* Các cách tổ chức các yêu cầu chi tiết?

**12**

**Phân tích các yêu cầu chi tiết**

* Làm thế nào để bạn thể hiện chi tiết giao diện người dùng?
* Làm thế nào để bạn thể hiện chi tiết yêu cầu bảo mật?
* Những loại điều kiện lỗi (error conditions) nào có thể được chỉ rõ?
* Truy xuất nguồn gốc (Traceability) là gì và tại sao nó quan trọng?
* Các cách để có yêu cầu ưu tiên?
* Tại sao các yêu cầu liên kết với các bài kiểm tra?
* Làm thế nào để phương pháp agile liên quan đến chi tiết các yêu cầu?
* Làm thế nào để bạn sử dụng công cụ cho các yêu cầu phân tích?

Sau khi các yêu cầu cấp cao được quy định, thường là một sự lặp lại, bước tiếp theo trong phân tích các yêu cầu là xác định các yêu cầu chi tiết. Mục đích của yêu cầu chi tiết là để cung cấp cho người đọc tất cả những gì cần thiết của ứng dụng. Với một ngoại lệ, có nơi nào khác để nhóm đưa ra chính xác những điều này bao gồm. Ví dụ, nếu chúng ta không nêu rõ các tiêu đề của video phải là phông chữ 16 trên màn hình, chúng tôi sẽ không cho xuất hiện trong các sản phẩm.

Những “ngoại lệ” được đề cập ở trên đề cập đến khả năng của trạng thái yêu cầu chi tiết như một bình luận trong các mã nguồn hoặc các kỳ vọng mã nguồn và kiểm tra đơn vị của nó để xác định hiệu quả các yêu cầu. Điều này có xu hướng tiếp cận các phương pháp của dự án agile mà chúng ta thảo luận trước đó. Dự án Agile không loại trừ các loại tài liệu mà chúng ta thảo luận trong chương trình này, nhưng nó có giá trị mã làm việc trước các tài liệu riêng biệt như vậy.

Chương này tập trung vào các yêu cầu chi tiết bằng văn bản. Tuy nhiên, cho dù ai viết ra các yêu cầu chi tiết hay không, không có sự lựa chọn nhưng cuối cùng họ nghĩ rằng vẫn được thông qua. Trong thực tế, chúng tạo thành một đồng tiền chung của các ứng dụng. Yêu cầu chi tiết thường được chia thành các phần, bao gồm các yêu cầu chức năng, yêu cầu không chức năng và các chi tiết GUI.

# 12.1 Ý nghĩa của các yêu cầu chi tiết

Yêu cầu chi tiết cung cấp cho kỹ sư phần mềm cơ bản về thiết kế và thực hiện. Chúng đôi khi được gọi là “yêu cầu cụ thể” , “đặc tả chức năng” hay “yêu cầu phát triển”. Yêu cầu chi tiết bao gồm một danh sách đầy đủ các đặc tính cụ thể và chức năng mà các ứng dụng phải có, thể hiện chi tiết đầy đủ. Mỗi một yêu cầu được dán nhãn và theo dõi thông qua việc thực hiện. Yêu cầu chi tiết phù hợp và được xây dựng trên các yêu cầu cấp cao. Chúng được dành để đọc chủ yếu bởi các nhà phát triển tuy nhiên, khách hàng quan tâm đến họ là tốt, và sẽ có thể hiểu và nhận xét cùng với vài trường hợp ngoại lệ. Nhớ lại rằng các đối tượng chính cho các yêu cầu cấp cao bao gồm các khách hàng.

Như các trường hợp nghiên cứu trong cuốn sách này chứng minh, khi nói đến công nghệ phần mềm, “the devil is in the details”. Ví dụ, năm 1999, NASA mất một vệ tinh thời tiết giá trị của một báo cáo 125 triệu $, vì dữ liệu kiểm soát họ đã giả định không có.

Mô tả này ngụ ý rằng các yêu cầu nên lên sự cần thiết cho các đơn vị số liệu nhưng phần mềm này đã không thực hiện các yêu cầu chính xác. Một thực tế thú vị là khuyết điểm này đã được xác định trong phạm vi chỉ ngay sau khi xảy ra thảm họa. Điều này có nghĩa rằng nó có thể không khó để xác định vị trí một khiếm khuyết một khi chúng ta biết nó hiện diện. Vấn đề thường là sự thiết hiểu biết của chúng ta về sự hiện diện của nó. Các yêu cầu chi tiết cho dòng đầu tiên của việc phòng chống sự hư hại hoặc bỏ sót chi tiết. Thay vì hoạt động thiếu suy nghĩ cho rằng nó có thể là lần đầu tiên xuất hiện, nhận tất cả các yêu cầu chi tiết đầy đủ liên quan đến những nhiệm vụ khó khăn của tổ chức con người và quá trình suy nghĩ của họ. Để hiểu được thách thức này, hãy tưởng tượng nhiệm vụ tổ chức một tài liệu yêu cầu 20 tập để có thể như một kỹ sư NASA, ví dụ, có thể biết chính xác nơi để thêm hoặc tìm kiếm một yêu cầu cụ thể. Lưu trữ và duy trì các yêu cầu này trong một cơ sở dữ liệu tìm kiếm có thể giúp rất nhiều nhưng công việc vẫn còn khó khăn.

# 12.2 Tổ chức các yêu cầu chi tiết

Yêu cầu thay đổi liên tục, do đó việc viết yêu cầu phải được tổ chức tốt và dễ dàng để cập nhật. Để đánh giá đúng giá trị các yêu cầu chi tiết tổ chức một cách cẩn thận, ta hãy xem xét văn bản yêu cầu chi tiết cho trò chơi Encounter. Lưu ý rằng các yêu cầu này vẫn còn nguyên và không được kiểm tra.

Mỗi nhân vật trong game có một tên.

Mỗi nhân vật game có cùng một tập các thuộc tính, đều có một giá trị kiểu dấu chấm động.

Mất ít hơn một giây để tính toán các kết quả của một sự trận chiến.

Mỗi khu vực có một tập hợp cụ thể gọi là "những thuộc tính phẩm chất cần thiết." Ví dụ, khu vực chiến đấu đòi hỏi sức mạnh và sức chịu đựng; phòng sinh hoạt đòi hỏi sự nhạy cảm và trí tuệ.

Khi hai nhân vật trong Encounter đang ở cùng khu vực cùng một lúc họ có thể thể chọn hoặc buộc phải đánh nhau.

Mỗi nhân vật trò chơi có các số điểm sống.

Tổng các giá trị của các thuộc tính phẩm chất của một nhân vật có liên quan đến các khu vực sẽ được gọi như giá trị khu vực của nhân vật. Trong một trận chiến, hệ thống so sánh giá trị khu vực của nhân vật và tính toán kết quả của sự tham gia nhân vật đó.

Tên của bất kỳ nhân vật nào trong Encounter sẽ không có nhiều hơn 15 chữ cái.

Khi nó phát triển, một danh sách chưa được tổ chức như ở trên nhanh chóng trở nên không thể quản lý.

* Vấn đề kích thước của nó làm cho nó khó hiểu ngay cả trên một đơn vị, đó là chưa tính đến nó phát triển thành hàng trăm, hàng ngàn.

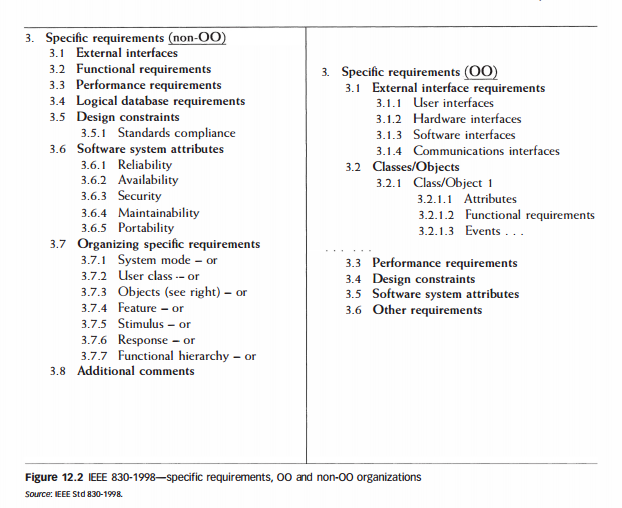
• Các yêu cầu của các loại hỗn hợp: yêu cầu thực hiện phải được xử lý khác nhau.

• Một số yêu cầu tự nhiên đi liền với những yêu cầu liên quan.

• Rất khó để xác định vị trí một yêu cầu cụ thể.

Yêu cầu chi tiết chức năng có thể được tổ chức theo nhiều cách phân loại, bao gồm cả theo tính năng, trường hợp sử dụng, CUI, giai đoạn, và lớp. Chúng tôi mô tả mỗi phương pháp chi tiết hơn trong các phần tiếp theo. Những công cụ cho yêu cầu quản lý giúp đỡ rất nhiều. Tuy nhiên, quyết định như thế nào để tổ chức yêu cầu chi tiết ở nơi đầu tiên là quan trọng bởi vì các nhóm giới thiệu cho họ liên tục nếu tài liệu được thực hiện tốt.

Các tiêu chuẩn IEEE 830- 1998 cung cấp tài liệu mẫu cho một số cách để phân loại các yêu cầu chi tiết. Hình 12. 2 cho thấy các đối tượng của IEEE thông thường và 830- 1 998 tiêu chuẩn. Việc phân loại đối tượng sử dụng các lớp / đối tượng như một phương pháp tổ chức yêu cầu chức năng. SRS thường được thiết kế riêng cho nhu cầu của công ty hoặc nhóm bằng cách thêm hoặc sửa đổi phần cho phù hợp. Ví dụ, tổ chức 00 thiếu một phần tương đương với 3. 4 trong không-OO tổ chức "các yêu cầu cơ sở dữ liệu Iogical." Nghiên cứu trường hợp Encounter sử dụng một hình thức sửa đổi của IEEE 00 phong cách và bao gồm một phần cho các trường hợp sử dụng. Hình 1 2. 3 bản đồ các yêu cầu chi tiết các phần của mục 3 để thể loại yêu cầu nó mô tả.

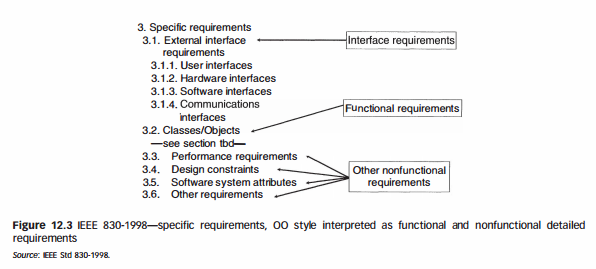


Nó có thể được khuyến khích để tổ chức các yêu cầu chi tiết bằng một sự kết hợp của các phân loại. Ví dụ, tổ chức dựa trên tính năng có thể được sử dụng trong các cấu hình, thực hiện, và sao lưu trạng thái của một ứng dụng kế toán. Các yêu cầu cho một hệ thống nhà máy tự động hóa có thể được tổ chức ở cấp cao nhất của chức năng (một phần sản xuất và lắp ráp); sau đó họ có thể được tổ chức bởi trong mỗi lớp.

Điều này có nghĩa là phương thức tổ chức yêu cầu chi tiết đôi khi có thể xảy ra liên quan đến kiến trúc hoặc triển khai các ứng dụng. Ví dụ, nếu thiết kế theo hướng đối tượng, chi tiết yêu cầu bởi trường hợp sử dụng hoặc theo lớp tổ chức nên được xem xét bởi vì họ tạo điều kiện truy xuất nguồn gốc. Nếu ứng dụng vay chính nó để một phân tích chức năng rõ ràng, sau đó tổ chức các yêu cầu của hệ thống phân cấp tính năng có thể thích hợp.

## 12.2.1 Yêu cầu chi tiết tổ chức theo tính năng

Cách thức cổ điển nhất của tổ chức yêu cầu chi tiết là bởi tính năng. Hình sau cung cấp một danh sách đơn giản các chức năng như sau đối với trò chơi Encounter. Nhiều tính năng được xác định bởi một cặp stimulusand phản ứng, đối với yêu cầu 127.



Sẽ có một nhóm vị trí siêu liên kết ở mọi lối ra vào mọi lĩnh vực.

126. Các nhân vật ngoài di chuyển từ vùng này sang vùng lân cận vào thời điểm ngẫu nhiên, với trung bình ba giây.

127. Khi người dùng nhấp chuột vào nút "Đặt chất lượng", cửa sổ trong hình xx xuất hiện trên màn hình.

Sắp xếp các yêu cầu bởi tính năng có lợi thế là đơn giản nhưng bất lợi là việc không tiếp cận; bởi vì nó cho phép nhảy từ một tính năng trong một phần của ứng dụng đến một tính năng trong một phần khác hoàn toàn. Ví dụ, nếu chúng ta muốn thay đổi cách thức mà các nhân vật ngoài di chuyển về, chúng ta sẽ phải xác định rằng các yêu cầu có liên quan là số 126. Làm thế nào chúng ta sẽ biết những yêu cầu khác bị ảnh hưởng? Công cụ tìm kiếm có thể giúp đỡ rất nhiều. Một nhược điểm nữa là việc thiếu truy xuất nguồn gốc. Một cách để áp đặt trật tự trên danh sách tính năng chức năng là để sắp xếp chúng trong một hệ thống phân cấp chức năng (tức là bằng phân hủy các ứng dụng vào một tập hợp các chức năng cao cấp, và sau đó các thành subfunctions, vv. ). Ví dụ, các yêu cầu cho một chương trình ngân sách tại nhà có thể được phân tích thành (1) kiểm tra chức năng, (2) chức năng tiết kiệm, và (3) các chức năng đầu tư. Các chức năng kiểm tra yêu cầu có thể được phân hủy thêm vào chức năng séc, hòa giải, và báo cáo v.v.

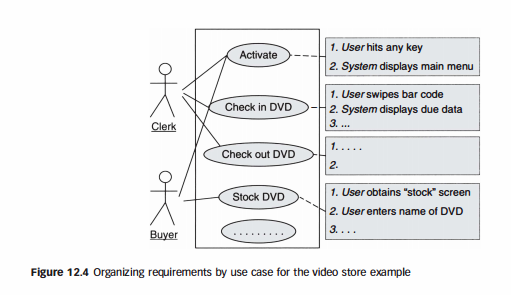
## 12.2.2 Yêu cầu chi tiết tổ chức theo trường hợp

Sử dụng trường hợp (giới thiệu trong chương 11) được nhận xét rằng xảy ra nhiều trong trình tự hoạt động. Ví dụ, một yêu cầu cá nhân như "một ứng dụng cửa hàng video sẽ cho phép nhập vào tiêu đề của video mới" thường diễn ra như là một phần của một chuỗi các giao dịch. Một sơ đồ trường hợp sử dụng cho thấy một bộ sưu tập các trường hợp sử dụng cho một cửa hàng ứng dụng video được minh họa trong hình 12 .4.

Các tác giả chủ trương cung cấp các trường hợp sử dụng cho các yêu cầu cao cấp. Một người có thể có tùy chọn sử dụng nguyên tắc tổ chức cùng này cho các yêu cầu chi tiết. Trong trường hợp này, chúng ta sẽ nhóm các chi tiết yêu cầu theo từng trường hợp sử dụng. Sau đây là một ví dụ về một nhóm các yêu cầu sẽ xuất hiện trong SRS.

3 .7. 1 Kiểm tra trong DVD

Đây là một phiên bản chi tiết của phần xx. (Lưu ý 1, xem "Ghi chú" giải thích trên trang tiếp theo)



Ứng dụng này sẽ cung cấp các khả năng sau để tương tác với các nhân viên cửa hàng. Những

yêu cầu của màn hình chính. (Lưu ý 2 )

3. 7. 1. 1 Bước 1: Kiểm tra trong nút nhấn (Lưu ý 3)

Các nhân viên bán hàng thì có thể nhấn nút Check In, sau đó, màn hình Check In sẽ xuất hiện (Mô tả trong phần xx trên).

3 .7. 1 .2 Bước 2: Điền in các trường (Lưu ý 4)

Các nhân viên bán hàng sẽ có thể điền vào các trường văn bản sau đây trong màn hình Check In

3 .7. 1 .2. 1 Video trường tên

Trường này sẽ cho phép 30 ký tự chữ và số, kể cả space.

Lưu ý 1: đoạn này tương ứng với phân đoạn Kiểm tra trong trường hợp sử dụng đĩa DVD được mô tả trong các cấp cao các yêu cầu.

Lưu ý 2: Phần này giới thiệu các trường hợp sử dụng và nêu điều kiện tiên quyết, nếu có. Một điều kiện tiên quyết là một thực tế giả định đúng tại thời điểm khởi trường hợp sử dụng.

Lưu ý 3: Điều này tương ứng với bước đầu tiên của các trường hợp sử dụng.

Lưu ý 4: Vì đây là những yêu cầu chi tiết, họ phải xác định các yêu cầu hoàn toàn. Các chi tiết sẽ không được quy định ở nơi khác.

Quá trình phát triển phần mềm thống nhất ủng hộ việc tổ chức tất cả các yêu cầu của các trường hợp sử dụng.

## 12.2.3 Yêu cầu chi tiết tổ chức theo giao diện người dùng

Ứng dụng hiển thị GUI là những phương tiện mà người dùng dùng để đánh giá, vì vậy chúng có thể được tự nhiên cung cấp các yêu cầu tổ chức theo cách này. Sử dụng cách này, các yêu cầu đối với các trò chơi Encounter sẽ như sau:

1. Khu vực GUI

Khu vực GUI sẽ có sự xuất hiện trong hình xx. Khi người dùng nhấp chuột vào nút "Đặt chất lượng ", cửa sổ trong hình xx xuất hiện trên màn hình. Khi người dùng nhấp vào ...

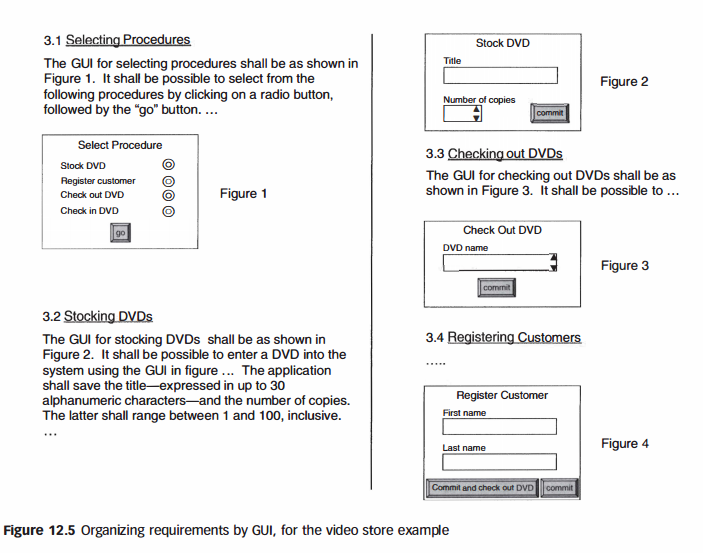
2. Dungeon GUI ...

3. Phòng khách GUI ...

4. Thiết lập chất lượng GUI ...

5. Xem Phẩm chất Window ...

Lợi thế của tổ chức yêu cầu bởi GUI là kết nối trực tiếp với việc sử dụng các ứng dụng. Một lợi thế khác là truy xuất nguồn gốc: chúng ta có một cơ hội tốt để theo dõi các yêu cầu liên quan đến một lớp GUI. Một bất lợi của phương tiện này của tổ chức là nó lỗi bao phủ tất cả các yêu cầu. Ví dụ Encounter, chúng ta cần phải mô tả các yêu cầu cho sự tương tác của nhân vật của người chơi và các nhân vật ngoài. Như một ví dụ, hình 12.5 minh họa một tổ chức của các yêu cầu lưu trữ video bằng GUI. Đặc điểm kỹ thuật của GUI được thảo luận thêm tại mục 12.3.



## 12.2.4 Yêu cầu chi tiết theo giai đoạn:

Cách này là thu thập yêu cầu chi tiết áp dụng cho mỗi giai đoạn. Ví dụ, các yêu cầu cho một ứng dụng điều khiển một quá trình hóa học có thể được phân loại theo quá trình của chính nó (bắt đầu lên, phản ứng, làm mát, vv). Trong mỗi loại, các sự kiện ảnh hưởng đến các ứng dụng trong trạng thái đã được liệt kê. Phân loại theo trạng thái có thể thích hợp khi yêu cầu cho mỗi trạng thái là khá khác biệt. Ví dụ, một hệ thống kế toán có thể được yêu cầu để xử lí hoàn toàn khác nhau tùy thuộc vào việc nó là đang trong giai đoạn cấu hình, thực hiện, hoặc sao lưu trạng thái.

## 12.2.5 Yêu cầu chi tiết theo lớp:

Cách này thực hiện theo cách hướng đối tượng để tổ chức yêu cầu, một phân loại được trình diễn lần đầu tương đương với việc lựa chọn các lớp; sau đó các yêu cầu cá nhân được đặt vào các lớp kết quả. Các lớp sử dụng để phân loại các yêu cầu được biết đến như là các lớp miền. Các lớp miền đại diện cho đối tượng hiện thực trên thế giới hoặc các khái niệm trong ứng dụng. Ví dụ, một ứng dụng ngân hàng sẽ có các lớp miền như khách hàng, kiểm tra tài khoản, và tiết kiệm. Bước đầu tiên phổ biến trong việc xác định các lớp miền là để thu thập các định nghĩa hoặc tương đương được sử dụng trong các yêu cầu cao cấp. Sau đó chúng tôi làm cho mỗi chức năng chi tiết yêu cầu tương ứng với một chức năng trong ngôn ngữ đích.

Một bất lợi của tổ chức yêu cầu theo lớp là rủi ro mà sau này chúng ta thay đổi các lớp, do đó phá vỡ sự tương ứng giữa các yêu cầu và thiết kế. Điều này được thảo luận bởi Jordan, Smilan, và Wilkinson trong [2]. Trong thực tế, một số nhà phát triển sử dụng theo lớp để tổ chức các yêu cầu không nghiêm trọng mục đích để sử dụng các lớp cho các thiết kế. Trong trường hợp này, truy xuất nguồn gốc bị tổn thất, nhưng có ít áp lực xác định các lớp kéo dài từ rất sớm trong quá trình này. Một bất lợi của việc phân loại này là nó đòi hỏi chúng ta chọn lớp rất sớm trong chu kỳ phát triển, và nhiều người cho rằng chúng tôi đang thực hiện một cách hiệu quả thiết kế khi làm như vậy. Hãy xem xét các trường hợp nghiên cứu Encounter như là một ví dụ. Chọn lớp như PlayerCharacter và các yêu cầu thời gian là vô hại khi sử dụng các các lớp. Mặt khác, nó có thể được lập luận hợp lý rằng khi các đối tượng AreaConnection tham chiếu

đối tượng khu vực mà chúng kết nối là một quyết định thiết kế.

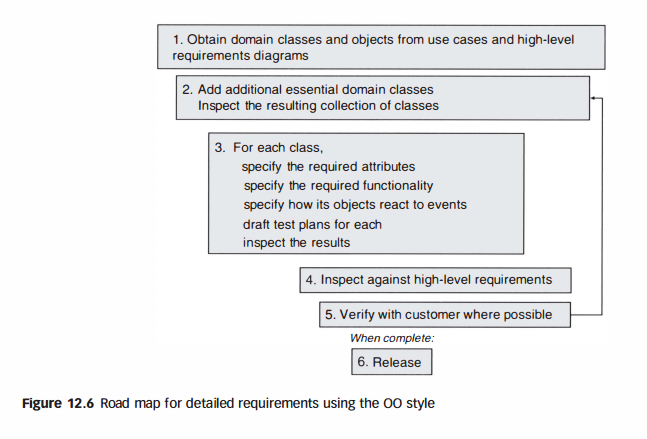
Những lợi thế lớn để tổ chức yêu cầu bởi các lớp sẽ được sử dụng trong thiết kế là nó thúc đẩy sự tương ứng chặt chẽ giữa yêu cầu, thiết kế và triển khai thực hiện. Đây là một lợi ích quan trọng cho cách sử dụng mô hình 00 trong mọi trường hợp.

Một danh sách điển hình cho việc thu thập yêu cầu chi tiết chức năng bằng cách sử dụng mô hình 00 như sau:

1. Liệt kê các khái niệm được đề cập trong trường hợp sử dụng và các yêu cầu cao cấp khác.

2. Bộ sưu tập kết quả của các lớp thường là không đầy đủ.

3. Đối với mỗi lớp thu được, viết xuống tất cả các chức năng cần thiết của ứng dụng chủ yếu liên quan đến lớp đó. Điều này được thực hiện dưới hình thức các thuộc tính và chức năng. Ví dụ, "mỗi khách hàng sẽ có một tên" (một thuộc tính được liệt kê theo khoản khách hàng)



và "các ứng dụng sẽ có thể tính tổng tài sản của từng khách hàng" (một chức năng được liệt kê dưới Khách hàng). Trong (yêu cầu) tài liệu mà bạn đang viết, tránh sử dụng thuật ngữ "c1ass." Sử dụng bình thường, không kỹ thuật bằng tiếng Anh.

4. Kiểm tra các yêu cầu chi tiết như quá trình phát triển.

5. Một cách lý tưởng, kế hoạch kiểm tra đối với từng yêu cầu chi tiết cần được thiết kế cùng một lúc, như được giải thích dưới đây.

6. Thanh tra các yêu cầu chi tiết đối với các yêu cầu cao cấp.

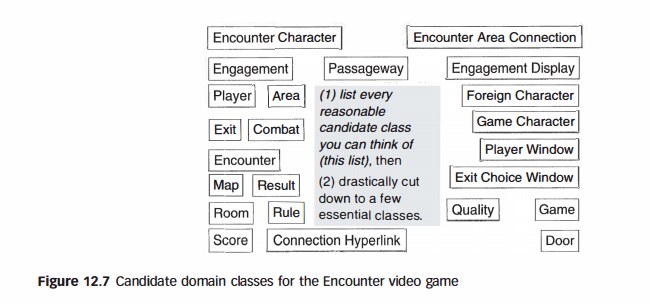
7. Xác nhận các yêu cầu chi tiết với khách hàng.

8. Quay lại Bước I để yêu cầu nhiều hơn.

Nhớ lại rằng các đối tượng chính cho các yêu cầu chi tiết bao gồm các nhà phát triển. Tuy nhiên, khách hàng là cực kỳ quan tâm đến các chi tiết, quá. Hình 12. 6 tóm tắt các bước sau.

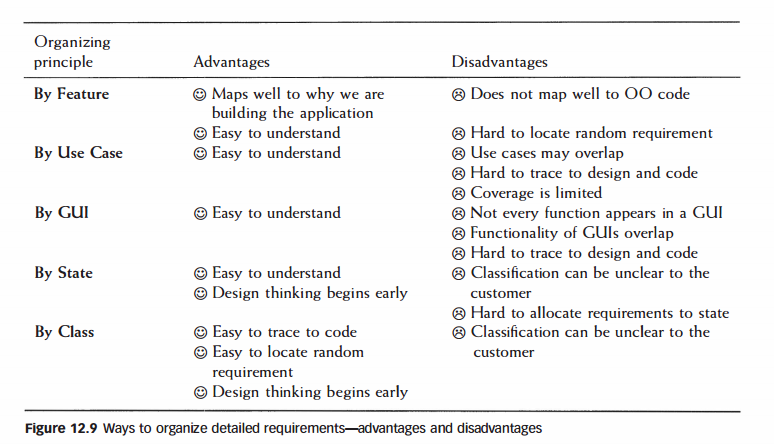
Đó là một lỗi phổ biến khi phân loại các yêu cầu của lớp học để sử dụng ngôn ngữ thiết kế thay vì

đồng bằng tiếng Anh. Ví dụ, các ngôn ngữ sau đây được chấp nhận.



## 12.2.6 Phương pháp Phân loại: Ưu điểm và nhược điểm

Hình 12.09 tóm tắt những cách khác nhau để tổ chức yêu cầu chi tiết chức năng mà chúng ta đã thảo luận trong các phần trước, cùng với lợi thế và bất lợi tương đối của chúng.



# 12.3 Giao diện người dùng: Các yêu cầu chi tiết

Nhớ lại các bước dưới đây trong các user interfaces cụ thể

**Step 1**: hiểu biết về user

**Step 2**: hiểu chức năng nghiệp vụ trong quá trình hỏi user

**Step 3:** ứng dụng các nguyên lý về thiết kế màn hình tốt

**Step 4:** lựa chọn các loại cửa sổ (windows) phù hợp

**Step 5:** phát triển menu cho hệ thống

**Step 6:** chọn các controls thiết bị phù hợp

**Step 7:** chọn ra các controls màn hình

**Step 8:** tổ chức các lay out windows

**Step 9:** chọn tông màu phù hợp

**Step 10:** tạo các icons rõ nghĩa

**Step 11:** cung cấp các message hiệu quả, mang tính feedback và thể hiện được sự hướng dẫn tới user

Đảm bảo tính nhất quán giữa các màn hình của các ứng dụng được thiết kế, giữa các màn hình với nhau

+ các chuẩn phong cách, các thủ tục, có khả năng nhìn dễ hiểu và cảm nhận được, vị trí phù hợp

Dự đoán user thường sẽ start tại đâu

+ thường là phía bên trái

Tạo ra các điều hướng càng đơn giản càng tốt

+ căng chỉnh các phần tử / control

+ nhóm các phần tử / control có liên quan với nhau

+ xem xét các viền bao quanh các phần tử / control

Ứng dụng một kế thừa để làm nổi bật thứ tự quan trọng

Ứng dụng các nguyên lý làm trực quan:

+ cân bằng, đối xứng, chuyên nghiệp, khả năng dự đoán

+ đơn giản, hữu dụng, cân đối, tiết kiệm không gian

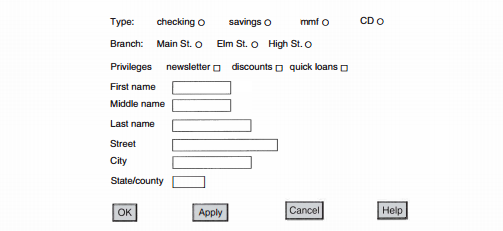
Cung cấp các tiêu đề

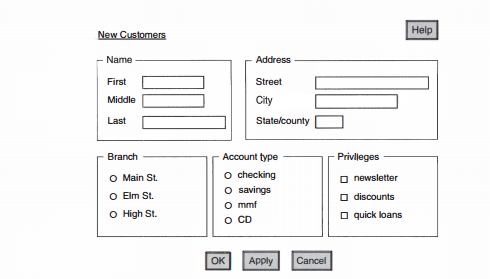
Các bước 1 và 2 được mô tả ở chương 11, vì nó được ứng dụng chủ yếu cho các yêu cầu cấp cao. Chúng ta mô tả các bước còn lại cho các yêu cầu về giao diện chi tiết, cụ thể

**Bước 3:** ứng dụng các nguyên lý của thiết kế màn hình tốt

Hình dưới đây liệt kê các thành phần chính để thiết kế một màn hình tốt. Trong hình bao gồm các hệ số thường dùng để tạo ra sự hài lòng về giao diện. Mặc dù các thành phần đó chỉ dùng để giới thiệu các vấn đề về hiệu quả trực quan, tuy nhiên các thành phần đó có thể sử dụng bởi nền công nghệ phần mềm hạng trung

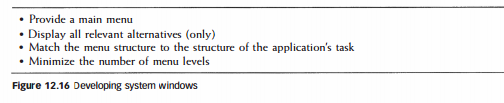
Ví dụ, chúng ta sử dụng một vài nguyên lý cho một màn hình được sử dụng để nhập thông tin về khách hàng và tài khoản của họ. Một thiết kế tạm được khởi tạo tại tầng GUI được show dưới đây. Để cải thiện interface, chúng ta bắt đầu tại top-left, đặt thành phần / control có quan trọng nhất, và nhóm các phần tử giống nhau.





**Step 5:** phát triển các menu hệ thống

Một vài quy tắc để tạo ra các main menu, được cung cấp bởi Calitz [3], được show dưới đây



Các user yêu cầu khả năng anchor (neo) ổn định, toàn vẹn cho ứng dụng, do đó cần các main menu cố định, ko thay đổi. Số lượng các items trên menu thường nên nằm giữa 5 và 9, vì hầu hết chúng ta cảm thấy thoải mái với lựa chọn đó. Chẳng hạn, bộ xử lý word với cuốn sách này chỉ có 9 main menu item: File, Eedit, View, Insert, Format, Tools, Table, Window, Help. Số lượng item có thể nhiều hơn, vì có nhiều không gian trống. Tuy nhiên, chúng ta có lẻ tìm kiếm liên tục danh sách các option mà chúng ta yêu cầu, và các kết quả ngoài mong đợi làm gia tăng sự lựa chọn.

Các items trong main menu được xác định bởi nghiệp vụ, trong tình huống này, ta cần xử lý về câu chữ. Ví dụ, các lệnh vè đồ họa sẽ được đặt trong menu thứ cấp

**Step 6:** lên danh sách các controls thiết bị

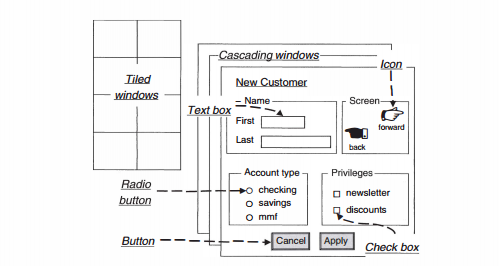
“Các controls thiết bị” là nghĩa vật lý bởi user communicate các mong muốn của họ tới ứng dụng, bao gòm joysticks, trackballls, máy tính bảng đồ họa, các bút màn hình, chuột, microphones and keyboards.

**Step 7:** Chọn ra các controls màn hình

“Controls màn hình” là các biểu tượng xuất hiện trên màn hình, user sẽ chú ý đến phần input của ứng dụng, bao gồm icons, buttons, text boxes, selections, and radio buttons. Các quy tắc sắp xếp các controls trên một cửa sổ giống với thiết kế màn.

**Step 8:** tổ chức layout windows.

Các quy tức lên lay out cho nhiều cửa sổ tương tự như thiết kế từng màn hình của chúng, bao gồm có sự đối xứng, cân bằng, … Nhưng chúng có liên quan đến sắp xếp và phân tầng, được minh họa trong hình dưới.



**Step 9:** Chọn màu phù hợp

Color không tự động làm tạo ra một user interface hữu dụng hơn hoặc hấp dẫn hơn, tuy nhiên, nó có thể dễ dàng làm xấu đi thiết kế. Theo nhà thiết kế danh tiếng Paul Rand, “mầu sắc là tính cách phức tạp”. Các kỹ sư phần mềm không phải là những nhà thiết kế chuyên nghiệp. Luôn sử dụng màu đen hoặc màu trắng trước tiên. Nếu nó làm rõ được mục đích, hãy giới thiệu mọt màu khác. Hãy đảm bảo nó có thể giúp ích được user. Suy nghĩ một cách nghiêm túc trước khi thêm vào một color.

Note lại các ứng dụng được làm tốt, chẳng hạn bộ sử lý word được sử dụng rộng rãi, có thể gợi ý các màu tốt để sử dụng. Bạn có thể được bảo đảm cho các giao diện được thiết kế chuyên nghiệp cho trải nghiệm, các kỹ sư phần mềm chưa thạo có thể được lợi từ điều này.

# 12.4 Các yêu cầu bảo mật chi tiết

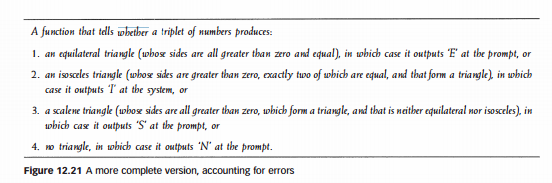
Viết các yêu cầu cụ thể cho các biện pháp bảo mật là một vấn đề hoàn toàn cụ thể và cấp bách.

Một vài trong số đó được thể hiện ở hình 12.18 và 12.19. Khi thoát khỏi bảo mật, ví dụ, chúng ta có muốn yêu cầu rằng liệu không có keystroke trên ứng dụng trong khoảng 10 phút, nó sẽ tự động đăng xuất. Kiểm tra các lỗ hỗng bảo mật là cực kỳ quan trọng. Một vài yêu cầu dễ dàng đủ để xác định, nhưng khó mà thực thi. Chẳng hạn, làm thế nào một ứng dụng biết có ai đó đã phá vỡ bảo mật.

# 12.5 Các điều kiện lỗi

Mỗi yêu cầu, chúng ta hỏi chuyện gì xảy ra nếu nó bị lỗi. Ví dụ, làm một yêu cầu như hình 12.20: một hàm nhận 3 số để tạo một tam giác đều, một tam giác cân, hoặc một tam giác thường

Đặc tả yêu cầu này không thể hoàn thành được vì nó không quản lý hết các lôi về điều kiên. Phiên bản ở hình 12.12 thì đầy đủ hơn.



Dịch: Một hàm nhận 3 số, liệu có thể là vẽ ra được:

1. Một tam giác đều, có 3 cạnh bằng nhau và lớn hơn 0, và xuất ra màn hình “E”
2. Một tam giác cân, có 3 cạnh bằng nhau và lớn hơn 0 và chỉ có 2 cạnh bằng bằng nhau, và xuất ra màn hình “I”.
3. Một tam giác thường, có 3 cạnh lớn hơn 0, và không có cạnh naoof bằng nhau, xuất ra màn hình “S”.
4. Không có một tam giác nào, xuất ra màn hình “N”.

Thiếu điều kiện lỗi trong đặc tả yêu cầu dẫn đến function cần phải được test, vì tester tập trung vào điều kiện lỗi, và cần biết yêu cầu về output.

Phân tích yêu cầu không trả về input bất hợp lệ: nó giải quyết với chúng một cách trực tiếp. Chẳng hạn, nó đang tạm thời để giả sử rằng GUI cho yêu cầu về tam giác không cho phép input số âm, và vì thế function không phải giải quyết với lỗi trên. Tăng sự phụ thuộc vào các phần ứng dụng, những nơi chúng ta có thể đạt được các phần độc lập. Mặc dù nó là một thói quen tốt để bẫy các lỗi user input không hợp lệ tại tầng GUI và để đảm bảo user chỉ nhập vào những giá trị hợp lệ, điều này không thay để cho một đặt tả tốt và nhận dạng lỗi. Các tác giả khuyến nghị yêu cầu các bẫy về lỗi dữ liệu từ input tại một vài điểm, không phải tất cả. Điều này tương đương với thói quen kỹ sư được thành lập từ từ.

# 12.6 Nguồn gốc của các yêu cầu chi tiết

Hãy tưởng tượng một ứng dụng với 1.000 yêu cầu cụ thể. Nếu không có một mô tả rõ ràng từ từng yêu cầu thông qua việc thiết kế các ứng dụng để mã thực tế mà thực hiện nó, nó là rất khó để đảm bảo rằng một ứng dụng như vậy vẫn còn phù hợp với yêu cầu. Khi các yêu cầu thay đổi, đó là an toàn để giả định, điều này trở nên khó khăn hơn. Khả năng để mỗi yêu cầu chi tiết cho một phần của nó có liên quan (s) của thiết kế và thực hiện được gọi là truy xuất nguồn gốc. Đầu tiên chúng ta thảo luận về truy xuất nguồn gốc của yêu cầu chức năng, sau đó yêu cầu chức năng.

Một cách để giúp thực hiện truy xuất nguồn gốc là đồ mỗi yêu cầu chi tiết chức năng để một chức năng cụ thể của ngôn ngữ đích. Kỹ thuật này được sử dụng trong các nghiên cứu trường hợp. Hình 12,22 cho thấy các phần của dự án mà chúng tôi muốn tiếp tục liên kết có nguồn gốc cụ thể. Đạt được và duy trì mức độ này truy xuất nguồn gốc trong quá trình phát triển là một thách thức lớn.

Như một ví dụ, hãy xem xét các yêu cầu chức năng sau cho các nghiên cứu trường hợp trò chơi Encounter video.

Khi một nhân vật game nước ngoài tiến vào khu vực chứa vật chính của người chơi, hoặc ngược lại, họ tham gia vào các phần khác.

Những gì còn lại đã được nhìn thấy, tuy nhiên, những gì mà một phần hoặc các bộ phận của thiết kế và mã sẽ chịu trách nhiệm thực hiện yêu cầu này. Khi sử dụng 00 mô hình, chúng tôi có thể liên kết được yêu cầu này đến một chức năng cụ thể của một lớp học cụ thể. Các vấn đề về những gì lớp có trách nhiệm một chức năng là không nhỏ, và nó phát sinh nhiều lần khi sử dụng kiểu 00.

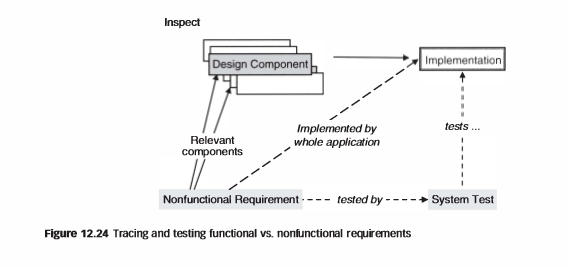
Như là quá trình thực hiện dự án, các tài liệu yêu cầu phải được giữ phù hợp với việc thiết kế và thực hiện. Khi yêu cầu này rất khó để theo dõi thông qua thiết kế và mã số, tuy nhiên, các nhà phát triển có xu hướng tránh việc cập nhật các tài liệu yêu cầu khi thực hiện thay đổi mã nguồn vì các nỗ lực mở rộng yêu cầu. Cuối cùng, như một sự suy giảm của các tài liệu dẫn đến leo thang chi phí phát triển và bảo trì. Hiện tượng này được minh họa bằng các ví dụ sau đây.

Khi một tài liệu yêu cầu như một toàn thể là không đáng tin cậy, thậm chí các nhà phát triển có lương tâm nhất phản đối các động cập nhật phần riêng biệt của mình. Mặt khác, khi các tài liệu rõ ràng là tham khảo chéo, và quản lý làm cho tài liệu là một yêu cầu thực hiện công việc, các kỹ sư giữ chúng trong hình dạng chuyên nghiệp rất tốt. Nói cách khác, hệ thống sử dụng để phù hợp với yêu cầu chi tiết với các mẫu thiết kế, và mã thực hiện chúng, phải là rất rõ ràng và cụ thể.

Khi mã thực hiện một yêu cầu cần phải tồn tại trong một số bộ phận của việc thực hiện, truy tìm được thực hiện bằng phương tiện của một ma trận các yêu cầu truy xuất nguồn gốc trong đó Hình 12,23 là một ví dụ. Ở đây, yêu cầu năm 1783 được thực hiện bởi các hành động của các chức năng getlnterest () trong module 1 computeBal () trong phần 2, và showName () trong mô-đun 3. Một sự thay đổi trong yêu cầu này đòi hỏi một sự thay đổi trong một hoặc nhiều hơn các chức năng. Điều này phải được quản lý một cách cẩn thận bởi vì các chức năng này có thể tham gia vào các yêu cầu khác đáp ứng (ví dụ, showName () được sử dụng để thực hiện yêu cầu 1784 cũng). Kết quả là, những thay đổi được thực hiện để đáp ứng một yêu cầu có thể thỏa hiệp khác. Vì nhiều-nhiều mối quan hệ rất khó để quản lý, chúng tôi cố gắng để làm cho các ánh xạ giữa các yêu cầu và chức năng một-một.

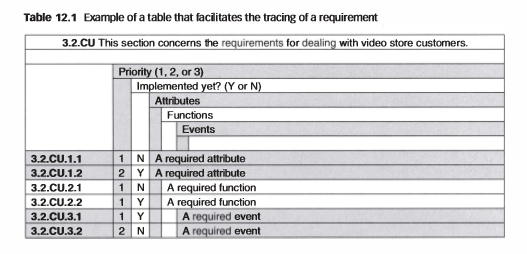
Chúng tôi muốn mỗi yêu cầu chi tiết để được theo dõi về phía trước và lạc hậu. Các mối quan tâm thảo luận trước về phía trước truy xuất nguồn gốc của yêu cầu chức năng từ yêu cầu chi tiết để thực hiện. truy xuất nguồn gốc lạc hậu của một yêu cầu chi tiết có nghĩa là yêu cầu là một kết quả rõ ràng của một hoặc nhiều yêu cầu cao cấp. Ví dụ, các yêu cầu chi tiết.

Để xác nhận yêu cầu không có chức năng do đó chúng tôi buộc mỗi người một kế hoạch kiểm tra, tốt nhất là vào thời điểm viết yêu cầu. Hình 12.24 minh họa mối quan hệ điển hình của các yêu cầu chức năng và không có chức năng để thực hiện và thử nghiệm, thảo luận ở trên. Nó minh họa thực tế là một số yếu tố có thể góp phần yêu cầu không có chức năng, và rằng hệ thống hoặc tích hợp thử nghiệm thường được yêu cầu phải xác nhận yêu cầu không có chức năng xác minh vì họ (tức là, trước khi thực hiện) có thể khó khăn.



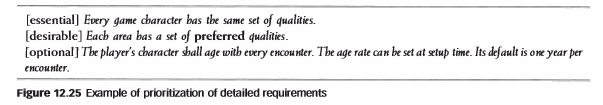
# 12.7 Sử dụng các yêu cầu chi tiết để quản lý dự án

Yêu cầu chi tiết có thể được xem là "tiền tệ" của một dự án vì họ theo dõi các yêu cầu được hoàn thành. Ví dụ, một người quản lý dự án có thể theo dõi các dự án như trong Bảng 12.1. Nhớ lại rằng điều quan trọng cho các nhà phát triển để nhanh chóng tìm thấy yêu cầu bộ phận liên quan giúp họ có thể dễ dàng giữ một tài liệu yêu cầu mới cập nhật. Trong các định dạng IEEE yêu cầu, yêu cầu chức năng nằm trong Phần 3.2. Nếu chúng ta đánh số các đoạn 3.2. 1, 3.2.2, ..., nó sẽ tốn thời gian để xác định vị trí các đoạn liên quan đến một lớp khách hàng, ví dụ. Vì lý do này, các tác giả sử dụng một nhãn hiệu chữ và số như 3.2.CU Cho Khách Hàng, 3.2.DV cho DVD, …

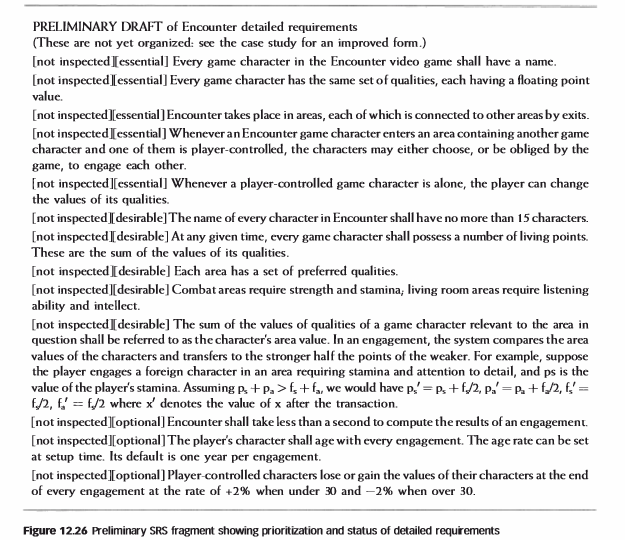


# 12.8 Các yêu cầu ưu tiên

Nó thường rất khó khăn, nếu không muốn nói là không thể để thực hiện tất cả các chức năng mong muốn của một ứng dụng đúng tiến độ và trong phạm vi ngân sách. Như đã thảo luận ở Chương 8, người ta có thể đánh đổi khả năng, tiến độ, mức độ chất lượng và chi phí. Như vậy, nếu mức kế hoạch, ngân sách, và chất lượng không thể thay đổi, chỉ yêu cầu là thay đổi khả năng-có nghĩa là, để giảm bớt các yêu cầu đó được thực hiện. quá trình giảm này được thực hiện trong một cách có kế hoạch. Một kỹ thuật là để ưu tiên các yêu cầu cụ thể. Bảng xếp hạng tất cả các yêu cầu thường là một sự lãng phí thời gian. Thay vào đó, nhiều tổ chức phân loại yêu cầu này thành ba (đôi khi bốn) loại. Chúng tôi sẽ gọi họ là "essentiaI." "Mong muốn", và "tùy chọn." Việc sử dụng ba loại là một ứng dụng của phân loại được mô tả trong Chương 5. Đầu tiên chúng tôi thực hiện tất cả các yêu cầu thiết yếu. Các yêu cầu mong muốn và tùy chọn chỉ hướng trong đó các ứng dụng được lãnh đạo và do đó ảnh hưởng đến thiết kế. Hình 12,25 cho một ví dụ về yêu cầu ưu tiên. Một số suy đoán rằng có tới 80 phần trăm của lợi ích thực sự của nhiều ứng dụng tích luỹ từ vài như là 20 phần trăm của các yêu cầu. Như vậy, nếu ưu tiên được thực hiện tốt (ví dụ, gọi khoảng 20 phần trăm, không thêm- "thiết yếu ''). Một người nào có thể đạt được hầu hết các lợi ích của một ứng dụng với chỉ một phần nhỏ của công việc.

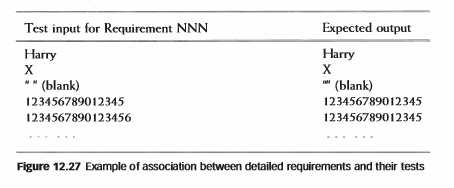


Đây là một điểm hữu ích để giữ trong tâm trí nếu dự án bắt đầu chạy ra khỏi thời gian. các thảo sơ bộ của một SRS thể hiện trong hình 12.26 có chứa một số yêu cầu chi tiết ưu tiên cho việc phát hành đầu tiên của Encounter. Họ được cung cấp ở đây, "mụn cóc" và tất cả, để cung cấp cho người đọc một cảm giác về vấn đề cần phải được xử lý. Một số "desirables" sẽ trở thành "thiết yếu" trong phiên bản tương lai. các yêu cầu ở dạng dự thảo và rõ ràng là cần tổ chức lại. Họ được cải thiện sau khi sau này trong chương này và trong các trường hợp nghiên cứu. các ưu tiên các yêu cầu thường liên quan đến việc lặp đi lặp lại rằng sẽ thực hiện chúng. Ví dụ, nếu chúng ta không thể thực hiện "tùy chọn" yêu cầu "Encounter sẽ mất ít hơn một giây để tính toán các kết quả của một sự dấn thân" trong phiên thứ hai, nó có thể xuất hiện với ưu tiên cao hơn trong một lần lặp tiếp theo. Các yêu cầu cho một sự lặp lại được duy trì một cách nhận dạng. Điều này giúp trong việc tìm hiểu các yêu cầu tiếp theo.



# 12.9 Liên kết các yêu cầu với các bài kiểm tra

Khi mỗi yêu cầu chi tiết được viết, kiểm tra đối với các yêu cầu cần được phát triển. Có một số lợi thế để viết bài kiểm tra đồng thời với yêu cầu. Đầu tiên, làm như vậy sẽ giúp làm rõ các yêu cầu cụ thể. Thứ hai, nó còn thay đổi một số công việc từ giai đoạn thử nghiệm của dự án đến giai đoạn yêu cầu. Điều này làm giảm một số áp lực về nửa cuối của dự án khi có ít linh hoạt trong việc sử dụng thời gian. Quy trình Agile đi thêm một bước nữa, và chỉ định mỗi yêu cầu chi tiết bằng phương tiện của một thử nghiệm.Yêu cầu của các loại thuộc tính như thế này thực sự cụ thể chức năng GET và set-, vì vậy mà hình 12.27 tạo sự khởi đầu của một kế hoạch kiểm tra cho yêu cầu này. Phần VII bao gồm các xét nghiệm cụ thể.

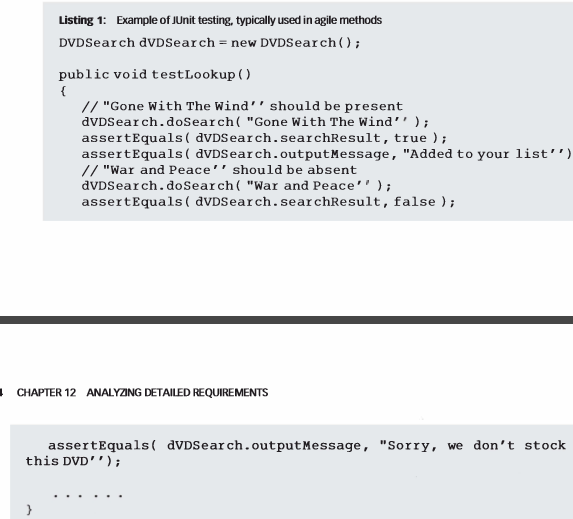


# 12.10 Phương pháp Agile cho các yêu cầu chi tiết

Trong một dự án agile, các yêu cầu chi tiết thường được thể hiện trong các điều kiện của các dòng comment code, và test unit. Ví dụ, xem xét một yêu cầu truyền thống chẳng hạn như dưới đây.

Các khách hàng nên có thể nhập vào tên của một DVD trong một textbox, lên đến 20 ký tự. Ứng dụng nên kiểm tra điều này, với các dấu chấm đánh dấu được thay thế bởi các khoảng trống, với các kho lưu trữ DVD, và hiển thị theo “Sorry we don’t stock this DVD” hoặc “Added to your list”.

Điều này được thay thế với một hoặc nhiều test như được show trong list 1, được tăng sự rõ ràng với các đoạn comment và điểm truy cập dữ liệu tương đương trong giao diện GUI tương ứng. Giả sử rằng đoạn code được setup trong một unit test chứa:



Lợi ích của việc sử dụng test như một yêu cầu là vì nó cố định, điểm yểu là nó không được hoàn thành. Không có bất cứ gì ngăn cản ta từ việc bao gồm các yêu cầu chi tiết với các unit test cố định, tuy nhiên, đạt được sự linh hoạt và để được một vài sự cố định, các thuận lợi không linh hoạt.

Khi các unit test được sử dụng như các yêu cầu chi tiết, chúng thương được viết trước khi code được viết. Phong cách này được biết như một sự phát triển vòng lặp test. Nó xây dựng nền tảng code đang tồn tại, và được tiến hành trong chuỗi dưới đây.

1. Hiểu yều mới hoặc thay chức năng được thay đổi
2. Nếu cần, tái thiết lập lài code để chuẩn bị cho các chức năng mới. Tái thiết lập bảo quản các chức năng code, hoặc tăng hoặc giảm các chức năng thực sự của nó. Nó được thảo luận chi tiết trong chương 24.
3. Viết test code sẽ test chức năng này nếu nó tồn tại. Điều này thường được thực hiện với một công cụ như JUnit. Test này sẽ khơi tạo fail, vì các test cho chức năng này không tồn tại.
4. Thêm vào code đến khi test được pass
5. Nếu cần, tái cấu trúc lại code để làm nó rõ ràng, toàn vẹn

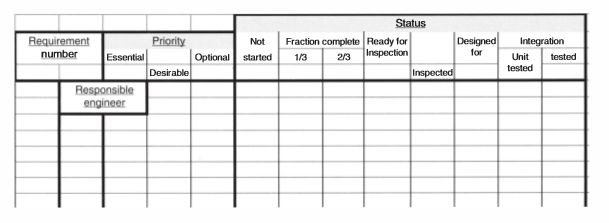
Phát triển test được một tả kỹ hơn trong chương 27

Nhớ lại phát triển agilebao gồm chu kỳ như hình 12.30. Tái thiết lập, được mô tả để xử lý form (không phải chức năng) của coe để chuẩn bị cho thêm vào một chức năng mới. Nó cũng được sử dụng để làm kết quả thêm vao vừa vặn, trôi chảy với ứng dụng, cũng như tất cả.

# 12.11 Sử dụng công cụ và web trong phân tích các yêu cầu

Các công cụ có thể giúp xử lý bắt lấy, quản lý yêu cầu – chẳng hạn, sorting, categorizing, ưu tiên, phân công, theo dõi. Một quyền lợi của các công cụ đó là biết được ai đang làm việc trên yêu cầu nào tại thời điểm hiện tại. Các công cụ có thể giúp để điều khiển các tính năng, xử lý bởi tính nào, nó thực sự ko cần thiết được thêm vào ứng dụng. Với công cụ thích hợp, mọt leader của dự án có thể đánh giá dễ dàng hơn trạng thái phân tích yêu cầu. Chẳng hạn, một leader có thể xác định một cách dễ dàng tỉ lệ của yêu cầu chi tiết cần thiết được thực thi và được test đầy đủ bởi QA chưa., một công cụ open source để quản lý các yêu cầu, được mô tả trong Eclipse, các yêu cầu cấp cao. Phần này cũng thảo luận quản lý yêu cầu cho dự án đơn giản, cũng như công cụ thương mại RequisitePro của IBM.

## 12.11.1 Dự án đơn giản

Với dự án đơn giản, hầu như nó có thể được thực hiện sủ dụng một bảng tính trên web, được minh hoạt trong hình 12.31hê

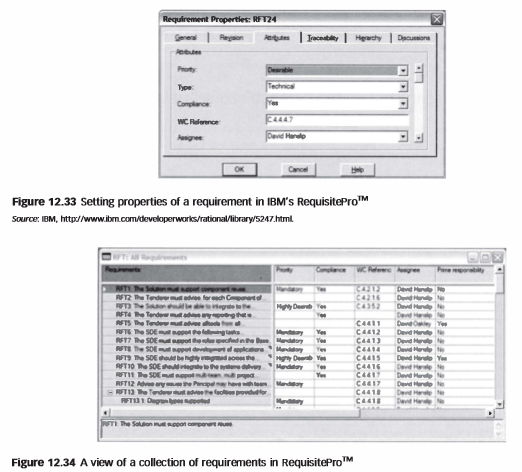
Tuy nhiên, hầu hết yêu cầu dự án có kích thước hợp lý là cần thiết. Thiết kế cho quá trình thiết kế chỉ ra rằng các yêu cầu được quản lý cho thiết keess. Unit test có nghĩa các yêu cầu đang thực thi trong code đã chịu sự thay đổi từ thiết kế độc lập. Intergration test có nghĩa là ứng dụng được test để xác định rằng nó thực thi yêu cầu. Một bảng trong hình 12.31 được duy trì như một phần của project.

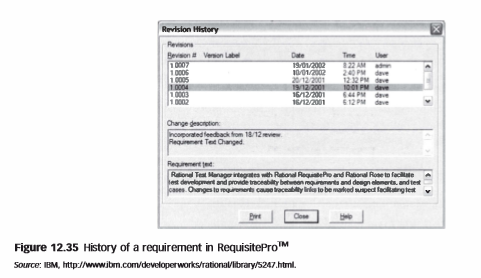
Các ô trong ma trận có thể nổi bật, làm rõ các phần trong document của project, vì thế, bảo quản tài liệu source code cho yêu cầu chi tiết.

Khuynh hướng nâng cấp liên tục trong quy trình bới những lập trình viên sẽ có thể quay lại dễ dàng và 4 giữa SRS, thiết kế, đồ họa giao diện và source code.

## 12.11.2 RequisitePro của IBM

Một dự án trọng yếu, các công cụ chuyên nghiệp được cần để theo dõi yêu cầu. Số lượng yêu cầu thường cần thiết. Một ví dụ sản phẩm RequisitePro của IBM. Các hình dưới đây mô tả điều đó.

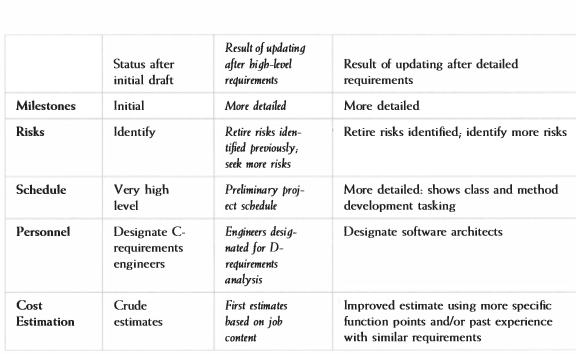




Nó có thể truy vấn cơ sở dữ liệu yêu cầu, hay chung quy, để toàn bộ yêu cầu thỏa mãn tiêu chí mong muốn, chẳng hạn bảo mật. Như các yêu cầu phân biệt được thực hiên bởi kỹ sư công nghệ phần mềm, các công cụ như vậy cho phép theo dõi lịch sử lướt web.

# 12.12 Sự ảnh hưởng đến dự án của quá trình các yêu cầu chi tiết

Một khi yêu cầu chi tiết đã được thu tập, các tài liệu của dự án có thể update để phản xạ kiến thức dự án được cải thiện. Chúng ta sẽ lấy một ví dụ về yêu cầu update để SPMP, có thể update nhu hình 12.36



Yêu cầu chi tiết được thay thể trong điều khiển cấu hình. Một vấn đề được đánh dấu level chi tiết gì nên được đếm như một item cấu hình (CI). Chẳng hạng, mỗi lớp có thể CI. Các yêu cầu riêng biệt thương có 2 cách để trở thành CI.

Khi list yêu cầu phát triên lên hàng trăm, sự không nhất quán có thể gia tăng. Tổ chức lớp yêu cầu theo lơp, các lớp được quản lý theo package bởi từng trang, và điều tương tự xảy ra với ta. Các pakeage thường tương ứng với hệ thống con trong tổ chức của ứng dụng..

Mặc dù sự hoàn thành là mục tiêu để ta phấn đấu trong tập hợp yêu cầu, nó thường là mục tiêu tương minh. Để ứng dụng chủ yếu, có sự thoải mai về yêu cầu cuối cùng. Vì nguyên nhât đó, chugns ta phấn đấu dự hoàn thành: đảm bảo dữ liệu cần thiết theo mỗi yêu cầu.

Dự án lớn yêu cầu tăng khả năng tổ chức. SRS có thể được chia thành các module phân biệt. Một phần nhỏ có thể mở rộng thành 700 phần nhỏ hơn.

# 12.13 Hướng dẫn dự án: Các yêu cầu cho nghiên cứu trường hợp đối kháng.

Phần này mô tả cách các nguyên tắc yêu cầu được đề cập trong cuốn sách này được dịch sang thực tế, bằng cách sử dụng các nghiên cứu về trò chơi video như là một ví dụ. Nó sử dụng phương pháp hướng đối tượng để thể hiện yêu cầu. Nhớ lại rằng cách tổ chức này có lợi thế là được cải thiện truy xuất nguồn gốc và những bất lợi của khả năng đọc giảm đi so với các cách tổ chức khác của trường hợp sử dụng. Các trường hợp nghiên cứu sẽ được tiếp tục trong các chương sau.

1. Chuẩn bị:

Hal Furness, đã được bầu làm đội trưởng yêu cầu, chịu trách nhiệm tổ chức việc phân tích các yêu cầu. Theo các tổ chức dự án, Hal đã được sao lưu bởi Karen Peters. Họ quyết định để thu thập các yêu cầu trong hai giai đoạn. Đầu tiên sẽ được chủ yếu từ quan điểm (yêu cầu trình độ cao) của khách hàng, và lần thứ hai chủ yếu dành cho nhà phát triển (yêu cầu chi tiết).

Hal và Karen chuẩn bị để thu thập số liệu về quá trình yêu cầu. Họ phân loại các giai đoạn của quá trình bằng cách chuẩn bị, phỏng vấn, viết lại và xem xét. Các số liệu họ đã chọn được quyết chủ yếu là do chính sách của công ty, và như sau:

* Thời gian tiêu tốn.
* Yêu cầu theo từng mức.
* Tự đánh giá của các đồ tạo tác trên thang điểm từ 1-10 (không bắt buộc theo chính sách của công ty).
* Điểm yếu tìm thấy trong quá trình thanh tra, nếu có thể áp dụng.

Người đọc được tham khảo phần 4 của hướng dẫn này để xem các số liệu được sắp xếp theo dạng bảng.

Karen đã chắc chắn rằng hệ thống để đăng nhập và theo dõi các khiếm khuyết được đặt ra, và Hal đã được trang bị các tài liệu hướng dẫn về cách sử dụng nó.

Các nhà đầu tư của công ty được xem trò chơi video nhu la một khu vực đầy hứa hẹn, và đã sẵn sàng để cung cấp tiền giống cho yêu cầu phân tích và một nguyên mẫu. Bây giờ, Hal và Karen có nhiệm vụ để xác định người để nhận yêu cầu cao cấp. Hal hiểu rằng không ai trong số các đội đã biết nhiều về trò chơi video. Ông quyết định phỏng vấn những người thường xuyên chơi trò chơi và quan tâm dành thời gian của họ cho một khoản phí nhỏ. Ông đã liên lạc với Betty Sims, Chủ tịch của Amateur Garners International, một người chơi trò chơi nhiệt tình nhìn thấy một tương lai tươi sáng cho các trò chơi video như đua xe, sự tham gia của cộng đồng, và giáo dục. Betty cũng biết nhiều nguồn có thể thu hút được. Hal và Karen quyết định viết lên yêu cầu kỹ thuật dựa trên đầu vào của Betty và sau đó hiển thị các thông số kỹ thuật cho người khác. Phần còn lại của nhóm là để điều tra những con đường khác cho đầu vào cùng một lúc.

Tại một cuộc họp hàng tuần, Hal trình bày một kế hoạch để phân tích các yêu cầu như sau:

Tuần 1:

Hal và Karen: Phỏng vấn Betty; bắt đầu soạn thảo yêu cầu cao cấp.

Fern và AI: Tìm kiếm các ứng cử viên khác cho các yêu cầu đầu vào.

Tuần 2:

Fern và AI: Báo cáo ứng cử viên cho cuộc họp hàng tuần.

Đội bóng: Chọn một hoặc hai người bổ sung để cung cấp các yêu cầu.

Hal và Karen: Hoàn thành dự thảo yêu cầu cấp cao, e-mail cho Betty để lấy ý kiến; sắp xếp để phỏng vấn những người khác được chỉ định; e-mail đặc điểm kỹ thuật cho họ hiện có; tạo ra một tài liệu yêu cầu hoàn chỉnh cho phép lặp 1; đặt dưới sự kiểm soát cấu hình.

Tuần 3:

Đội bóng: Phê duyệt SRS cho phép lặp 1.

Hal và Karen: Phỏng vấn được nhân dân; chỉnh sửa và mở rộng các đặc điểm kỹ thuật; e-mail đến tất cả người được phỏng vấn; đối chiếu các câu trả lời; chỉnh sửa tài liệu, để lại những vấn đề được lựa chọn để giải quyết nhóm; có kế hoạch yêu cầu phân tích chi tiết (xem Phần V của cuốn sách này).

Tuần 4:

Đội bóng: Cung cấp các inputon draftSRS; Phê duyệt kế hoạch để phân tích các yêu cầu chi tiết (xem Phần V ofthis cuốn sách).

Hal và Karen: Viết lên SRS và e-mail đến tất cả người phỏng vấn.

Tuần 5:

Hal và Karen: Giải quyết vấn đề do người được phỏng vấn; viết lên kết quả; e-mail để đội; bắt đầu thực hiện chi tiết quá trình yêu cầu (xem Chương 13).

Đội bóng: Kiểm tra yêu cầu cao cấp.

Mặc dù chi phí, Hal cảm thấy nó quan trọng để có toàn bộ đội ngũ kiểm tra các yêu cầu cấp cao vì tầm quan trọng của tài liệu. Nói chung, nhóm nghiên cứu dự định sử dụng các đội kiểm tra ba người. Hal dự buổi phỏng vấn đầu tiên với Betty trong phòng 1428 của tòa Stewart từ 10:00 đến 11: 30 sáng Ông e-mail của mình một cách ngắn gọn viết lên lịch sử của dự án, và tạo ra các chương trình rất đơn giản sau đây.

10:00-IO: 15 sáng Hal: động cơ cho dự án

10: 15 sáng-I 01:30 Phỏng vấn của Betty: yêu cầu khách hàng

Hal quyết định không giới thiệu chi tiết hơn vì anh muốn yêu cầu của Betty để ảnh hưởng đến phần còn lại của cuộc họp.

2. Phỏng vấn khách hàng

Hal và Karen đến các cuộc phỏng vấn với Betty, được trang bị với thiết bị ghi âm tốt. Betty không thể hiểu tại sao người ta muốn xây dựng một trò chơi video trừ khi nó cạnh tranh với các tốt nhất. Hal giải thích rằng đây chỉ là một bước đầu tiên, để cung cấp cho các nhóm có kinh nghiệm loại này của chương trình, để có được một ý tưởng về phạm vi công việc được yêu cầu, và cho thấy các nhà đầu tư những gì có thể được thực hiện với một số tiền nhất định của nguồn tài trợ. Động cơ khác là xác định xem có bất kỳ bằng khen cho các ý tưởng rằng các trò chơi video có sức hấp dẫn tiềm năng rộng, và được áp dụng cho giáo dục. Sau này, cuộc họp trở nên tập trung hơn. Các máy ghi âm đã được bật và Hal và Karen đã bắt đầu để ghi chép chi tiết.

Vai trò Betty là trò chơi nhập vai (không phải trò chơi hành động) tổ chức hứa hẹn nhất cho việc mở rộng cộng đồng người chơi. Cô đã thảo luận về khả năng tối thiểu mà một nguyên mẫu sẽ cần. Điều này bao gồm các khu vực nơi mà các nhân vật trò chơi sẽ tham gia, cách để có được từ vùng này sang vùng khác, cách để gây ra tương tác giữa các nhân vật, và những gì sẽ xảy ra khi các nhân vật tham gia. Hal và Karen đã cố gắng để tách các vấn đề và các tính năng như chúng xuất hiện, vào "quan trọng cho phiên đầu tiên", "có thể được hoãn lại," và "khác" (tức là họ đã sử dụng một phương pháp phân loại). Tầm quan trọng của các yêu cầu được liệt kê trong "khác" sẽ được xác định sau.

Do tính chất kịch bản giống như các yêu cầu Betty mô tả, Hal tập trung vào việc thu thập các trường hợp sử dụng từ cô ấy. Ông hỏi cô ấy để mô tả tình huống điển hình cho các trò chơi. Betty đã mô tả những gì sẽ xảy ra khi hai nhân vật tương tác. Karen đã ghi chép và trình bày điều này như là một sử dụng trường hợp một chuỗi các hành động được thực hiện bởi người chơi và / hoặc các trò chơi và sau đó đọc lại cho Betty.

Betty không thể nghĩ ra bất kỳ kịch bản khác. Hal cảm thấy rằng phải có nhiều hơn, và hỏi làm thế nào các trò chơi được bắt đầu. Điều này dẫn đến một trường hợp sử dụng thứ hai. Các trường hợp sử dụng thứ ba mà họ nhận biết giải thích thế nào người chơi di chuyển nhân vật của mình từ vùng này sang vùng khác. Ba trường hợp sử dụng dường như là một khởi đầu khả quan. Hal và Karen cảm thấy rằng có thể có thêm các trường hợp sử dụng thiết yếu, nhưng họ sẽ phải thu thập chúng sau này.

Betty, Karen, và Hal phác thảo một vài màn hình với nhau. Một cho thấy một cuộc gặp gỡ thông thường, và một người khác đã cho thấy một màn hình để nhập những phẩm chất của nhân vật game. Đã thảo luận đáng kể trong những quan điểm mà người chơi sẽ có. Betty muốn có một quan điểm phát có nhìn thấy trên màn hình là quan điểm nhìn thấy bởi người chơi. Karen cảm thấy rằng sự phức tạp yêu cầu cho xem mà có thể đưa dự án cũng vượt quá ngân sách ban đầu khiêm tốn của họ. Nó đã được đồng ý rằng một biến đổi từ-trên xem sẽ được đầy đủ cho các mẫu thử nghiệm. Các phác thảo màn hình phản ánh điều này. Họ đồng ý rằng tinh tế đáng kể về giao diện người dùng sẽ được yêu cầu.

Do các giao diện mà họ phác thảo, Karen cảm thấy rằng các trò chơi thực sự có thể được hiểu chỉ bằng phương tiện của các quốc gia. Betty đã không quen với thuật ngữ này, nhưng cô cảm thấy thoải mái diễn tả những gì cô gọi là "chế độ" của một trò chơi nhập vai điển hình, mà hóa ra lại là một khái niệm. Karen và Hal sau đó phác thảo ra các tiểu bang yêu cầu của trò chơi, và xem xét lại với Karen thế trận được từ một tiểu bang khác.

Hal brieRy coi làm rõ trò chơi hơn nữa bằng cách phân tích các Row của dữ liệu, nhưng sớm nhận ra rằng quan điểm dữ liệu Row thêm ít giá trị. Karen xét cô lưu ý với những người khác. Một vài điểm cần sửa chữa, nhưng có thỏa thuận chung về mô tả.

3. Viết Up Phần mềm Yêu cầu kỹ thuật

Hal và Karen chia nhiệm vụ viết lên SRS ra từng phần. Họ đã sử dụng các tiêu chuẩn IEEE SRS, Phần 1 và 2 (Phần 3 bao gồm các yêu cầu chi tiết, quá trình đó được thảo luận trong Hướng dẫn dự án của sinh viên về chương 4). Để tránh conRicting write-up, họ đã chắc chắn rằng các phần của họ là độc lập càng tốt. Hal nhớ dự án trước đây của mình, nơi mà các đội đã dành rất nhiều thời gian hòa giải những tác phẩm viết bởi những người khác nhau mà nó đã nhanh hơn cho một người để thực hiện toàn bộ công việc một mình.

Họ đã thảo luận làm thế nào để ưu tiên các yêu cầu, bởi vì nó đã trở nên rõ ràng rằng nếu không danh sách các yêu cầu sẽ trở nên lớn hơn nhiều so với nhóm có thể xử lý. Hal muốn để xếp hạng tất cả, nhưng Karen đã chỉ ra rằng các nỗ lực tham gia sẽ bị lãng phí, nhất phần lớn các yêu cầu hàng đầu sẽ được thực hiện anyway, do đó thứ tự chính xác của họ sẽ không quan trọng. Hầu như không ai trong số những người phía dưới sẽ được thực hiện, vì vậy thời gian dành xếp hạng chúng cũng sẽ bị lãng phí. Họ quyết định sử dụng một phương pháp phân loại để xếp hạng các yêu cầu vào thiết yếu tại một cực đoan, tùy ý tại khác, và mong muốn cho các loại trung (trong đó chỉ có nghĩa là không cần thiết và cũng không bắt buộc). Họ cảm thấy rằng nó có thể là cần thiết để xếp hạng các yêu cầu mong muốn sau này. Điều này tiết kiệm được một lượng lớn thời gian tranh luận vô ích. Họ mô tả sơ đồ phân loại của họ trong phần 2.6 của SRS ( "tỷ lệ án yêu cầu").

Mục 2.1.1 (khái niệm về hoạt động, chứa sơ đồ nhà nước cho các trò chơi) đã Hal thời gian nhất để viết, vì ông đã phải dịch ý kiến chính thức của Betty vào một hình thức cụ thể. Họ đã cố gắng để thích hợp các phần tham chiếu chéo của SRS với các xét nghiệm mặc dù các bài kiểm tra vẫn còn sơ sài tương ứng. Điều này giúp làm rõ các yêu cầu tự. Khi Betty nhìn vào thử nghiệm cho Mục 2.1.1, cô nhận ra rằng Hal và Karen đã không hiểu một số vấn đề. Betty nhận thấy đây là bất lợi và là một cách để các cầu thủ để ngăn chặn có hiệu quả các trò chơi. Các khiếm khuyết đã được bổ sung vào danh sách các khiếm khuyết với một loại "lớn". Karen phác thảo các giao diện người dùng bằng PowerPoint ™ là một công cụ vẽ, thay vì xây dựng chúng với Java, ngôn ngữ đích. Cô coi PowerPoint đủ vì ULS trong phần này của SRS có nghĩa là để được phác thảo, các chi tiết ULS được quy định tại Mục 3, và trong mọi trường hợp, họ đã chịu được thay đổi rất nhiều. Điều này đã giúp Hal và Karen để hiển thị các bản phác thảo để Betty và những người khác, có được thông tin phản hồi, và sau đó xác định ULS chính xác cho các yêu cầu chi tiết.

4. Following up

Các mục SRS 1 và 2 đã được e-mail đến Betty. Cô nhận ra rằng Hal và Karen đã chỉ bao gồm hai trong ba trường hợp sử dụng, và các trường hợp sử dụng thứ ba mô tả chuyển động của nhân vật của người chơi đã vắng mặt. khiếm khuyết này đã được ghi lại với một ưu tiên cao.

Betty đã ngạc nhiên khi thấy rằng SRS không reRect một số vấn đề mà cô nghĩ rằng cô đã nói rõ là quan trọng, và được hạ xuống để thấy rằng SRS reRected "yêu cầu" cô cách tự nhiên đề cập nhưng bây giờ nhận ra sẽ là một sự lãng phí thời gian. Sau này bao gồm khả năng của các cầu thủ để thay đổi trang phục trong khi tham gia đang tiến triển. Cô đã có nhiều ý kiến, hầu hết trong số đó Hal và Karen trả lời, và một số trong đó đã được thêm vào danh sách các khiếm khuyết. Hal qua e-mail các mục SRS (và 2 cho đội để họ có thể chuẩn bị cho một cuộc thanh tra.

Trưởng nhóm Ed đã học về Arlan Howard, một giám đốc tiếp thị những người đã rất quen thuộc với các ngành công nghiệp video game. Những người ủng hộ tài chính sẵn sàng để tài trợ cho các yêu cầu tiếp tục phân tích ở cấp khách hàng, và Hal và Karen chuẩn bị để đáp ứng với Howard. Sau này không thể cấp cho họ nhiều hơn nửa tiếng đồng hồ kể từ khi ông đã rất bận rộn. Karen đã phát triển một danh sách ưu tiên của các câu hỏi và các chủ đề và gửi chúng và dự thảo hiện có của SRS Chương 1 và 2 Howard. Họ lên kế hoạch để kết thúc yêu cầu cao cấp với Howard. Nhóm nghiên cứu cũng dự kiến quá trình phát triển các yêu cầu chi tiết.

5. Số liệu và kết thúc cho yêu cầu cấp cao

Các yêu cầu cấp cao đã phải chịu sự kiểm tra của toàn đội và các khuyết tật đã được ghi lại. Đối với các cuộc họp hàng tuần tiếp theo, Hal và Karen tóm tắt các số liệu như trong hình 12.37. Các nhóm đã đồng ý trên những quan sát sau khi chết được hiển thị.

6. Chuẩn bị yêu cầu chi tiết

Hal và Karen đã hoàn thành của họ viết lên những yêu cầu cao cấp, dựa trên các cuộc thảo luận và các cuộc phỏng vấn với Betty Sims và Arlan Howard. Họ đã sử dụng các tiêu chuẩn IEEE, có tiêu đề nhắc nhở họ về yêu cầu không có chức năng như GUls, hiệu suất, và các nền tảng phần cứng. Bây giờ họ đã phải xác định cách thức mà họ sẽ tổ chức các chi tiết yêu cầu chức năng. Họ dự đoán phải xem lại và cập nhật SRS nhiều lần, phối hợp các thiết kế và mã với nó: họ muốn quá trình này được đơn giản càng tốt. Kết quả là, tiêu chuẩn chính của họ là khả năng dễ dàng duy trì sự thống nhất giữa các SRS, thiết kế, và các mã.

Đầu tiên, họ đã thảo luận việc tổ chức các yêu cầu chi tiết của các quốc gia và hành động, dựa trên sơ đồ statetransition được mô tả trong các yêu cầu cao cấp. phương pháp tổ chức này sẽ bao gồm một danh sách các hành động mà một cầu thủ sẽ mất, chẳng hạn như bấm vào siêu liên kết lối ra vào một khu vực, tiếp theo là ảnh hưởng của hành động này. Cả hai đều đồng ý rằng đây sẽ là một tổ chức có thể hiểu được, nhưng quyết định rằng nó sẽ không theo dõi việc thực hiện cũng như họ muốn. Họ bắt đầu tìm kiếm những cách khác mà tổ chức yêu cầu chi tiết.

Hal là có lợi cho việc tổ chức các yêu cầu chi tiết chức năng của trường hợp sử dụng, đặc biệt là kể từ khi ông muốn theo Quy trình phát triển phần mềm thống nhất. Ông nói rằng, ở giai đoạn này, các trò chơi video có thể dễ dàng nhất được nghĩ về các trường hợp sử dụng thiết lập, việc di chuyển giữa các khu vực trò chơi sử dụng trường hợp, và tham gia các trường hợp sử dụng ký tự nước ngoài. Ông chỉ ra cách tiện lợi nó sẽ được sử dụng chỉ trong ba trường hợp sử dụng như tổng mức độ yêu cầu chức năng. Ông cũng là vui mừng về những triển vọng có lẽ là có thể sử dụng lại các trường hợp sử dụng để chỉ định các trò chơi trong tương lai.

Karen đã đồng ý rằng các yêu cầu sẽ là khá dễ hiểu nếu do trường hợp sử dụng tổ chức, nhưng cô đã có một số phản đối. Đầu tiên là một số yêu cầu sẽ là một phần của nhiều hơn một trường hợp sử dụng. Một ví dụ là những gì sẽ xảy ra khi một lối ra từ một căn phòng được nhấp. Đây có thể là một phần của tất cả các trường hợp sử dụng ba họ đã xác định, và vì vậy nó sẽ không được rõ ràng nơi để tìm nó. phản đối khác của Karen đã được thực tế rằng ánh xạ từ các trường hợp sử dụng để mã sẽ không được sạch sẽ như các tổ chức, cô đã có trong tâm trí. Cuối cùng, cô đã chỉ ra rằng tổ chức này vẫn chưa được trang bị để đúng kho lưu trữ trường hợp sử dụng để tái sử dụng trong tương lai.

Karen muốn tổ chức các trường hợp sử dụng chức năng của lớp, trong đó, cô cho biết, tạo điều kiện truy xuất nguồn gốc từ các yêu cầu để thực hiện. Cô muốn để chọn chúng một cách cẩn thận, đủ để đảm bảo rằng họ sẽ được sử dụng như là một phần của thiết kế (và thực hiện). Hal chỉ ra một bất lợi của phương pháp này: thực tế là nó buộc họ phải quyết định rất sớm trên một số các lớp học mà họ sẽ sử dụng trong việc thực hiện các ứng dụng. Ông lo lắng về khả năng sau này họ có thể thay đổi suy nghĩ của họ về việc lựa chọn. Sau khi thảo luận thêm, họ quyết định rằng tổ chức yêu cầu chi tiết của lớp có nhiều lợi ích hơn những hạn chế, và họ cam kết phương pháp này. Họ đã quyết định rất bảo thủ về việc lựa chọn lớp học.

7. Phân loại các yêu cầu chi tiết

Hal và Karen đầu tiên thử từng trường hợp sử dụng, và xác định những gì đối tượng của những lớp bắt đầu hành động và những đối tượng có trách nhiệm thực hiện các hành động. Quá trình này nhắc nhở chúng để tạo ra và / hoặc xác định các lớp. Họ tìm thấy nó cần thiết để gọi Betty và Arlan nhiều lần để làm rõ các bước trường hợp sử dụng họ nghĩ rằng họ đã hiểu nhưng thực sự thì không.

Hal liệt kê các lớp và các đối tượng được đề cập trong trường hợp sử dụng. Sau đó, họ động não, cọ rửa mọi khía cạnh của Encounter họ có thể hợp lý tưởng tượng cho các lớp học thêm có thể. Bước cuối cùng trong quá trình lựa chọn lớp, họ quyết liệt cắt giảm danh sách chỉ còn một vài điều cần thiết, nhưng để bảo vệ tất cả các lớp được đề cập trong các trường hợp sử dụng. Danh sách cuối cùng gồm Diện tích, EncounterCharacter, EncounterGame, Engagement, EngagementDisplay, ConnectionHyperlink, ForeignCharacter, PlayerCharacter, và PlayerQualityWindow. Bây giờ họ đã hoàn thành các đề mục của SRS tại mục 3.2 ( "các yêu cầu cụ thể"). Họ thu thập các yêu cầu chi tiết liên quan đến các lĩnh vực tại tiểu mục 3 .2.A, tương ứng với các lớp Area. Họ ra lệnh cho các tiểu mục theo thứ tự abc, vì họ dự đoán thêm các lớp sau. Họ phỏng đoán rằng nếu họ đã ra lệnh cho chủ đề theo số (ví dụ, PlayerCharacter là 3.2.14), sau đó định vị một yêu cầu cá nhân sẽ rất khó khăn hơn, bởi vì người sử dụng của SRS sẽ phải tìm kiếm nhiều 3 .2. N phần phụ trước khi tìm ra một áp dụng. Các khóa học sắp tới phúc EncounterAreaConnection, họ đánh số tiểu mục tiếp theo 3.2. EAC, và như vậy. Trong mỗi phân loại, họ đã tạo ra phần phụ cho các thuộc tính, các thực thể, junctionality, và các sự kiện.

8. Viết các yêu cầu chi tiết

Karen và Hal đã viết Phần 3.1 trên giao diện người dùng bằng cách điền vào các chi tiết trên bản phác thảo mà họ đã làm cho các yêu cầu Corporate, sau đó yêu cầu Betty, cũng như các bộ phận yếu tố con người, để xem xét chúng. Biết rằng đây sẽ là tài liệu chính thức mà từ đó chúng được kết xây dựng, họ đã có những khách hàng đồng thuận về mọi chi tiết.

Họ kiểm tra ghi chú cuộc phỏng vấn của họ với Betty và Arlan như các thuộc tính ( "thuộc tính") của mỗi phân loại (class). Ví dụ, họ hỏi những gì thuộc tính được yêu cầu cho các kết nối giữa hai khu vực Encounter. (Một tài sản của các kết nối như vậy là "khu vực đầu tiên" kết nối, và một người khác là "khu vực thứ hai.") Đối với mỗi lớp, họ hỏi mình những gì thực thể (instance của lớp) đã được yêu cầu cho các trò chơi. Ví dụ, có phải là một khu vực phòng thay đồ và khu vực sân. Sau đó họ hỏi những gì các chức năng lớp phải có. Ví dụ, một chức năng của từng nhân vật Encounter là khả năng cấu hình các giá trị của các phẩm chất của nó (yêu cầu 3.2.EC3.2). Cuối cùng, họ liệt kê tất cả các sự kiện thể hiện của lớp đã được yêu cầu để đáp ứng (ví dụ, nhấp vào một lối ra từ một khu vực).

Một khía cạnh làm khó họ là thời gian cần thiết cho các giá trị mới có hiệu lực. Họ nhận ra rằng đây là một khía cạnh quan trọng để các trò chơi: nếu không có thời gian đã trôi qua, người chơi sẽ chỉ đơn giản là thiết lập những phẩm chất liên quan đến diện tích hiện tại đến tối đa, và ít kỹ năng sẽ được yêu cầu để chơi các trò chơi. Việc chậm trễ thực hiện các trò chơi thú vị, nhưng vấn đề là, làm thế nào nhiều lần trì hoãn nên có? Họ coi đó nêu "được quyết định" trong thời gian, nhưng cuối cùng kết luận rằng điều này sẽ không giúp đỡ. Họ quyết định để xác định bốn giây, cảm thấy rằng việc thay đổi số tiền này nên được đơn giản.

Karen lo ngại về sự thiếu chính xác của một số các yêu cầu, đặc biệt là những người có liên quan đến cách thức mà điểm chất lượng cần được trao đổi khi hai nhân vật tham gia mỗi khác. Cô cảm thấy rằng các lập trình viên có thể dễ dàng hiểu sai yêu cầu. Điều này sẽ lãng phí thời gian vào các khuyết tật và tạo ra một trò chơi bị lỗi. Cô đề nghị sử dụng Z-thông số kỹ thuật. Hal đưa ra luận điểm rằng không có ai ngoại trừ Karen sẽ hiểu họ cũng đủ, kể từ khi phần còn lại của nhóm đã không có sự giáo dục cần thiết. Họ thỏa hiệp bằng cách đồng ý để sử dụng toán học thích hợp trong xác định yêu cầu này, nhưng không phải là định dạng Z-đặc điểm kỹ thuật. Karen đã chú ý rằng nếu cô ấy đã từng dạy kỹ thuật phần mềm, cô sẽ nhấn mạnh rằng tất cả các sinh viên phải hoàn toàn thoải mái với Z-thông số kỹ thuật.

Được thúc đẩy bởi các tiêu đề mục trong các tiêu chuẩn IEEE SRS, Karen và Hal làm chắc chắn để trang trải tất cả các yêu cầu thực hiện và kiểm tra chúng với Betty và Arlan, chủ yếu liên quan đến tốc độ mà các trò chơi sẽ có để có được thú vị. Họ cũng nghĩ thông qua các yêu cầu bộ nhớ (RAM và đĩa). Sau đó, họ đã hoàn thành tài liệu.

9. Following up: Metrics và kết thúc của yêu cầu chi tiết

Nhóm nghiên cứu phân tích yêu cầu hỏi Betty, Arlan, và phần còn lại của đội để kiểm tra các yêu cầu chi tiết. Họ thực hiện kiểm tra này chủ yếu đối với các yêu cầu cao cấp bằng cách đảm bảo rằng tất cả các phần của các yêu cầu cao cấp được xây dựng trên bằng các yêu cầu chi tiết. Họ cũng sử dụng một danh sách kiểm tra như một trong những mô tả trong Bảng 13.3 của Chương đã được tìm thấy 13. Một số khiếm khuyết, mà Hal và Karen ghi nhận và sửa chữa. Kết quả của quá trình này là tương tự như mô tả trong hình 12.37.

# 12.14 Trường hợp nghiên cứu: Các yêu cầu chi tiết cho trò chơi đối kháng.

Phần này hoàn thành các yêu cầu đặc điểm kỹ thuật của các trò chơi video ở định dạng Encounter IEEE.

3. Yêu cầu chi tiết

Lưu ý cho các học sinh: Các hạn IEEE sử dụng trong nhóm này là yêu cầu "cụ thể". Chúng tôi đã thay thế bằng chữ "chi tiết" để phù hợp với các văn bản.

3.1 Các yêu cầu Giao diện bên ngoài

3. 1. 1 giao diện người dùng

Phần 2.1.2 trong SRS cho các trò chơi Encounter video cho thấy chỉ có bản phác thảo của giao diện người dùng để cung cấp quan điểm của sản phẩm. Nó thiếu detai Is và không nên được coi là tiếng nói cuối cùng. Nếu giao diện người dùng không phải là hoàn toàn cụ thể · fied sau của tài liệu này, sau đó tất cả các chi tiết cần được đưa ra trong phần này. Vì chúng ta đang sử dụng các đối tượng phong cách của đặc điểm kỹ thuật trong nghiên cứu trường hợp này, các chi tiết của mỗi cửa sổ được đóng gói với các lớp học của họ tại mục 3.2.2 trong SRS. Trong mọi trường hợp, phần này nên giải thích mối quan hệ vật lý giữa các yếu tố đồ họa (ví dụ, tầng, lát gạch, chồng).

Gặp phải diễn ra trong khu vực. Hình 12,38 cho thấy một ảnh chụp màn hình tiêu biểu của khu vực sân, với một nhân vật người chơi điều khiển một nhân vật ngoài, và kết quả của một sự nhập vai. Giao diện này chiếm toàn bộ màn hình. Khu vực có các kết nối đến các khu vực lân cận, dán nhãn của các siêu liên kết. Nhấp vào một trong các liên kết di chuyển nhân vật của người chơi vào khu vực tương ứng. Toàn bộ các giao diện như sau:

a. Một giao diện người dùng cho từng khu vực, quy định tại Mục 3.2AR dưới đây.

b. Một giao diện người dùng để thiết lập các giá trị chất lượng của các nhân vật của người chơi, quy định tại Mục 3.2.PQ.

c. Một giao diện người dùng để hiển thị các kết quả của một sự dấn thân, quy định tại Mục 3.2.ED. Giao diện người dùng tương tự được sử dụng để hiển thị trạng thái của nhân vật của người chơi.Một giao diện của loại một trên sẽ luôn luôn có mặt trên màn hình. Khi kêu gọi của các yêu cầu này, giao diện của các loại b hoặc c sẽ được chồng. Yêu cầu này được kiểm tra trong tài liệu phần mềm thử nghiệm (STD). <Tham khảo kiểm tra tại đây>.

3.1.2 Giao diện phần cứng

Các phần cứng mà Encounter (đó là một phần mềm ứng dụng) đềKhông (Trong một tương lai phát hành, Encounter sẽ được kiểm soát bởi một phím điều khiển.)

3.1.3 Phần mềm giao diện

Phần mềm khác mà gặp phải giao diện: một ví dụ sẽ là một Không điều khiển máy in (Trong một thông cáo ufture, Encounter sẽ được chơi từ nItergalactic Internet Gaming Site.)

3.1.4 Giao diện truyền thôngKhông (Trong một tương lai phát hành, gặp phải giao tiếp với Internet thông qua một modem có ít nhất 56 Kb / s.).

3.2 Các yêu cầu chi tiết của loại

IEEE sử dụng tiêu đề "Các lớp / đối tượng" cho mục 3.2. Điều này giả định một khán giả mà biết định hướng đối tượng. Nó là cần thiết để hiểu 00 để tạo ra phần này nhưng nó không phải là cần thiết để đọc và hiểu nó.

Kể từ khi chúng tôi đang phân loại các yêu cầu chi tiết của lớp, đầu tiên chúng tôi danh sách các lớp học mà chúng ta có v (ery cẩn thận!) Chọn. Đây không phải là tất cả các lớp học sẽ được sử dụng bởi các ứng dụng-chỉ đơn thuần là các lớp cốt lõi liên quan đến các lĩnh vực ứng dụng, mà là đầy đủ cho tổ chức tất cả các yêu cầu. Trong trường hợp này, ví dụ, tất cả trong số họ là những khía cạnh của trò chơi Encounter video.

Danh mục cho các trò chơi video Encounter đủ để thể hiện yêu cầu này là Diện tích, EncounterCharacter, EncounterGame, Engagement, EngagementDisplay, ForeignCharacter, PlayerCharacter, và PlayerQualityWnidow.

Việc đánh số "3.2.Area.N.N ...", vv được sử dụng tại mục 3.2 làm cho nó dễ dàng hơn cho chúng tôi để chèn, xóa bỏ, và xác định vị trí các yêu cầu của tổ chức theo thứ tự abc các lớp có chứa chúng. Hãy suy nghĩ về hàng trăm yêu cầu. Nếu chúng ta đánh số các lớp học sử dụng "3.2.1 ...", "3.2.2 ...", vv, sau đó chèn các lớp học mới sẽ phải được thực hiện vào cuối danh sách, vì số hiện có, đã gọi ở nơi khác trong dự án, không thể bị quấy rầy. Các yêu cầu sẽ không được đặt theo thứ tự abc. Kết quả là, người ta sẽ phải đi qua các yêu cầu từng người một để xác định vị trí một cụm.

3.2. Khu vực AR

Đầu tiên, chúng tôi mô tả những gì lớp tôi (.e., Phân loại này yêu cầu) đề cập đến.Một khu vực là một nơi có thể xem trên màn hình. Tất cả các hoạt động của Encounter i (cam ncluding) diễn ra trong khu vực. Phòng, vườn và sân là những ví dụ của khu vực.

3.2.AR.1 thuộc tính của khu vực

Ở đây chúng ta nói với những thuộc tính gì từng đối tượng s (pecific thực thể) của lớp phải có.

3.2.AR.1.1 Khu vực (tên cần thiết; chưa thực hiện)

Những tuyên bố trên trong ngoặc chỉ ra các ưu tiên và trạng thái của yêu cầu. Một khi các yêu cầu được mã hoá và thử nghiệm, các tuyên bố "chưa thực hiện" hoặc là xóa hoặc thay đổi để "thực hiện". "" Yêu cầu cơ bản được thực hiện đầu tiên. Khi một yêu cầu đã được thiết kế cho và thực hiện, "thiết yếu" có thể được loại bỏ. Đây là một kỹ thuật để theo dõi trạng thái của ứng dụng và mối quan hệ của nó với SRS này. Một kỹ thuật khác là để xác định lặp đi lặp lại mà yêu cầu áp dụng. Mỗi khu vực sẽ gặp phải một tên duy nhất bao gồm 1-15 ký tự. nhân vật chấp nhận được sẽ bao gồm khoảng trống, 0 đến 9, a đến z, và A chỉ đến Z.kế hoạch kiểm tra <tham khảo để kiểm tra tại đây>.Mỗi yêu cầu thuộc tính kiểu bản đồ đến một cặp chức năng GET và set-. Tài liệu này cho thấy làm thế nào mỗi yêu cầu có thể được siêu liên kết đến một bài kiểm tra đơn vị trong Tài liệu phần mềm thử nghiệm.

3.2.AR.1.2 Diện tích hình ảnh (rất cần thiết; chưa thực hiện)

Sẽ có một hình ảnh để hiển thị từng đối tượng Diện tích trên toàn bộ màn hình. Các hình ảnh sẽ lấp đầy toàn bộ màn hình.

3.2.AR.1.3 khu vực chất lượng đặc biệt (điều cần thiết; chưa thực hiện)

Chỉ có một số phẩm chất nhân vật game được áp dụng trong từng khu vực. Những phẩm chất cụ thể đối với từng khu vực được quy định tại Mục 3.2.AR.2.

Các đối tượng 3.2.AR.2 Diện tích

Chúng tôi chỉ định đối tượng khu vực cụ thể mà phải tồn tại trong ứng dụng. Phần này đã được thêm vào các tiêu chuẩn IEEE. Các thay thế sẽ có được để thể hiện các yêu cầu như các chức năng: ". Encounter phải có khả năng hiển thị khu vực XYZ với các đặc điểm sau" chẳng hạn,

Diện tích 3.2.AR.2.1 Courtyard (cần thiết; chưa thực hiện)

Có phải là một đối tượng Area có tên "sân" đòi hỏi những phẩm chất thể lực và sức mạnh. Những hình ảnh sân sơ bộ thể hiện trong hình 12,39 bao gồm một bản đồ của khu vực lân cận.

Diện tích 3.2.AR.2.2 Dressing Room (cần thiết; chưa thực hiện)

Sẽ có một khu vực có tên "phòng thay đồ" không cần chất lượng. hình ảnh sơ bộ của mình, thể hiện trong hình 12,40, bao gồm một bản đồ của khu vực lân cận.

3.2.AR.2.3 khu vực hang động (cần thiết; chưa thực hiện)

Sẽ có một khu vực có tên "dungeon" đòi hỏi những phẩm chất sức chịu đựng và kiên nhẫn. hình ảnh sơ bộ của nó thể hiện trong hình 12,41 bao gồm một bản đồ của khu vực lân cận.

Diện tích 3.2.AR.2.4 bếp (cần thiết; chưa thực hiện)

Sẽ có một khu vực có tên "nhà bếp" đòi hỏi sự tập trung chất lượng. Hình ảnh bếp sơ bộ thể hiện trong hình 12 .42 bao gồm một bản đồ của khu vực lân cận.3.2.AR.2.5 Diện tích phòng khách

(thiết yếu; chưa thực hiện) Sẽ có một khu vực có tên "phòng khách" đòi hỏi sự tập trung chất lượng và sức chịu đựng. hình ảnh sơ bộ của nó thể hiện trong hình 12.43 bao gồm một bản đồ của khu vực lân cận.

3.2.AR.4 Sự kiện Liên quan đến khu vực

Chúng tôi tách các sự kiện liên quan đến các lĩnh vực từ các thuộc tính, đối tượng và phương pháp. Một sự kiện là một hành động xảy ra cho các ứng dụng và xúi giục từ bên ngoài của ứng dụng.

3.2.AR.4.1 Display nhập cảnh của các cầu thủ Character (cần thiết; chưa thực hiện) Bất cứ khi nào nhân vật chính của người chơi tiến vào khu vực, khu vực đó và các nhân vật trong nó sẽ được hiển thị trên màn hình, làm đầy màn hình.

3.2.AR.4.2 những cam Xử lý (cần thiết; chưa thực hiện) Khi một nhân vật game nước ngoài tiến vào khu vực chứa vật chính của người chơi, hoặc ngược lại, họ tham gia mỗi khác.

.--- Phòng bếp

Sân

phòng khách

Phòng thay đồ

Dungeon Nghiên cứu

3.2.AR.4.3 làm gián đoạn Engagements (tùy chọn, chưa thực hiện)

Người chơi có thể làm gián đoạn cam kết một cách ngẫu nhiên. Tính trung bình, người chơi có thể ngăn chặn một trong mười cam kết bằng cách thực hiện các thủ tục để thiết lập chất lượng. Người dùng cố gắng để làm gián đoạn một cam kết bằng cách cố gắng để thiết lập những phẩm chất của người chơi. Nếu trò chơi không cho phép điều này, không có dấu hiệu được cho: các trò chơi tiến hành nếu như các nỗ lực đã không được thực hiện.

3.2.AR.4.4 Nhấn cài đặt nút bấm (thiết yếu; chưa thực hiện)

Khi người dùng nhấn nút Set, một cửa sổ để thiết lập các giá trị của chất xuất hiện chồng lên khu vực, với điều kiện là không có nhân vật nước ngoài trong khu vực . Xem 3.2.PQ cho các chi tiết kỹ thuật của cửa sổ này.

3.2.AR.4.S Nhấn nút End game (tùy chọn, chưa thực hiện)

Khi người dùng nhấn nút End game, trò chơi kết thúc. Không có màn hình bổ sung xuất hiện.

Câu trước đó, một nghịch đảo yêu cầu · ment, được cảm là cần thiết bởi vì trò chơi thường làm hiển thị một bản tóm tắt của một phiên.

3.2.AR.4.6 Nhấn nút Get (tùy chọn, chưa thực hiện)

Khi người dùng nhấn vào nút trạng thái Get, một cửa sổ hiển thị cam kết sẽ xuất hiện cho tình trạng của các nhân vật của người chơi trước và sau khi tham gia cuối cùng.

3.2.CH kết nối siêu liên kết giữa khu vực siêu liên kết kết nối những siêu liên kết đặt tại mỗi lối ra khu vực, cho thấy khu vực mà nó được kết nối.

3.2.CH.1 thuộc tính của kết nối Hyperlinks

3.2.CH.1.1 kết nối (thiết yếu; chưa thực hiện) Mỗi siêu liên kết kết nối tương ứng với một kết nối khu vực.

3.2.CH.2 Các đối tượng kết nối Hyperlink (cần thiết; chưa thực hiện) Có hai siêu liên kết kết nối tương ứng với mỗi kết nối khu vực, một ở từng khu vực của kết nối.

3.2.CH.3 Chức năng Connection Hyperlinks Không

3.2.CH.4 Sự kiện Liên quan đến kết nối Hyperlinks

3.2.CH.4.1 tài Nhấp chuột vào một kết nối Hyperlink Hiệu quả của cách nhấn vào một liên kết kết nối là nhân vật của người chơi sẽ được hiển thị trong khu vực ở phía bên kia của kết nối khu vực.

3.2.CO kết nối giữa khu vực Nhân vật di chuyển từ vùng này sang vùng lân cận bằng các phương tiện của các kết nối. Mỗi một kết nối hai khu vực. Hình 12.45 cho thấy các kết nối cần thiết giữa các khu vực.

3.2.CO.1 thuộc tính của kết nối giữa khu vực

3.2.CO.1.1 Đầu tiên và khu vực II (cần thiết; chưa thực hiện)

Mỗi kết nối được kết nối với cặp khu vực, chúng tôi sẽ gọi "đầu tiên" và khu vực "thứ hai".

3.2.CO.2 Connections Entities

3.2.CO.2.1 Dressing Room-Courtyard (cần thiết; chưa thực hiện)

Sẽ có một kết nối giữa các phòng thay đồ và sân.

3.2.CO.2.2 Dungeon-Study (cần thiết; chưa thực hiện)

Sẽ có một kết nối giữa các dungeon và nghiên cứu.

3.2.CO.2.3 Nghiên cứu-Living Room (cần thiết; chưa thực hiện)

Sẽ có một kết nối giữa nghiên cứu và phòng khách.

3.2.CO.2.4 Courtyard-Living Room (cần thiết; chưa thực hiện)

Sẽ có một kết nối giữa sân và phòng khách.

3.2.CO.2.5 Dressing Room-Dungeon (cần thiết; chưa thực hiện)

Sẽ có một kết nối giữa các phòng thay đồ và các dungeon.

3.2.CO.2.6 Courtyard-bếp (cần thiết; chưa thực hiện)

Sẽ có một kết nối giữa sân và nhà bếp.

3.2.CO.3 chức năng của Khu Connections

Không

3.2.CO.4 Sự kiện liên quan đến kết nối Diện tích

3.2.CO.4.1 Di chuyển một nhân vật thông qua một kết nối (thiết yếu; chưa thực hiện) Kết nối được hiển thị như là các siêu liên kết tại các biên giới khu vực mỗi khi nhân vật của người chơi là trong khu vực. Khi người dùng nhấp vào một liên kết như vậy, vùng liên kết được hiển thị với các nhân vật trong lĩnh vực này.

3.2.EC Nhân vật Encounter

3.2.EC.1 thuộc tính của nhân vật Encounter

3.2.EC.1.1 Tên của nhân vật Encounter (cần thiết; chưa thực hiện)

Mỗi nhân vật trò chơi trong các trò chơi video Encounter có một tên duy nhất. Các thông số kỹ thuật cho tên phải giống như đối với tên Diện tích, quy định tại 3.2. AR.l.

3.2.EC.1.2 Chất lượng của nhân vật Encounter (cần thiết; chưa thực hiện)

Mỗi nhân vật game có cùng một phẩm chất. Mỗi chất lượng phải là một số điểm nổi không âm có ít nhất một chữ số thập phân của độ chính xác. Đây là tất cả đều khởi tạo để tổng hợp các giá trị của họ là 100. Giá trị của chất lượng không thể được cả hai lớn hơn a và nhỏ hơn 0,5. Đối với phiên bản đầu tiên những phẩm chất được tập trung, trí thông minh, sự kiên nhẫn, sức chịu đựng và sức mạnh.

3.2.EC.1.2 Ảnh của vật Encounter (cần thiết; chưa thực hiện)

Mỗi nhân vật game sẽ có một hình ảnh.

3.2.EC.2 Encounter thực thể vật Các nhân vật của trò chơi được mô tả trong các loại ký tự Encounter.

3.2.EC.3 Chức năng của nhân vật Encounter

3.2.EC.3.1 điểm sinh hoạt chung (cần thiết; chưa thực hiện)

Các trò chơi gặp phải có thể sản xuất tổng của các giá trị của phẩm chất nào của nhân vật, được gọi là điểm sống của mình.

3.2.EC.3.2 năng cấu hình các giá trị chất lượng Encounter Character (cần thiết; chưa thực hiện)

Bất cứ khi nào một nhân vật Encounter là một mình trong một khu vực, giá trị của bất kỳ phẩm chất của nó có thể được thiết lập. Các giá trị được chọn phải nhỏ hơn hoặc bằng tổng của các giá trị chất lượng. Các giá trị của những phẩm chất còn lại sẽ được tự động điều chỉnh để duy trì tỷ lệ chung của mình, ngoại trừ kết quả với số lượng ít hơn một, mà được thay thế bằng giá trị chất lượng của số không.

3.2.ED Engagement Hiển thị (thiết yếu; chưa thực hiện)

Sẽ có một cửa sổ hiển thị các kết quả của cam kết. Các định dạng được thể hiện trong hình 12.46.

3.2.ED.4 Engagement hiển thị sự kiện

3.2.ED.4.1 Không chấp nhận hiển thị (thiết yếu; chưa thực hiện)

Khi người dùng nhấn OK, màn hình biến mất.

3.2.EG Các Encounter game Các yêu cầu trong phần này liên quan đến các trò chơi như một toàn thể.

3.2.EG.1 Thuộc tính của Encounter game

3.2.EG.1.1 Thời gian (tùy chọn, chưa thực hiện)

Một hồ sơ sẽ được lưu lại trong suốt thời gian của mỗi trò chơi, tính thời gian từ khi người chơi bắt đầu trò chơi.

3.2.EG.2 Các đối tượng của trò chơi Encounter

3.2.EG.2.1 Độc Game (cần thiết; chưa thực hiện)

Có phải là một trò chơi duy nhất.

phiên bản tương lai sẽ cho phép một số phiên bản của trò chơi để chạy cùng một lúc.

3.2.EN cam Một cam kết là sự tương tác giữa một nhân vật trò chơi điều khiển bởi người chơi và một nhân vật ngoài.

3.2.EN.1 thuộc tính của những cam None

3.2.EN.2 Engagement Các đối tượng không có thực thể tham gia vĩnh viễn.

3.2.EN.3 Chức năng của những cam

3.2.EN.3.1 Tham một nhân vật ngoài (chủ yếu; chưa thực hiện)

yêu cầu đặc biệt này là toán học trong tự nhiên và vì vậy không có nỗ lực để thay thế cho toán học với ngôn ngữ tự nhiên, trong đó sẽ có nguy cơ ảnh hưởng đến độ chính xác của nó. Việc sử dụng ngôn ngữ tự nhiên để giải thích toán học là một thực hành tốt, tuy nhiên.

Khi tương tác diễn ra, các "mạnh mẽ" của hai nhân vật là một trong những giá trị của chất areaspecific tổng hợp với số lượng lớn hơn. Việc chuyển hệ thống một nửa giá trị của mỗi chất lượng khu vực cụ thể của các yếu đến mạnh. Không chuyển điểm diễn ra nếu không phải nhân vật mạnh hơn. Nếu một trong hai nhân vật không có điểm sau khi tái phân bổ giá trị được thực hiện, trò chơi kết thúc. Nếu trò chơi không kết thúc, nhân vật của người chơi là di chuyển đến một khu vực ngẫu nhiên và kết quả của sự tương tác hiển thị. Như một ví dụ về việc phân bổ lại giá trị, cho rằng các cầu thủ tham gia vào một nhân vật nước ngoài trong một khu vực thích khả năng chịu đựng và sự tập trung. Nếu Ps là giá trị của sức chịu đựng của người chơi, và giả sử Ps + Pc> Là + Trong chúng ta sẽ có Ps '= Ps + 1, / 2, P: = Pc + Ii2,} s' = 1, / 2, Andi: = fa / 2.

Người đọc sẽ nhận ra những khiếm khuyết trong phương trình cuối cùng, mà phải là J: = 1/2. Chúng tôi sẽ để lại những khiếm khuyết còn nguyên vẹn như một ví dụ.

Để lấy một ví dụ số của một sự dấn thân trong lĩnh vực này: Nếu giá trị sức chịu đựng của người chơi là 7 và giá trị tập trung là 19, và Freddie sức chịu đựng của người nước ngoài là 11 và nồng độ 0,6, sau đó người chơi mạnh hơn. Kết quả của sự tham gia sẽ là:

Player: khả năng chịu đựng 7 + 11/2 = 12,5; Nồng độ 19 + (0,6) 12 = 19,3 [OJ Freddie: sức chịu đựng

11/2 = 5,5; Nồng độ 0 vì (0,6) / 2 là ít hơn 0,5

3.2.FC nhân vật nước ngoài A ký tự nước ngoài là một nhân vật gặp không thuộc quyền kiểm soát của người chơi.

3.2.FC.1 thuộc tính của nhân vật nước ngoài Xem các yêu cầu nhân vật Encounter. Đây được khởi tạo để được bình đẳng.

Trong phiên bản tương lai, nhân vật nước ngoài có thể đột biến thành hình thức mới.

3.2.FC.2 Các đối tượng nhân vật nước ngoài

Phần này cho biết rằng chỉ có một nhân vật nước ngoài.

3.2.FC.2\_1 Freddie nhân vật ngoài (chủ yếu; chưa thực hiện)

Sẽ có một nhân vật nước ngoài có tên là "Freddie", mà hình ảnh được hiển thị trong hình 12.47. Nhân vật này ban đầu sẽ có tổng cộng 100 điểm được phân bố đều cho chất lượng của nó.

3.2.FC.3 Chức năng của nhân vật ngoài

3.2.FC.3.1 Nhân vật bên ngoài (cần thiết; chưa thực hiện)

Miễn là nó còn sống, một nhân vật nước ngoài nên di chuyển từ vùng này sang vùng lân cận tại các khoảng thời gian ngẫu nhiên trung bình hai giây. Sau khi được hiện diện trong một khu vực cho một ngẫu nhiên lượng thời gian trung bình từ một thứ hai, tất cả các điểm cuộc sống của nhân vật được phân chia giữa các chất có liên quan đến khu vực, như vậy mà giá trị của mỗi chất lượng như gần bằng nhau càng tốt.

3.2.PC Player Characters Đây là những nhân vật Encounter dưới sự kiểm soát của các người chơi.

3.2.PC.1 Thuộc tính của Player Characters Xem thuộc tính nhân vật Encounter. hình ảnh nhân vật người chơi có thể lựa chọn từ một trong những hình ảnh trong hình 12,48.

Các đối tượng nhân vật 3.2.PC.2 Chơi

Nhân vật chính 3.2.PC.2.1 cầu thủ Người chơi sẽ có thể kiểm soát một nhân vật trò chơi đặc biệt

gọi là "chính" nhân vật. Bản chất của kiểm soát này là tùy thuộc vào các hạn chế quy định tại các yêu cầu còn lại. Nhân vật này ban đầu sẽ có tổng cộng 100 điểm được phân bố đều cho chất lượng của nó.

3.2.PC.2.2 nhân vật khác dưới sự kiểm soát của các Player (tùy chọn, chưa thực hiện)

Người chơi sẽ có thể giới thiệu các nhân vật khác hơn so với các nhân vật chính mà các điều khiển máy nghe nhạc. Thông tin chi tiết sẽ được quyết định.

3.2.PC.3 chơi nhân vật chức năng

3.2.PC.3.1 năng cấu hình của các giá trị chất lượng chơi Character (cần thiết; chưa thực hiện)

Bất cứ khi nào tất cả các cầu thủ nước ngoài vắng mặt tại khu vực chứa vật chính của người chơi, người chơi có thể thiết lập giá trị của bất kỳ chất lượng của các nhân vật chính sử dụng Player cửa sổ chất lượng hiển thị trong hình 12,49. Các giá trị được chọn phải nhỏ hơn hoặc bằng tổng của các giá trị chất lượng. Các giá trị của những phẩm chất còn lại sẽ được tự động điều chỉnh để duy trì tỷ lệ chung của mình, ngoại trừ kết quả với số lượng nhỏ hơn 0,5, mà được thay thế bằng giá trị chất lượng của số không.

3.2.PC.3.2 năng cấu hình của Player Character hình ảnh (mong muốn; chưa thực hiện)

Người chơi sẽ có tùy chọn để chọn hình ảnh đại diện cho nhân vật chính của mình từ ít nhất hai hình ảnh. Các tùy chọn được hiển thị trong hình 12,48.

3.2.PC.3.3 lão hóa của Player Character hình ảnh (tùy chọn, chưa thực hiện)

Các nhân vật người chơi chính sẽ tự động tăng mỗi chất lượng bằng một tỷ lệ phần trăm trong nửa đầu của cuộc sống của mình, sau đó giảm dần mỗi chất lượng bằng tỷ lệ phần trăm tương tự cho các một nửa thứ hai. Thông tin chi tiết sẽ được quyết định.

3.2. PQ The Player Chất lượng cửa sổ này là một cửa sổ từ đó người chơi có thể phân phối các giá trị của các nhân vật của mình.

3.2.PQ.1 thuộc tính của chất lượng Window Player cửa sổ để thiết lập những phẩm chất của một nhân vật trong Encounter được thể hiện bằng phương tiện của một ví dụ điển hình trong hình 12 .49. Các biểu tượng nhân vật game xuất hiện ở trung tâm, và tên của nó xuất hiện ở phía trên bên trái của màn hình. điểm cuộc sống của nhân vật này xuất hiện ở trung tâm. Trên trung tâm bên trái là một hộp danh sáchhiển thị bốn phẩm chất tại một thời điểm. Nhấp vào một trong những phẩm chất này cho phép người chơi phải chọn một giá trị cho nó vào hộp văn bản bên phải. Giải thích về cách tính toán số học được thực hiện được thể hiện trong một hộp màu vàng nhạt ở phần dưới của màn hình.

3.2.PQ.2 chơi chất lượng Window Entity

3.2.PQ.2.1 Window Giao Phẩm chất (thiết yếu; chưa thực hiện)

Một cửa sổ sẽ được áp dụng theo các điều kiện được mô tả ở trên để phân bổ các giá trị của các nhân vật người chơi. Cửa sổ sẽ có sự xuất hiện của CUI thể hiện trong mục 3.1.1.2- các đặc điểm kỹ thuật này.

3.2.PQ.3 Chơi Chất lượng chức năng

3.2.PQ.3.1 Khởi tạo Display (cần thiết; chưa thực hiện)

Cầu thủ menu chất lượng sẽ có thể hiển thị bản thân.

3.2.PQ.4 chơi chất lượng Window Sự kiện

3.2.PQ.4.1 Hiển thị giá trị của một chất lượng (cần thiết; chưa thực hiện)

Khi nhấp chuột chơi trên một chất lượng trong hộp danh sách bên trái, giá trị của chất lượng đó sẽ được hiển thị trong hộp văn bản bên phải.

3.2.PQ.4.2 Thiết lập các giá trị của một chất lượng (cần thiết; chưa thực hiện)

Khi người dùng nhập vào một giá trị hợp lệ cho một chất lượng và truy cập vào nút "enter", giá trị của chất lượng được đặt thành số lượng nhập vào. Nếu giá trị là không hợp lệ, một cửa sổ báo lỗi sẽ xuất hiện thông báo "giá trị không hợp lệ:. Thử lại"

3.2.PQ.4.3 Bác bỏ các cửa sổ (cần thiết; chưa thực hiện)

Khi người dùng nhấn vào nút OK, một thời gian bốn giây trôi qua, sau đó cửa sổ biến mất. Vào cuối của khoảng thời gian này (tức là, nếu không có sự gián đoạn) phân bổ giá trị được thực hiện.

3.2.PQ.4.4 gián đoạn (thiết yếu; chưa thực hiện)

Sau khi gián đoạn của màn hình hiển thị của cửa sổ giá trị chất lượng, cửa sổ biến mất. Lưu ý rằng sự gián đoạn sẽ được gây ra bởi một nhân vật nước ngoài vào khu vực. Cũng lưu ý trong trường hợp này là các giá trị chất lượng không thay đổi và tương tác diễn ra.

3.3 Yêu cầu Hiệu suất

yêu cầu thực hiện bao gồm tốc độ và / hoặc thời gian để hoàn thành yêu cầu. Trừ khi tài liệu trong một phần khác nhau của SRS, họ cũng có thể bao gồm việc sử dụng bộ nhớ (RAM và / hoặc đĩa), ghi nhận một trong hai tĩnh hoặc động (ví dụ, bộ nhớ cần thiết trong thời gian chạy).

Các ứng dụng sẽ tải và hiển thị hình ảnh ban đầu trong ít hơn một phút. Cam kết sẽ thực hiện trong vòng chưa đầy một giây. Những yêu cầu này sẽ được kiểm tra STO <tham khảo để kiểm tra tại đây>.

3.4 Các ràng buộc thiết kế

Phần này quy định hạn chế về thiết kế. Nếu không có các tài liệu trong phần này, các nhà thiết kế có thể tự do tạo ra bất kỳ thiết kế (tốt) thỏa mãn các yêu cầu. Ví dụ, chúng ta có thể thêm các thiết kế chế "một câu chuyện" như sau: "Một căn nhà với bốn phòng ngủ, tất cả trong số đó là ít hơn 1/32 đi bộ từ phòng gia đình."

Encounter được thiết kế sử dụng UML và thiết kế hướng đối tượng. Nó được thực hiện trong Java. Phần mềm này sẽ chạy như một ứng dụng Java trên Windows 95. Nó được thiết kế theo một cách mà làm cho nó tương đối dễ dàng để thay đổi các quy tắc theo đó các hoạt động trò chơi để những người khác có thể tùy chỉnh các trò chơi.

3.5 Thuộc tính Hệ thống phần mềm

3.5. 1 Độ bền Encounter sẽ không thiếu thốn nhiều hơn một lần trong mỗi 1.000 cuộc gặp gỡ. Tài liệu kiểm thử <tham khảo để kiểm tra tại đây>.

3.5.2 Sẵn gặp phải có sẵn để chơi trên bất kỳ máy tính chạy Windows 95 chỉ (tức là, không có ứng dụng khác cùng một lúc). Tài liệu kiểm thử <tham khảo để kiểm tra tại đây>.

3.5.3 An ninh

phiên bản tương lai sẽ cho phép truy cập vào trò chơi lưu chỉ với một mật khẩu.

3.5.4 Bảo trì

3.5.4.1 Nhân vật thay đổi và khu vực (cần thiết)

Nó phải đơn giản để thay đổi nhân vật và các khu vực.

3.5.4.2 Styles toàn cầu Thay đổi (mong muốn)

Nó sẽ được đơn giản để thay đổi trên toàn cầu theo phong cách của các khu vực và các kết nối. (Thay đổi phong cách phản ánh mức độ khác nhau của trò chơi chơi trong cùng một môi trường.)

3.5.4.3 Rules of Engagement Thay đổi nội quy (tùy chọn)

Sự gắn kết cần được dễ dàng thay đổi.

3.6 Các yêu cầu khác

không ai

4. Thông tin hỗ trợ

không ai

4.1 Mục lục và chỉ số

Không bao gồm.

4.2 Phụ lục

Không bao gồm.

Phụ lục có thể bao gồm

(A) Mẫu I / định dạng O, giới thiệu các nghiên cứu phân tích chi phí, hoặc kết quả của các cuộc khảo sát người dùng (b) Hỗ trợ hoặc thông tin cơ bản có thể giúp các độc giả của SRS (c) Mô tả các vấn đề cần được giải quyết bằng các phần mềm ( d) hướng dẫn đóng gói đặc biệt cho các mã và các phương tiện truyền thông để đáp ứng an ninh, xuất khẩu, nạp ban đầu, hoặc các yêu cầu khác của Nhà nước một cách rõ ràng hay không mỗi phụ lục là trở thành một phần chính thức của SRS.

# 12.15 Tổng kết

Yêu cầu chi tiết (yêu cầu "chi tiết" hay yêu cầu của người lập trình) được viết chủ yếu bởi quan diểm của những nhà thiết kế hay những nhà phát triển. Chúng được tạo ra từ những yêu cầu có trình độ cấp cao, cũng như từ sự tương tác liên tục với khách hàng. Các yêu cầu chi tiết phải được kiểm chứng, theo dõi, và nhất quán với nhau. Khi chúng có số lượng nhiều thì chúng phải được phân loại một cách hệ thống. Có một số cách để tổ chức yêu cầu chi tiết theo tính năng, trường hợp sử dụng, CUI, trạng thái, và lớp miền.

Dự án Agile có xu hướng không tạo ra các tài liệu riêng biệt từ code và thử nghiệm với các yêu cầu chi tiết.

Các yêu cầu chi tiết phải được theo dõi để thiết kế và thực hiện để xác nhận giá trị của chúng. Nếu không có sự theo dõi rõ ràng từng yêu cầu thông qua các thiết kế của ứng dụng để thực hiện, nó sẽ rất khó để đảm bảo rằng một ứng dụng như vậy vẫn còn phù hợp với các yêu cầu của nó. Khi thay đổi các yêu cầu, đó là an toàn để giả định, điều này càng trở nên khó khăn hơn. Các yêu cầu chi tiết cũng phải được tìm theo hướng khác, với yêu cầu cao cấp từ nơi chúng hình thành, để đảm bảo rằng tất cả được quy định đầy đủ. Thật khó khăn để thực hiện tất cả các chức năng mong muốn, vì vậy các yêu cầu thường được ưu tiên. Sự cần thiết, mong muốn, và tùy chọn được sử dụng để chỉ định các ưu tiên. Tổ chức cam kết cung cấp tính năng cần thiết, và nếu có thời gian họ thực hiện các tính năng mong muốn và sau đó là tùy chọn.

Khi yêu cầu chi tiết đã được thu thập, các tài liệu dự án được cập nhật để phản ánh tiến độ của dự án. Ví dụ, lịch thi đấu được cập nhật với ngày chính xác hơn, rủi ro ít hơn và biết được thêm nhiều thông tin hơn.

# 12.16 Bài tập

1. Mục tiêu chủ yếu của yêu cầu chi tiết hướng đến mọi người là gì?
2. Nêu tên cho năm cách tổ chức yêu cầu chi tiết.
3. Yêu cầu chi tiết nào dưới đây là sai? Giải thích cách sửa chúng.

a. HomeBudget sẽ hiển thị một giao diện thuận tiện cho việc nhập dữ liệu cá nhân.

b. SatControl phải tính toán thời điểm để mất vòng tròn trái đất trên quỹ đạo hiện tại và thời gian thực tế của vòng tròn trái đất trên quỹ đạo trước đây.

c. InvestKing phải xác định chiến lược đầu tư tốt nhất.

1. Nêu ba ưu điểm và nhược điểm của tổ chức yêu cầu chi tiết theo lớp hơn theo tính năng?
2. Giả sử rằng bạn đang xác định các yêu cầu cho một ứng dụng mô phỏng chiều hướng hoạt động của khách hàng tại một ngân hàng. Liệt kê năm lớp mà có thể được sử dụng để tổ chức yêu cầu.
3. Cung cấp các yêu cầu chi tiết cho từng lớp được xác định trong bài tập 3 bằng cách mô tả bằng một thuộc tính và một chức năng tương ứng với mỗi lớp.
4. Khi xác định các lớp miền (như trong hình 12.8), tại sao nó hữu ích để biểu thị mối quan hệ giữa chúng (tức là, thừa kế, tập hợp)?
5. Áp dụng các nguyên tắc thiết kế màn hình tốt nêu ở bước 3 Mục 12.3 phác thảo CUI tốt màn hình cho một ứng dụng tài chính nhà để:

a. Hiển thị một bản tóm tắt tài chính của người dùng theo loại hình tổ chức có tổ chức, và

b. Cho phép người dùng nhập vào các thông tin chi tiết cho một tổ chức mới

1. Đối với mỗi yêu cầu chi tiết được liệt kê trong bài tập 6, chỉ định một ưu tiên (ví dụ, thiết yếu, mong muốn, tùy chọn) và giải thích lý do tại sao bạn chọn chúng.
2. Tạo một biểu đồ tương tự như Hình 12.27 để kiểm tra đầu vào và đầu ra cho một trong những thuộc tính được xác định trong bài tập 6