**12**

**Phân tích các yêu cầu chi tiết**

* Các cách tổ chức 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 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 với, 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, vì vậy việc viết yêu cầu sẽ được tổ chức tốt hơn và dễ dàng cập nhật. Để đánh giá đúng giá trị của tổ chức một cách cẩn thận các yêu cầu chi tiết, hãy xem xét các nỗ lực hơn là ngẫu nhiên cố gắng viết yêu cầu chi tiết cho Encounter game. Lưu ý rằng những yêu cầu vẫn còn nguyên và không bị kiểm tra.

Mỗi nhân vật trong game Encounter đều phải có tên.

Mỗi nhân vật game có cùng phẩm chất đều có một điểm giá trị động.

Encounter phải mất ít hơn 1 giây để tính toán kết quả của 1 công việc.

Mỗi một khu vực có một tập hợp cụ thể của những “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 2 nhân vật của trò chơi Encounter đang ở cùng khu vực cùng một lúc họ có thể chọn lựa chọn khác hoặc thu hút lẫn nhau.

Mỗi nhân vật trò chơi có một số lượng HP.

Tổng của các giá trị của các phẩm chất của một nhân vật trò chơi có liên quan đến các khu vực trong câu hỏi sẽ được gọi là giá trị khu vực của nhân vật. Trong một sự dấn thân, hệ thống so sánh giá trị khu vực của các nhân vật và tính toán kết quả của sự tham gia.

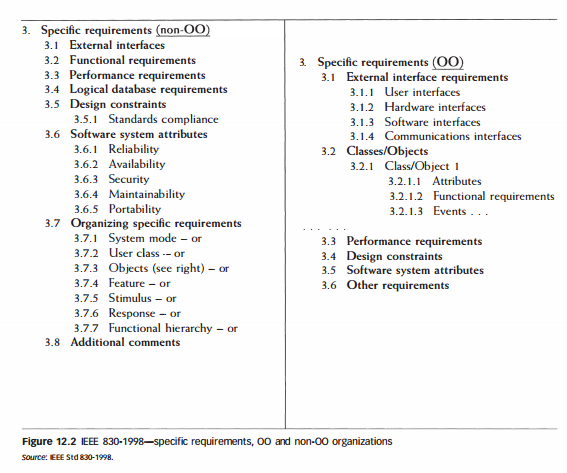
Tên của bất kỳ nhân vật trong Encounter sẽ không có nhiều hơn 15 chữ cái.

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

* Rất khó để hiểu như một đơn vị thậm chí trước khi nó phát triển thành hàng trăm, nếu không phải là hàng ngàn.
* Các yêu cầu của các loại hỗn hợp: yêu cầu thực hiện phải được xử lý khác với yêu cầu chức năng.
* Một số yêu cầu tự nhiên thuộc với những người liên quan.
* Rất khó để xác định một yêu cầu cụ thể.

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

Chuẩn IEEE 830-1998 cung cấp tài liệu mẫu cho một số cách để phân loại các yêu cầu chi tiết. Figure 12.2 cho thấy các đối tượng mẫu phân loại của tiêu chuẩn IEEE 830-1998 thông thường. Việc phân loại hướng 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 sử đổi phần cho phù hợp. Ví dụ, tổ chức 00 thiếu một phần tương đương với 3.4 trong tổ chức non-00 “các yêu cầu cơ sở dữ liệu hợp lý”.



**12.2.1 Tổ chức các yêu cầu chi tiết bằng Feature**

**12.2.2 Tổ chức các yêu cầu chi tiết bằng Use Case**

**12.2.3 Tổ chức các yêu cầu chi tiết bằng GUI**

**12.2.4 Tổ chức các yêu cầu chi tiết bằng State**

**12.2.5 Tổ chức các yêu cầu chi tiết bằng Class**

**12.2.6 Phương pháp phân loại: Ưu điểm và nhược điểm**

|  |  |  |
| --- | --- | --- |
|  | Ưu điểm | Nhược điểm |
| By Feature | Maps well to tại sao chúng ta xây dựng ứng dụng  Dễ hiểu | Does not map well to OO code  Khó để xác định các yêu cầu ngẫu nhiên. |
| By Use Case | Dễ hiểu | Use cases có thể chồng lên nhau  Khó để trace thiết kế và code  Sự theo dõi bị giới hạn |
| By GUI | Dễ hiểu |  |
| By State |  |  |
| By Class |  |  |