hợp các câu chuyện của người dùng sẽ được hoàn thành trong một lần chạy nước rút. Trong quá trình chạy nước rút tồn đọng, công việc không bao giờ được giao và nhóm tự đăng ký làm việc. Nó được sở hữu và quản lý bởi nhóm trong khi công việc ước tính còn lại được cập nhật hàng ngày. Đây là danh sách các nhiệm vụ phải được thực hiện trong Sprint
* **Danh sách chặn:** Đây là danh sách các khối và các quyết định chưa được thực hiện do chủ scrum sở hữu và được cập nhật hàng ngày
* **Biểu đồ Burndown:** Biểu đồ Burndown thể hiện tổng thể tiến độ công việc đang thực hiện và công việc đã hoàn thành trong suốt quá trình. Nó thể hiện ở định dạng biểu đồ các câu chuyện và tính năng chưa hoàn thành

### 3. Nghi lễ (Quy trình) trong Scrum

* **Lập kế hoạch Sprint:** Một Sprint bắt đầu với việc nhóm nhập các câu chuyện từ bản phát hành tồn đọng vào bản ghi của Sprint; nó được lưu trữ bởi chủ scrum. Người kiểm tra ước tính nỗ lực để kiểm tra các câu chuyện khác nhau trong Sprint Backlog.
* **Daily Scrum:** Nó được tổ chức bởi scrum master, nó kéo dài khoảng 15 phút. Trong Scrum Hằng ngày, các thành viên sẽ thảo luận về công việc đã hoàn thành của ngày hôm trước, công việc đã lên kế hoạch cho ngày hôm sau và các vấn đề phải đối mặt trong một lần chạy nước rút. Trong cuộc họp đứng hàng ngày, tiến trình của nhóm được theo dõi.
* **Sprint Review/ Retrospective:** Nó cũng được tổ chức bởi scrum master, kéo dài khoảng 2-4 giờ và thảo luận về những gì nhóm đã hoàn thành trong lần chạy nước rút trước và những bài học rút ra được.

## Vai trò của Tester trong Scrum



**Không có vai trò tích cực của Tester trong** Quy trình Scrum. Thông thường, kiểm thử được thực hiện bởi nhà phát triển với Kiểm thử đơn vị. Trong khi chủ sở hữu sản phẩm cũng thường xuyên tham gia vào quá trình thử nghiệm trong mỗi lần chạy nước rút. **Một số dự án Scrum có các nhóm kiểm thử riêng tùy thuộc vào tính chất và độ phức tạp của dự án** .

Câu hỏi tiếp theo là, người kiểm thử làm gì trong một cuộc đánh nhau? Note sau sẽ giải đáp

## Các hoạt động kiểm thử trong Scrum

Người kiểm thử thực hiện các hoạt động sau trong các giai đoạn khác nhau của Scrum-

**Kế hoạch nước rút**

* Trong lập kế hoạch chạy nước rút, người kiểm thử nên chọn một câu chuyện của người dùng từ sản phẩm tồn đọng cần được kiểm thử.
* Là người thử nghiệm, anh ấy/cô ấy nên quyết định cần bao nhiêu giờ (Ước tính nỗ lực) để **hoàn thành** thử nghiệm cho từng câu chuyện của người dùng đã chọn.
* Là một tester, anh ấy/cô ấy phải biết mục tiêu chạy nước rút là gì.
* Là người thử nghiệm, hãy đóng góp vào quá trình ưu tiên

**tăng tốc**

* Hỗ trợ các nhà phát triển trong thử nghiệm đơn vị
* Kiểm tra user-story khi hoàn thành. **Việc thực hiện kiểm thử được thực hiện** trong phòng thí nghiệm nơi cả người kiểm thử và nhà phát triển cùng làm việc với nhau. Lỗi được ghi vào công cụ Quản lý lỗi được theo dõi hàng ngày. Các lỗi có thể được đưa ra và phân tích trong cuộc họp scrum. Các lỗi được kiểm tra lại ngay sau khi nó được **giải quyết** và triển khai để kiểm tra
* Là một người thử nghiệm, anh ấy / cô ấy tham dự tất cả các cuộc họp đứng hàng ngày để lên tiếng
* Với tư cách là người kiểm thử, anh/cô ấy có thể mang bất kỳ hạng mục tồn đọng nào không thể hoàn thành trong lần chạy nước rút hiện tại và đưa vào lần chạy nước rút tiếp theo
* Tester chịu trách nhiệm phát triển các kịch bản tự động hóa. Anh ấy lên lịch thử nghiệm tự động hóa với [hệ thống Tích hợp liên tục (CI)](https://www.guru99.com/continuous-integration.html) . Tự động hóa nhận được tầm quan trọng do thời gian giao hàng ngắn. Tự động hóa thử nghiệm có thể được thực hiện bằng cách sử dụng các công cụ nguồn mở hoặc trả phí khác nhau có sẵn trên thị trường. Điều này chứng tỏ hiệu quả trong việc đảm bảo rằng mọi thứ cần kiểm tra đều được bảo hiểm. Có thể đạt được phạm vi kiểm tra đầy đủ khi giao tiếp chặt chẽ với nhóm.
* Xem xét kết quả tự động hóa CI và gửi Báo cáo cho các bên liên quan
* Thực hiện kiểm tra phi chức năng cho các câu chuyện của người dùng đã được phê duyệt
* Phối hợp với khách hàng và chủ sở hữu sản phẩm để xác định tiêu chí chấp nhận cho Thử nghiệm chấp nhận
* Vào cuối sprint, tester cũng thực hiện kiểm thử chấp nhận (UAT) trong một số trường hợp và xác nhận tính đầy đủ của kiểm thử cho sprint hiện tại

**Cải tiến nước rút**

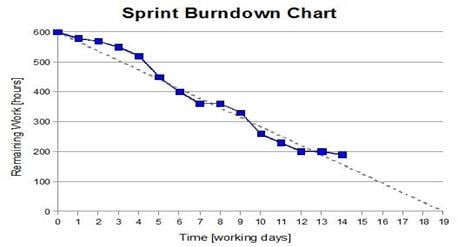
* Là một tester, anh ta sẽ tìm ra điều gì sai và điều gì đúng trong Sprint hiện tại
* Là người thử nghiệm, anh ấy xác định bài học kinh nghiệm và các phương pháp hay nhất

## Báo cáo thử nghiệm

Báo cáo số liệu Kiểm tra Scrum cung cấp tính minh bạch và khả năng hiển thị cho các bên liên quan về dự án. Các số liệu được báo cáo cho phép một nhóm phân tích tiến độ của họ và lập kế hoạch chiến lược trong tương lai để cải thiện sản phẩm. Có hai số liệu thường được sử dụng để báo cáo.

**Biểu đồ ghi lại:** Mỗi ngày, Scrum Master ghi lại công việc ước tính còn lại cho lần chạy nước rút. Đây không là gì ngoài Biểu đồ Burn Down. Nó được cập nhật hàng ngày.

Biểu đồ burndown cung cấp tổng quan nhanh về tiến độ dự án, biểu đồ này chứa thông tin như tổng khối lượng công việc trong dự án phải hoàn thành, khối lượng công việc đã hoàn thành trong mỗi lần chạy nước rút, v.v.



**Biểu đồ lịch sử vận ​​tốc:** Biểu đồ lịch sử vận ​​tốc dự đoán vận tốc nhóm đạt được trong mỗi lần chạy nước rút. Nó là một biểu đồ thanh và thể hiện cách đầu ra của các đội đã thay đổi theo thời gian.

Các số liệu bổ sung có thể hữu ích là đốt cháy lịch biểu, đốt cháy ngân sách, hoàn thành phần trăm chủ đề, câu chuyện đã hoàn thành – câu chuyện còn lại, v.v.

Đây là tất cả về Scrum trong công nghệ phần mềm

Câu 4.2(18/8)

**2. Kiểm tra khả năng sử dụng :**

[Kiểm tra khả năng sử dụng](https://www.guru99.com/usability-testing-tutorial.html) hiện đã trở thành một phần quan trọng của bất kỳ dự án dựa trên web nào. Nó có thể được **thực hiện bởi những người thử nghiệm** như bạn **hoặc một nhóm tập trung nhỏ** tương tự như đối tượng mục tiêu của ứng dụng web.

**Kiểm tra** trang web **Điều hướng** :

* Menu, nút hoặc Liên kết đến các trang khác nhau trên trang web của bạn phải dễ dàng nhìn thấy và nhất quán trên tất cả các trang web

**Kiểm tra** nội **dung** :

* Nội dung phải rõ ràng, không có lỗi chính tả hoặc ngữ pháp.
* Hình ảnh nếu có phải chứa văn bản “alt”