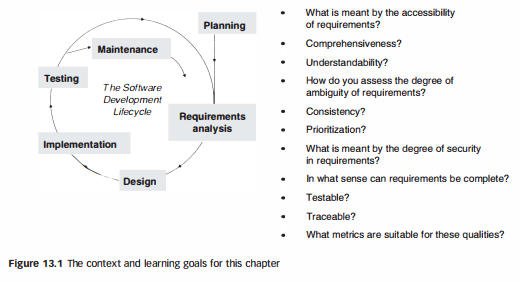
**CHƯƠNG 13**

**CHẤT LƯỢNG VÀ SỐ LIỆU TRONG PHÂN TÍCH YÊU CẦU**

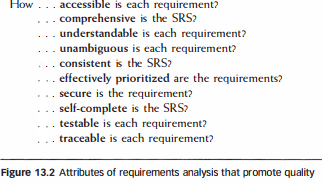


***Hình 13.1 Bối cảnh và mục đích học của chương này***

- Khả năng tiếp cận các yêu cầu có ý nghĩa gì?

* Tính toàn diện, bao hàm của các yêu cầu?
* Tính dễ hiểu của các yêu cầu?
* Đánh giá mức độ mơ hồ của các yêu cầu?
* Tính nhất quán của các yêu cầu?
* Sự ưu tiên của các yêu cầu?
* Có thể kiểm tra các yêu cầu?
* Mức độ bảo mật có ý nghĩa gì trong các yêu cầu?
* Theo nghĩa nào thì có thể các yêu cầu được hoàn thành?
* Những số liệu nào phù hợp cho những chất lượng này?
* Tính theo dõi, giám sát trong các yêu cầu?

Chương này mô tả các thước đo chất lượng trong yêu cầu. Càng nhiều tài liệu yêu cầu thể hiện rõ những gì khách hàng muốn và cần thì chất lượng của nó càng cao hơn. Chúng ta thường nghĩ rằng chi tiết thường ít quan trọng hơn "bức tranh lớn", nhưng việc thiếu một chi tiết về yêu cầu có thể gây ảnh hưởng nghiêm trọng đến với dự án, như nhiều trường hợp nghiên cứu đã chứng minh. Hồi tưởng lại , ví dụ điển hình là vụ bỏ qua chi tiết của số liệu chuyển đổi khoảng cách đã đưa một con tàu vũ trụ trị giá 125 triệu đô đi vào quên lãng.



***Hình 13.2 Các thuộc tính của việc phân tích các yêu cầu nhằm thúc đẩy chất lượng:***

* Mỗi yêu cầu có thể truy cập như thế nào?
* Sự toàn diện của đặc tả yêu cầu phần mềm là như thế nào?(SRS)
* Khả năng hiểu rõ của mỗi yêu cầu là như thế nào?
* Tính rõ ràng trong mỗi yêu cầu là như thế nào??
* Tính nhất quán đối với đặc tả yêu cầu phần mềm là như thế nào?
* Ưu tiên hiệu quả là những yêu cầu?
* Yêu cầu an toàn như thế nào?

Để giúp đảm bảo rằng tất cả những yêu cầu cần thiết được bảo vệ, chúng ta tập trung vào những chât lượng mà các yêu cầu nên sỡ hữu. Các yêu cầu nên đầy đủ và nhất quán, mỗi người nên có khả năng vạch ra các thiết kế và thực hiện, kiểm tra tính hợp lệ, và thực hiện theo mức độ ưu tiên hợp lý. Liệt kê các thuộc tính này và cho chúng ta biết những gì nên tìm kiếm trong các yêu cầu tốt. Chúng ta có thể xem xét nó một cách có hệ thống và kiểm tra các yêu cầu dựa trên danh sách này. Ví dụ: một tập hợp các yêu cầu nhất quán có nhiều khả năng thể hiện những gì mà các bên liên quan muốn và cần hơn một danh sách mâu thuẫn.

Chương này thảo luận về cách một trong số những chất lượng được xác định như thế nào. "Giá trị mục tiêu" thảm khảo các mục tiêu số trong dự án liên quan đến nhiều số liệu khác nhau. Số liệu có hiệu quả nhất khi mục đích giá trị của chúng được đặc tả trước. ví dụ: chúng ta có thể nói trước rằng, dựa vào kinh nghiệm của những dự án trước, các yêu cầu được xem như hoàn thành khi tỷ lệ hiệu chỉnh và bổ sung dưới 1% trong tuần.

Các dự án được cải thiện rất nhiều khi tổ chức QA tham gia vào giai đoạn phân tích yêu cầu. Cụ thể là QA xác thực rằng quy trinh đang được thực hiện theo đúng kế hoạch. QA nên tham gia vào việc kiểm tra các tài liệu yêu cầu. Họ thường có những quan điểm tốt bởi vì họ hiểu rằng phải xác nhận sản phẩm dựa trên các yêu cầu. Trong các dự án tổ chức kém, QA có thể được đưa 1 ứng dụng vs ít hoặc thậm chí không có các tài liệu yêu cầu và được yêu cầu phải kiểm tra nó. Điều này đặt ra câu hỏi"ứng dụng này phải làm gì?

**13.1 Yêu cầu chất lượng đối với một dự án Alige**

Trước khi thảo luận về các thuộc tính của việc phân tích chất lượng các yêu cầu được liệt kê ở trên, chúng ta hãy thảo luận về chất lượng của các yêu cầu đối với quy trình Alige. Quy trình chính cho các yêu cầu ở đây bao gồm việc gợi ý những câu chuyện của người dùng từ khách hàng, cùng với những thử nghiệm được chấp nhận và sau đó thực hiện các thử nghiệm đó cho đến khi hoàn thành. Hơn nữa, khách hàng phải cảm thấy hài lòng với kết quả này. Điều này có thể có hoặc không được hỗ trợ bởi các tài liệu quan trọng. chất lượng phải được đánh giá, nếu không thì phải được xác định theo tiêu chuẩn này. Điều này đòi hỏi phải tính toán tỷ lệ các phép thử đã được thông qua và đánh giá phản ứng của khách hàng, có thể thông qua bảng câu hỏi. Vì khách hàng - thường đại diện thành viên của nhóm là một phần của những nổ lực phát triển, đánh giá yêu cầu bao gồm sự thể hiện của khách hàng. Với tính chất của việc phân tích yêu cầu, thì đây đều là những điều tốt

* 1. **Tiếp cận yêu cầu**

1. Để giải quyết một loạt các yêu cầu, chúng ta có thể tiếp cận những yêu cầu mà chúng ta muốn. Đây là tính chất của khả năng tiếp cận. Tính chất đầu tiên mà chúng ta cần trong khía cạnh này là một phương tiện để xác định các yêu cầu chi tiết.
2. Chúng ta làm điều này bằng cách đánh số chúng theo một cách nào đó. Một hệ thống đánh số tốt cho phép chúng ta biết được yêu cầu nào đã thực hiện hay chưa và để theo dõi mã thực sự thực hiện nó.
3. Yêu cầu của dự án thay đổi liên tục trong suốt vòng đời của nó. Ví dụ khi lập trình viên cố gắng nhưng không thành công trong việc thực hiện các yêu cầu và giải thích điều này cho khách hàng, sau này thường tìm thấy những phần còn thiếu xót trong yêu cầu. Các đặc tả yêu cầu phải được xem xét một cách chắn chắn rằng các yêu cầu nào còn thiếu hay không và bổ sung nếu chúng không có. Ví dụ trong trường hợp nghiên cứu của cửa hàng video, khách hàng (cửa hàng video) có thể đặt câu hỏi tại sao thời gian phát DVD không xuất hiện trên màn hình. Người phát triển và khách hàng sẽ muốn biết rằng điều này đã có trong đặc tả các yêu cầu của phần mềm hay chưa. Họ nên tìm ở chỗ nào trong tài liệu đặc tả đó? Việc tìm kiếm trong các tài liệu được tổ chức kém thì rất tốn thời gian.
4. Dưới đây là danh sách câu hỏi để cải thiện khả năng tiếp cận yêu cầu:

* Bạn có biết nơi các yêu cầu cao được trình bày?
* Bạn có biết nơi các chi tiết yêu cầu được liệt kê?
* Các yêu cầu chi tiết được tổ chức theo nhóm, vậy tốt nhất thì với mỗi nhóm tương ứng với bao nhiêu yêu cầu cao?
* Tất cả các yêu cầu chi tiết được sắp xếp thành một danh sách hay một danh sách được hiểu rõ ràng của nhiều danh sách khác?
* Bạn có thể tra cứu các yêu cầu theo từ khóa ? theo chủ đề? Theo trường hợp sử dụng? theo giao diện? theo loại người dùng?
* Bạn có thể tra cứu các yêu cầu theo các tiêu chí khác liên quan đến ứng dụng hay dự án cụ thể không?

1. Một số liệu có khả năng tiếp cận là thời gian trung bình thực hiện truy cập một yêu cầu chi tiết. Để xác định điều này, một thử nghiệm sẽ được thực hiện bởi các yêu cầu hiện có và còn thiếu. Theo thống kê thì khoảng 150 là cỡ mẫu tốt. Cỡ mẫu nhỏ hơn tuy không đánh tin cậy nhưng vẫn tốt hơn là không có gì. Khi chọn một mẫu, người ta lựa chọn quá trình ngẫu nhiên như thời gian cho phép
   1. **Tính toàn vẹn của yêu cầu**
2. Một đặc tả các yêu cầu chất lượng sẽ thể hiện tất cả các yêu cầu cho một sản phẩm. Do đó, một số liệu hợp lý sẽ tỷ thuận với tỷ lệ phần trăm các yêu cầu của khách hàng xuất hiện trong đặc tả yêu cầu. Một cách rõ ràng để đảm bảo điều này là để khách hàng xác nhận những yêu cầu đó nhưng đó không phải là vấn đề đơn giản.
3. Tính toàn diện của yêu cầu tạo ra một mục tiêu khó nắm bắt và mơ hồ, sự hoàn thành các yêu cầu là chìa khóa để thành công trong dự án và theo dõi tiến độ thực hiện. mỗi lần lặp lại sẽ làm cho các yêu cầu được toàn diện hơn. Một cách để đối phó với tập hợp các yêu cầu đang phát triển là chứa các yêu cầu đó trong các lần lặp trong tương lai.
4. Các vấn đề trong việc đạt được sự toàn vẹn của yêu cầu

* Không đủ tài nguyên để đáp ứng mong muốn của khách hàng

Giải quyết: ưu tiên sự toàn diện theo từng nhóm yêu cầu

* Khách hàng không thể và sẽ không đọc hết toàn bộ các đặc tả yêu cầu phần mềm

Giải quyết: Làm tài liệu đặc tả yêu cầu dễ dàng để theo dõi.

Dùng một tiêu chuẩn nào đó.

Đọc đặc tả yêu cầu cho khách hàng.

* Sự hạn chế của việc tự kiểm tra

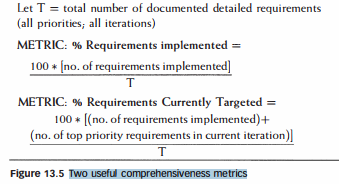
Giải pháp: Kiểm tra ngang hàng

* Sự mâu thuẫn trong yêu cầu của các bên liên quan

Giải pháp: Áp dụng các kỹ năng ngoại giao và mong đợi sự thỏa hiệp

1. Một cách xử lý dễ dàng hơn là tự hoàn thiện, trong đó các yêu cầu chứa đựng tất cả các dữ kiện liên quan đến nó. Dưới đây là danh sách những câu hỏi để cải thiện tính toàn vẹn của yêu cầu:

* Tóm tắt hoặc đưa ra một mô tả sơ bộ ngắn về các yêu cầu cần thiết chưa được đưa vào.
* Khách hàng có hài lòng rằng các yêu cầu đó đáp ứng tất cả các nhu cầu và mong muốn của họ hay không?
* Phần nào trong các yêu cầu được liệt kê sẽ được thực hiện trong bản phát hành hiện tại và trong phát hành tương lai?



Đặt T = tổng số các yêu cầu chi tiết được ghi lại (tất cả ưu tiên và tất cả số lần lặp)

METRIC: % Các yêu cầu được thực hiện = 100 \* (số yêu cầu đã được thực hiện) / T

METRIC: % Các yêu cầu mục tiêu = 100 \* (Số yêu cầu được thực hiện + số yêu cầu ưu tiên hàng đầu trong lần lặp hiện tai) / T

* 1. **Sự hiểu các yêu cầu**

1.Sự hiểu dường như là một chất lượng chủ quan cao bởi vì nó phụ thuộc vào ý kiến của nhiều người. Tuy nhiên nó có thể xác định được.

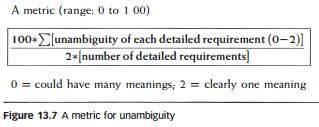
Dưới đây là danh sách kiểm tra để cải thiện tính toàn vẹn của các yêu cầu:

* Những yêu cầu được viết bằng ngôn ngữ mà người thường đọc có thể hiểu được?
* Các yêu cầu có sử dụng từ vựng của miền vấn đề khách hàng không?
* Các yêu chỉ chỉ mô tả các hành vi bên ngoài theo quan điểm của người dùng? Người dùng ở đây bao gồm cả các hệ thống bên ngoài chứ không chỉ riêng con người.
* Các yêu cần có tránh việc nêu cách giải quyết vấn đề, sử dụng các kỹ thuật nào hoặc ứng dụng được thiết kế như thế nào không?
  1. **Sự rõ ràng của yêu cầu**

1. Trừ khi chi tiết của yêu cầu được viết rõ ràng và rành mạch, chúng ta sẽ không thể xác định được yêu cầu đó có được thực hiện đúng hay không.

Dưới đây là danh sách để kiếm tra tính rõ ràng của các yêu cầu:

* Với mỗi yêu cầu, liệu rằng chỉ có một cách mà người đọc có thể diễn tả nó?
* Với mỗi yêu cầu, các thuật ngữ có thể tránh khỏi việc hiểu nhầm theo nhiều cách hay không?



*Hình 13.7 Số liệu cho tính rõ ràng*

Một số liệu có phàm vi từ 0 đến 100

Tính rõ ràng của yêu cầu sẽ bằng 100 nhân với tổng xích ma của sự rõ rang của mỗi yêu cầu chi tiết và đem chia cho 2 nhân với số lượng các yêu cầu chi tiết.

0 có nghĩa là các yêu cầu có thể có rất nhiều nghĩa.

2 có nghĩa là các yêu cầu có duy nhất một nghĩa

# **13.6 Tính thống nhất (nhất quán) yêu cầu.**

Một tập hợp các yêu cầu chi tiết là phù hợp nếu không có sự mâu thuẫn giữa chúng. Khi số lượng yêu cầu chi tiết tăng lên. Sự không nhất quán có xu hướng trở nên khó phát hiện.

* Ví dụ về tính không nhất quán trong yêu cầu
* Yêu cầu 14: Chỉ những mặt hang thực phẩm cơ bản mới được mang theo bởi các nhân vật.
* Yêu cầu 223: Mỗi nhân vật chỉ được mang nước.
* Bột, bơ, sữa và muối được coi là thực phẩm duy nhất.
* Tổ chức những yêu cầu.
* Tổ chức yêu cầu: phong cách tổ chức những yêu cầu theo hướng đối tượng giúp tránh khỏi sự không nhất quán bằng cách chi tiết yêu cầu theo lớp và bằng cách phân tách chúng thành 1 hình thức đơn giản. Tuy nhiên, điều này thì không đảm bảo tính nhất quán, và do đó sự kiểm tra những yêu cầu bao gồm kiểm tra nhất quán.
* Danh sách kiểm tra cải thiện tính nhất quán các yêu cầu.
* Đối với mỗi yêu cầu, có những yêu cầu nào khác có thể dẫn đến mâu thuẫn hoặc hủy bỏ nó không?
* Đối với mỗi yêu cầu, có những yêu cầu khác rất giống nhau và có thể tạo ra sự không nhất quán không?
* Mỗi yêu cầu có tránh được một chuỗi hậu quả không thể thực hiện được không?
* Một số liệu nhất quán là tỷ lệ phần trăm của các chi tiết yêu cầu một phần hoặc hoàn toàn mâu thuẫn ở nơi khác. Để có được như vậy một số liệu, người ta sẽ xem xét một mẫu các yêu cầu chi tiết - 150 mẫu sẽ phù hợp - và điều tra mỗi người lần lượt xác định xem nó có mâu thuẫn ở nơi khác trong tài liệu hay không. Điều này đòi hỏi phải so sánh nó cho tất cả các yêu cầu chi tiết còn lại.
  1. **Ưu tiên các yêu cầu.**

Vì chất lượng cuối cùng được xác định bởi sự hài lòng của khách hàng, quá trình phân tích yêu cầu được liên tục hướng đến khái niệm về sự hài lòng của khách hàng.

+ Các nhóm thường tỏ ra các bên liên quan tạm thời hoàn thành và các bên liên quan sau đó ảnh hưởng đến tiến trình công việc tương ứng.

=> Bởi vì điều này, sự ưu tiên về các yêu cầu - thứ tự thực hiện các yêu cầu, làm cho một sự khác biệt đáng kể trong sự hài lòng của khách hàng.

+ Trong ngôn ngữ toán học, đây là một hoạt động không giao hoán kể từ chuỗi SRS:

Ví dụ: 1. thực hiện yêu cầu A sau đó lên kế hoạch thực hiện yêu cầu B.

cũng có thể sản xuất một sản phẩm khác với

2. thực hiện yêu cầu B sau đó lên kế hoạch để thực hiện yêu cầu A.

***Số liệu để đo lường chất lượng ưu tiên***

Giả sự có 3 độ ưu tiên: cao, trung bình và thấp.  
Số liệu: Variation from # high = # medium = # low  
T = tổng số chi tiết yêu cầu:

0 = tệ nhất; 100 = tốt nhất.

+ Giả sử rằng mỗi yêu cầu là một trong ba ưu tiên. Làm thế nào chúng ta sẽ đo lường chất lượng của sự ưu tiên? Một ưu tiên chất lượng tốt phân loại các yêu cầu thành các phần bằng nhau, chỉ ra rằng không thể loại đã bị bỏ qua.

+ Ví dụ: nếu 900 yêu cầu được ưu tiên rất tốt, mỗi danh mục sẽ chứa 300 yêu cầu,

và công thức sẽ cho 100 \* [900 - 0 - 0 - 0] / 900 = 100%.

\* Mặt khác, nếu 700 được phân loại là mức độ ưu tiên cao, 100 là trung bình và 100 là thấp, số liệu sẽ mang lại

100 \* [900 – |300 – 700| - |300 – 100| - |300 - 100 |]/ 900 = 100 \* 100/900 = 1 1. 1%

Tỷ lệ thấp cho thấy mức độ ưu tiên kém.

* Danh sách kiểm tra để cải thiện mức độ ưu tiên của các yêu cầu:

• Có rõ ràng những yêu cầu nên được làm việc trên đầu tiên? Thứ hai?

• Các ưu tiên ở mức cao có phù hợp với các ưu tiên ở cấp chi tiết không?

• Có một quy trình ưu tiên tại chỗ để nó vẫn rõ ràng yêu cầu quan trọng tiếp theo cần được thực hiện là gì?

• Ưu tiên có phù hợp với mong đợi của khách hàng không?

• Ưu tiên có phù hợp với rủi ro dự án không?

# **13.8 Yêu cầu bảo mật và cấp cao**

* Bảo mật có thể được xử lý như một yêu cầu thực tế (rõ ràng) hoặc như một thuộc tính của các yêu cầu.
* Bảo mật trong các yêu cầu là một trường hợp đặc biệt ở chỗ nó liên quan đến việc khai thác có chủ ý từ phía người khác để lạm dụng nó. Các yêu cầu truyền thống, xét cho cùng, được dự định để xác định những gì một ứng dụng nên làm và không giải quyết việc sử dụng sai.
* Điều này đang thay đổi. Ở một mức độ ngày càng tăng, các tài liệu yêu cầu đang giải quyết vấn đề bảo mật bằng cách bao gồm các nội dung như các trường hợp sử dụng sai. Đây là tương tự như ý tưởng, được đề cập trước đó, nghịch đảo yêu cầu. Các ví dụ về các trường hợp sử dụng sai mà hệ thống yêu cầu không cho phép:

+ Một người dùng ứng dụng tự động nhập vào một User ID đã biết và hơn 10 mật khẩu mỗi giây.

+ Một người dùng truy cập hơn 30 hồ sơ khách hàng trong một lần và truyền chúng cho địa chỉ người khác trong vòng 10 giây sau khi truy cập chúng.

* Danh sách kiểm tra để cải thiện các khía cạnh bảo mật của các yêu cầu.
* Xem xét các địa điểm trong ứng dụng được đề xuất nơi có thể xâm nhập. Là yêu cầu an ninh cụ thể được nêu cho những nơi đó?
* Tính bảo mật của dữ liệu, nếu có, được yêu cầu cụ thể?
* Bảo mật danh tính người dùng đã được chỉ định chưa?
* Bảo mật của mật khẩu đã được gọi một cách rõ ràng chưa?
* Quyền sở hữu của fles hoặc truy cập đã được chỉ định?
* Mã hóa đã được gọi khi nào thích hợp?
* Các khai thác bảo mật cụ thể, đã biết ("hack") đã được chỉ định chống lại? Một ví dụ là "SQL Ịnjection sẽ bị ngăn chặn. "(SQL Ịnection là phương tiện truy cập cơ sở dữ liệu trái phép.)

# **13.9 Tự hoàn thành các yêu cầu**

+ Thông thường, một yêu cầu phụ thuộc vào các yêu cầu khác. Một tập hợp các yêu cầu là tự hoàn thành nếu nó chứa mọi phần mà sự bao gồm là bắt buộc bởi các phần đã có.

+ Ví dụ: giả sử rằng SRS(đặc tả yêu cầu) cho một ứng dụng lịch có chứa các yêu cầu sau:

Ứng dụng sẽ giữ lại tất cả thông tin được nhập bởi người dùng cho mỗi cuộc hẹn.

* Ví dụ về sự không hoàn thành trong yêu cầu.

1. Ứng dụng sẽ hiển thị 1 đĩa DVD trong kho khi tiêu đề được nhập tại lời nhắc, trường hợp ngược lại nó sẽ hiển thị hết hang.
2. Ứng dụng sẽ hiển thị tất cả các DVD của cửa hàng bởi bất kỳ giám đốc nào có họ được nhập tại lời nhắc. Chúng sẽ được hiển thị từng cái một. Tiến qua DVD sẽ được điều khiển bằng phím mũi tên phía trước.

\* Chưa hoàn thành: thiếu thông số kỹ thuật về cách hiển thị video.

***Một số liệu tự hoàn thành.***

* Khi có yêu cầu, tất cả những yêu cầu đó phải cần thiết bởi sự hiện diện của nó.

Một số liệu (0 = tốt; 1 nghèo nàn; không có giới hạn trên lý thuyết)

- Số lượng thiếu các yêu cầu liên quan cần thiết/ số lượng yêu cầu chi tiết hiện tại.

- Sự hiện diện của yêu cầu này đòi hỏi một yêu cầu mô tả phương tiện để nhập thông tin cuộc hẹn. Nó cũng đòi hỏi một yêu cầu giải thích phương tiện để hiển thị thông tin này. Đây là ý nghĩa của "tự hoàn thành.".

* Danh sách kiểm tra cải thiện tự hoàn thành các yêu cầu. (tự đặt ra câu hỏi)

+ Đối với mỗi yêu cầu được nêu, có phải tất cả các yêu cầu hiện tại mà nó đề cập đến?

+ Đối với mỗi yêu cầu đã nêu, có phải tất cả các yêu cầu hiện tại phụ thuộc vào?

- Để đo lường mức độ tự hoàn thiện, chúng tôi xem xét từng yêu cầu chi tiết và lưu ý mọi yêu cầu liên quan yêu cầu còn thiếu.

- Số lượng yêu cầu còn thiếu được xác định bằng cách lấy mẫu.

# **13.10 Khả năng iểm tra yêu cầu.**

Mỗi yêu cầu chi tiết phải được kiểm tra; điều đó là, nó phải có khả năng xác nhận chắc chắn rằng yêu cầu là hoạt động trong ứng dụng đã hoàn thành bằng cách thử nghiệm cho nó.

* Ví dụ về khả năng kiểm tra.
* Hệ thống sẽ hiển thị sự khác nhau về tiền lương giữa khách hàng và trung bình trên toàn thế giới cho cùng giao dịch.

+ Không thể được kiểm tra vì mức trung bình được đề cập không thể được xác định (ngay cả khi nó tồn tại).

+ Hệ thống sẽ hiển thị sự khác biệt về tiền lương giữa khách hàng và trung bình trên toàn thế giới cho cùng một giao dịch như được công bố bởi Liên Hợp Quốc trên trang web của mình www.tbd tại thời điểm hiển thị.

* Ví dụ về một câu hỏi về sự hài lòng của người dùng
* A pleasure to use: một niềm vui để sử dụng.

1. Ngoại hình tổng thể.
2. Bố cục các vùng văn bản
3. Bố trí các nút.
4. Dễ đọc.
5. Dễ nhập dữ liệu.
6. Mức độ ngăn chặn nhập dữ liệu sai.

- Các yêu cầu không thể kiểm tra được có giá trị không đáng kể. Không thể đánh giá liệu một yêu cầu đã đạt được. Đây là một tài sản tất cả hoặc không có gì. Có rất ít giá trị trong "mức độ khả thi" của một yêu cầu. Khả năng kiểm tra đôi khi có thể được sử dụng để xác định một yêu cầu GUI chi tiết.

\* Danh sách cải thiện kiểm tra khả năng yêu cầu.

* Có yêu cầu đủ rõ ràng để đưa ra các mẫu dữ liệu đầu vào / đầu ra thực hiện nó? Chỉ định các bài kiểm tra.
* Có một tập hợp các bài kiểm tra mà nếu được thông qua sẽ cung cấp sự xác nhận rõ ràng rằng yêu cầu sẽ được thỏa mãn.
* Nếu một người thích hợp trong cộng đồng khách hàng đề xuất một bộ thử nghiệm cho yêu cầu, liệu người khác có thể đồng ý rằng nó thử nghiệm yêu cầu đó không?
  1. **Khả năng truy xuất nguồn gốc của yêu cầu**
* Một yêu cầu được gọi là có thể truy xuất nguồn gốc được nếu nó chỉ ra một cách rõ ràng và đẩy đủ cách thức để thiết kế, thực thi và kiểm thử.

0 = không thể truy xuất nguồn gốc

1 = chưa xác định

1. = có thể truy xuất một các rõ ràng để xác định class

* Dưới đây là 1 checklist để cải thiện khả năng truy xuất nguồn gốc của yêu cầu:

+ Với mỗi yêu cầu chi tiết, có thể xác định rõ ràng một phần mã nguồn thực thi yêu cầu đó hay không? (trong điều kiện lí tưởng: mỗi một method đơn lẻ thực thi một yêu cầu chi tiết đơn lẻ)

+ Có thể chắc chắn rằng một phần đơn lẻ thiết kế có thể chứa và thực thi nó? (Một yêu cầu chi tiết mà cần nhiều module thì không dễ để truy xuất nguồn gốc)

* 1. **Số liệu phân tích yêu cầu**
* Như chúng ta đã biết, một yêu cầu bảo đảm các thuộc tính như: khả năng tiếp cận, tính toàn diện, dễ hiểu, không mơ hồ, tính nhất quán, mức độ ưu tiên, bảo mật, khả năng tự hoàn thiện, dễ kiểm thử và truy xuất nguồn gốc.
* Danh sách về các số liệu đảm bảo chất lượng sau đây bao gồm các số liệu phân tích yêu cầu theo tiêu chuẩn IEEE 982.2-1988:

+ Tỉ lệ phần trăm các yêu cầu cụ thể rõ ràng (số liệu IEEE 6)

+ Mức độ hoàn thiện (số liệu IEEE 23 và 35)

+ Tỉ lệ phần trăm các yêu cầu chi tiết được phân loại sai

+ Khả năng truy xuất nguồn gốc (số liệu IEEE 7)

+ Mức độ nguyên tử (không thể chia thành các phần nhỏ hơn)

+ Phù hợp với các yêu cầu còn lại (số liệu IEEE 12 và 23)

+ Các biện pháp về tính hiệu quả của việc kiểm tra yêu cầu

+ Tỉ lệ phần trăm các yêu cầu bị thiếu được tìm thấy trong mỗi giờ kiểm tra

+ Các biện pháp về tính hiệu quả của quá trình phân tích yêu cầu

+ Chi phí cho mỗi yêu cầu chi tiết

+ Trên cơ sở tổng thể (tổng thời gian sử dung/số yêu cầu chi tiết)

+ Trên cơ sở phát sinh (chi phí để thêm 1 đơn vị)

+ Tỉ lệ các yêu cầu bị sửa đổi, bị đổi hoặc được thêm vào

+ Đo lường mức độ hoàn chỉnh của yêu cầu. Điều này có thể được ước tính từ tỉ lệ, sau khi tập hợp các yêu cầu chi tiết kết thúc, tại đó các yêu cầu cụ thể bị sửa đổi hoặc thêm vào

* 1. **Kiểm tra yêu cầu chi tiết**
* Yêu cầu chi tiết là tài liệu quy trình phần mềm đầu tiên, nó có thể được kiểm tra, đối chiều với tài liệu trước đó (yêu cầu cấp cao). Người kiểm tra cần chuẩn bị tốt cho việc kiểm thử bằng cách đọc kĩ các yêu cầu cấp cao và so sánh chúng với các yêu cầu chi tiết. Để đạt hiệu quả cao thì cần kiểm tra yêu cầu ở từng điểm thuộc tính và số liệu được nêu ở phần trước.
* Đây là phiên bản chưa kiểm tra của các yêu cầu chi tiết cho phân tích các trường hợp trong videogame Encounter. Trong đó chúng ta sẽ thấy được ví dụ về kiểm tra, hãy cùng xem qua bảng dưới đây.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Mô tả** | **Truy xuất ngược** | **Hoàn thiện** | **Tương tích** | **Khả thi** | **Minh bạch** | **Rõ ràng** | **Ngắn gọn** | **Dễ hiệu chỉnh** | **Dễ kiểm thử** | **Note 14** |
| 1001 | Y/c khu vực 1 | Note 2 | Note 1 | Có | Có | No 1 | Có | No 1 | No 2 | No 1, 2 | Có |
| 1002 | Y/c khu vực 2 | Có | Có | Có | Có | No 3 | Có | No 3 | Note 3 | Có | Có |
| 1003 | Y/c khu vực 6 | Có | Note 5 | Note 5 | Có | No 3 | No 3 | No 5 | Có | Có | Có |
| 1004 | Y/c khu vực 3 | Có | Có | Có | Có | Có | Có | No 6 | Note 3 | Có | Có |
| 1005 | Y/c khu vực 4 | Có | Note 7 | Có | Có | Có | Có | Có | Có | Có | Có |
| 1006 | Y/c chiến đấu 1 | Note 2 | Có | Có | Có | Có | Có | Có | Note 3 | Có | Có |
| 1007 | Y/c nhân vật 1 | Có | Note 1 | Có | Có | No 1 | Có | No 1 | No 2 | No 1, 2 | Có |
| 1008 | Y/c nhân vật 2 | Có | Note 11 | Có | Có | Có | Note 8 | Có | No 6 | Có | Note 9 |
| 1009 | Y/c nhân vật 3 | Có | Có | Có | Có | Có | Có | Có | Note 3 | Có | Có |
| 1010 | Y/c nhân vật 4 | Note 11 | Có | Có | Có | Có | Note 12 | Note 12 | No 7 | Có | Có |
| 1011 | Y/c trò chơi 1 | Có | Có | Có | Có | Có | Có | Có | Note 13 | Có | Có |
| 1012 | Y/c nhân vật bên ngoài 1 | Note 11 | Có | Có | Có | Có | Có | No 8, 9 | Có | Có | Có |
| 1013 | Y/c người chơi 1 | Có | Có | Có | Có | Có | No 10 | Có | Note 15 | Có | Có |
| 1014 | Y/c người chơi 2 | Có | Có | Có | Có | No 12 | No 12 | No 12 | Note 3 | No 12 | Có |
| 1015 | Y/c người chơi 3 | Có | Có | Có | Có | Note 10 | No 10 | Có | Có | Có | Có |

* Các “Note”:

1. Tất cả ký tự bàn phím đều được chấp nhận?
2. Kiểm tra tính hợp lệ với khách hàng
3. Không rõ điều này có thể sửa đổi như thế nào
4. Thật khó để nói là “đã hoàn thành” vì nó không rõ ràng. Xem ghi chú được tham chiếu trong cột “Xóa” để giải quyết vấn đề
5. Giả định rằng khách hang có một số phân vân đề việc “courtyard” chính xác sẽ trông như thế nào
6. Có phòng thay đồ nào dẫn tới khu vực khác không?
7. Điều này có phần được viết một cách vụng về: có thể dẫn đến sự hiểu lầm
8. Thông thường, sẽ có 1 yêu cầu đơn lẻ phù hợp với mỗi thuộc tính. Điều này có vẻ không hợp lí, vì các thuộc tính sẽ có quan hệ với nhau.
9. Tạo ra bất cứ lúc nào? Theo yêu cầu? Hiển thị mọi lúc?
10. Những chi tiết này không được đề cập trong các yêu cầu cấp cao: kiểm tra với khách hàng.
11. Làm rõ "50% cơ sở", nếu có thể.
12. Đối với các phiên bản Internet, có thể cần có nhiều hơn một phiên bản cho đối tượng EncountGame. Chúng ta sẽ không loại trừ khả năng này trong tương lai
13. Không rõ rằng điều này sẽ được sửa đổi theo hướng nào
14. Yêu cầu được viết theo cách nào đó tiện cho việc theo dõi thông qua mã nguồn

* Các “No”:

1. Một nhân vật hay khu vực trò chơi có thể có tên không bao gồm kí tự nào không?
2. Con số 15 là bắt buộc
3. Chỉ có một?
4. Nếu người chơi điều khiểu một vài nhân vật thì tất cả khu vực của họ sẽ được hiển thị hay chỉ hiển thị với nhân vật chính?
5. Làm kín toàn bộ màn hình?
6. Sẽ dễ dàng hơn để thêm thuộc tính mới hoặc loại bỏ chúng
7. Khi nào có Freddie? Khi nào anh ấy xuất hiện?
8. Trong tương lai, các nhân vật có thể biến hóa
9. Làm rõ những thứ có sẵn
10. Đặc trung có thể nhận giá trị âm?
11. Mơ hồ, bởi vì người chơi không thể kiểm soát mọi thứ xảy ra với nhân vật mọi lúc
12. Tinh chỉnh “kiểm soát hoàn toàn”

* Chúng ta sẽ sử dụng kỹ thuật tự động thêm các yêu cầu mà chưa được kiểm tra vào và ghi chú cho từng cái. Kết quả của việc kiểm tra sẽ được thể hiện trong những phân tích trường hợp Encounter được đính kèm.

+ Y/c khu vực 1 (tên khu vực): (chưa kiểm tra) tất cả các khu vực sẽ có 1 cái tên gồm tối đa 15 kí tự

+ Y/c khu vực 2 (hình ảnh khu vực): (chưa kiểm tra) mỗi khu vực sẽ có 1 hình ảnh dạng gif đại diện

+ Y/c khu vực 3 (cách thức hiển thị): (chưa kiểm tra) bất cứ khi nào 1 nhân vật bước vào khu vực thì sẽ hiển thị nhân vật đó ở khu vực tương ứng

+ Y/c khu vực 4 (đối tượng courtyard): (chưa kiểm tra) Sẽ có một khu vực tên là “courtyard”. Tên của nó thể hiện trong số liệu xx ở trang xx

+ Y/c khu vực 5 (đối tượng phòng thay đồ): (chưa kiểm tra) Sẽ có một khu vực tên là “Phòng thay đồ” với background trống, nằm liền kề khu vực “courtyard”

+ Y/c nhân vật 1 (tên nhân vật): (chưa kiểm tra) Mỗi nhân vật sẽ có 1 tên riêng biệt, gồm tối đa 15 kí tự

+ Y/c nhân vật 2 (thuộc tính của nhân vật): (chưa kiểm tra) Mỗi nhân vật sẽ có 1 loạt các thuộc tính ứng với một giá trị nhất định. Các giá trị này được khai báo bằng 100/n, trong đó n là số thuộc tính. Các thuộc tính gồm có tầm đánh, sức chịu đứng, trí thông minh, kiên nhẫn và sức mạnh.

+ Y/c nhân vật 3 (hình ảnh nhân vật): (chưa kiểm tra) Mỗi nhân vật sẽ có một hình ảnh đại diện to không quá 1/8 kích thước màn hình.

+ Y/c nhân vật 4 (chiến đấu với nhân vật bên ngoài): (chưa kiểm tra) Bất cứ khi nào 1 nhân vật bước vào khu vực có mặt 1 nhân vật khác và 1 trong số đó do người chơi điều khiển có thể chọn (hoặc bắt buộc, tùy trường hợp) chiến đầu với nhân vật còn lại.

**+** Y/c trò chơi 1 (đối tượng game): (chưa kiểm tra) Sẽ có 1 đối tượng game đơn lẻ

+ Y/c nhân vật bên ngoài 1 (đối tượng nhân vật bên ngoài Freddie): (chưa kiểm tra) Sẽ có một nhân vật bên ngoài tên là Freddie, tất cả các thuộc tính của Freddie đều có giá trị bằng nhau và giá trị này được thể hiện trong Số liệu 4.57

+Y/c người chơi 1 (cấu hình): (chưa kiểm tra) Bất cứ khi nào tất cả các nhân vật bên ngoài vắng mặt trong 1 khu vực, người chơi có thể cài đặt giá trị các thuộc tính của mình, sử dụng Cửa-sổ-đặc-điểm-người-chơi, miễn là tổng các giá trị đó không đổi.

+ Y/c người chơi 2 (nhân vật chính): (chưa kiểm tra) Người chơi sẽ điều khiển hoàn toàn một nhân vật cụ thể, gọi là nhân vật chính.

+ Y/c người chơi 3 (chỉ số sự sống): (chưa kiểm tra) Tổng giá trị các thuộc tính của nhân vật gọi là chỉ số sự sống

* 1. **Tổng hợp**
* Chất lượng của 1 tài liệu yêu cầu được phản ánh ở chỗ nó thể hiện những điều khách hàng muốn và cần tốt như thế nào. Để đảm bảo chất lượng của yêu cầu, chúng ta nên tập trung vào các thuộc tính mà chúng nên sở hữu. Chúng bao gồm những điều sau đây:

+ Có thể xem xét được

+ Toàn diện

+ Dễ hiểu

+ Không mơ hồ

+ Tương thích

+ Ưu tiên

+ Bảo mật

+ Tự hoàn thiện

+ Dễ kiểm thử

+ Có thể truy xuất nguồn gốc

* Yêu cầu được kiểm tra sau khi được viết ra. Chúng là tài liệu quy trình phần mềm đầu tiên, có thể được kiểm tra, đối chiếu với tài liệu trước đo (yêu cầu của khách hàng). Để hiệu quả thì cần kiểm tra yêu cầu dựa trên mỗi thuộc tính ở trên.