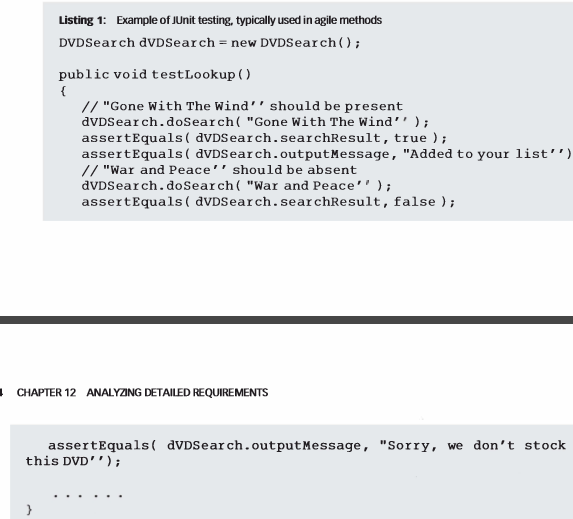
12.10 Các phương thức linh hoạt cho các yêu cầu chi tiết

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

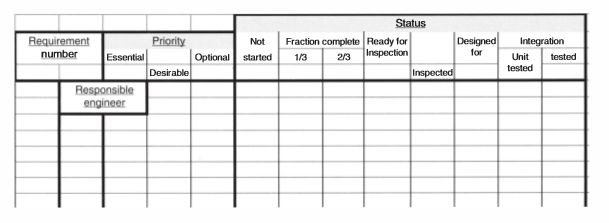
Phát triển test được một tả kỹ hơn trong chương 27

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 các công cụ và website cho phân tích 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

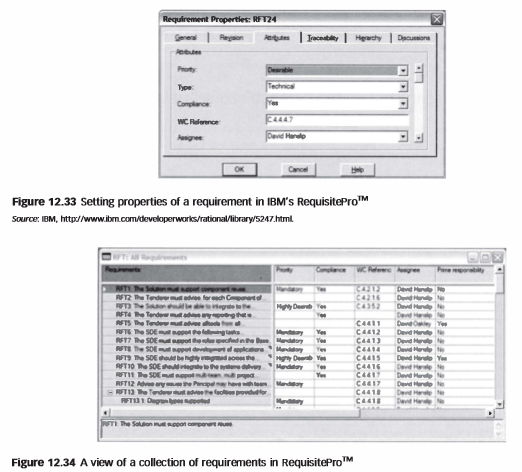
Tuy nhiên, hầu hết yêu cầu dự án có kích thước hợp lý là cần thiết. Thiết kế cho quá trình thiết kế chỉ ra rằng các yêu cầu được quản lý cho thiết keess. Unit test có nghĩa các yêu cầu đang thực thi trong code đã chịu sự thay đổi từ thiết kế độc lập. Intergration test có nghĩa là ứng dụng được test để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

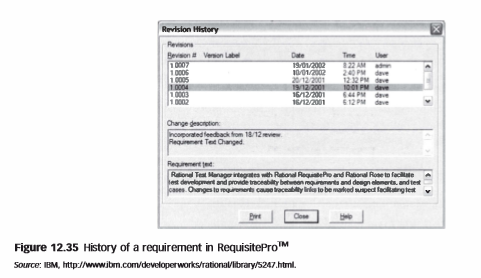
Các ô trong ma trận có thể nổi bật, làm rõ các phần trong document của project, vì thế, bảo quản tài liệu source code cho yêu cầu chi tiết.

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.

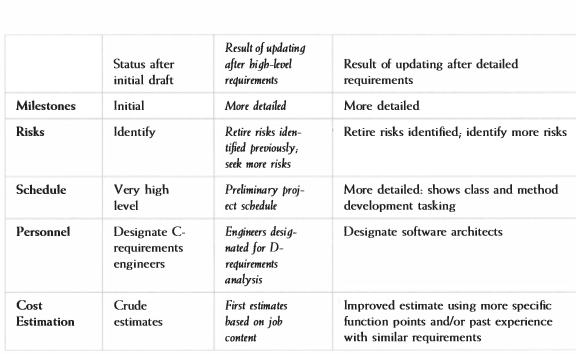




Nó có thể truy vấn cơ sở dữ liệu yêu cầu, hay chung quy, để toàn bộ yêu cầu thỏa mãn tiêu chí mong muốn, chẳng hạn bảo mật. Như các yêu cầu phân biệt được thực hiên bởi kỹ sư công nghệ phần mềm, các công cụ như vậy cho phép theo dõi lịch sử lướt web.

12.12 Hiệu quả trên dự án của xư lý 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.