4.5 4.6

Yêu cầu quy trình kỹ thuật, phân tích yêu cầu, xác nhận các yêu cầu, quản lý các yêu cầu

Mời nhân viên kỹ thuật làm việc với khách hàng để dễ dàng tìm hiểu các yêu cầu

Phân loại các yêu cầu, xem xét về mức độ ưu tiên cái nào nên triển khai trước

Các bên liên quan có thể có những yêu cầu khác nhau và các yêu cầu có thể thay đổi trong quá trình phân tích

(• Các loại bên liên quan

• Người dùng cuối

• Người quản lý hệ thống

• Chủ sở hữu hệ thống

• Các bên liên quan bên ngoài)

Used methods:

• Interviewing

• Ethnography (ét nó grow phi)

• Stories and scenarios (sớ ne ri ơ)

Chúng ta sẽ phòng vấn các bên liên quan

Có thể phỏng vấn mở hoặc kín

Khi phỏng vấn cần sẵn sàng lắng nghe các bên liên quan.

Phỏng vấn thực hành

• Thường là sự kết hợp giữa phỏng vấn kín và phỏng vấn mở.

• Người phỏng vấn cần có tư duy cởi mở ko nhất thiết phải bó buộc, bằng cách gợi ý

Chúng ta có các câu hỏi khi phỏng vấn như

Bạn muốn giải quyết vấn đề như thế nào?

• Ai sẽ là người sử dụng hệ thống?

• Người dùng có những kỹ năng máy tính nào?

Các yêu cầu phi chức năng của bạn là gì?

Tóm tắt vấn đề

Theo tối hiểu là bạn đang cần những thứ này, bạn cần phát triển theo hướng này đúng không?

Vấn đề bạn đang gặp có phải là vậy không? Nếu đúng vậy thì chúng ta sẽ giải quyết như sau:

Xác định các yêu cầu phi chức năng

Bạn cần bảo mật như thế nào?

Tính khả dụng của hệ thống ra sao?

Có bất kỳ yêu cầu pháp lý hoặc yêu cầu quy định nào khác cần được đáp ứng không?

Bạn có cần bổ sung thêm gì không?

Các nghiên cứu ethnography (ét nó grow phi) đã chỉ ra rằng tác phẩm thường phong phú hơn và nhiều hơn phức tạp hơn được đề xuất

Các Stories and scenarios (sớ ne ri ơ) là mô tả về cách một hệ thống có thể được sử dụng

cho một nhiệm vụ cụ thể.

**4.7 YÊU CẦU KỸ THUẬT REQUIREMENTS SPECIFICATION**

Người dùng cuối và khách hàng phải hiểu được( là nhũng người không có nền tảng kỹ thuật)

Yêu cầu hệ thống là những yêu cầu chi tiết hơn và có thể bao gồm

thêm thông tin kỹ thuật.

• Các yêu cầu có thể là một phần của hợp đồng cho hệ thống

sự phát triển

• Do đó, điều quan trọng là chúng phải đầy đủ nhất có thể

**Cách viết đặc tả yêu cầu hệ thống**

**Natural language**

Mỗi yêu cầu được đánh số

Mỗi câu cần diễn đạt một yêu cầu.

**Structured natural language**

Được viết trên bản mẫu

Mỗi trường cung cấp thông tin về một khía cạnh của yêu cầu.

**Ngôn ngữ mô tả thiết kế**

Sử dụng một ngôn ngữ giống như ngôn ngữ lập trình, nhưng nhiều tính trừu tượng (hiện nay ít được sử dụng mặc dù hữu ích cho người đặc tả giao diện)

**Đặc tả toán học**

Dựa trên các khái niệm toán học, hầu hết khách hàng không hiểu, khách hàng khnog thể kiểm tra xem nó có đại diện cho những gì họ muốn không

và miễn cưỡng chấp nhận nó như một hệ thống

**The software requirements document - Tài liệu yêu cầu**

Những gì được yêu cầu đối với các nhà phát triển hệ thống.

Bao gồm cả định nghĩa về yêu cầu của người dùng và

đặc điểm kỹ thuật của các yêu cầu hệ thống.

Nó KHÔNG phải là một tài liệu thiết kế.

**Users of a requirements document -Người dùng tài liệu yêu cầu**

khách hàng hệ thống-> xác định yêu cầu và đọc để kiểm tra có đáp ứng nhu cầu không

Người quản lý-> lập kế hoạch đấu thầu cho quá trình phát triển hệ thống

Kỹ thuật hệ thống-> phát triển kiêm trả xác thực cho hệ thống

Bảo trì hệ thống-> Hiểu hệ thống và mối quan hệ giữa các bộ phận của hệ thống

**Đặc tả bằng ngôn ngữ tự nhiên**

Yêu cầu được viết dưới dạng câu ngôn ngữ tự nhiên được bổ sung bằng sơ đồ và bảng.

Được sử dùng nhiều vì dễ hiểu cho người dùng và khách hàng

**Hướng dẫn về yêu cầu viết**

Làm ra định dạng tiêu chuẩn để sử dụng nó cho tất cả yêu cầu

Sử dụng ngôn ngữ một cách nhất quán

Sử dụng đánh dấu văn bản để xác định các phần chính của yêu cầu.

Tránh sử dụng biệt ngữ

Giải thích lí do tại sao lại có yêu cầu đó

**Các vấn đề với ngôn ngữ tự nhiên**

• Thiếu rõ ràng

• Khó chính xác mà không làm cho tài liệu khó đọc.

• Yêu cầu nhầm lẫn

• Các yêu cầu chức năng và phi chức năng có xu hướng trộn lẫn với nhau.

• Hợp nhất các yêu cầu

• Một số yêu cầu khác nhau có thể được thể hiện cùng nhau

**Thông số kỹ thuật có cấu trúc**

Một cách tiếp cận để viết các yêu cầu trong đó quyền tự do của người viết yêu cầu bị hạn chế và các yêu cầu được viết theo cách chuẩn.

• Điều này hoạt động tốt đối với một số loại yêu cầu, ví dụ: yêu cầu cho

hệ thống điều khiển nhúng nhưng đôi khi quá cứng nhắc để viết

yêu cầu hệ thống kinh doanh.

**Thông số kỹ thuật dựa trên biểu mẫu**

Định nghĩa của chức năng hoặc thực thể.

Mô tả các yếu tố đầu vào và nguồn gốc của chúng.

Mô tả kết quả đầu ra và nơi chúng đi đến.

Thông tin về thông tin cần thiết cho việc tính toán và các thực thể khác được sử dụng.

Mô tả hành động sẽ được thực hiện.

Điều kiện trước và sau (nếu phù hợp).

Các tác dụng phụ (nếu có) của hàm.

**Đặc điểm kỹ thuật dạng bảng**

Dùng để bổ sung ngôn ngữ tự nhiên.

• Đặc biệt hữu ích khi bạn phải xác định một số

Các khóa học thay thế.

• Ví dụ, hệ thống bơm insulin dựa trên các tính toán của nó dựa trên tốc độ thay đổi của lượng đường trong máu và thông số kỹ thuật dạng bảng

giải thích cách tính nhu cầu insulin cho các tình huống khác nhau.

**Use cases**

là một loại kịch bản được bao gồm trong UML.

xác định các tác nhân trong một tương tác và mô tả

tương tác chính nó.

Một tập hợp các use case phải mô tả tất cả các tương tác có thể có với hệ thống

• Mô hình đồ họa cấp cao được bổ sung bằng bảng chi tiết hơn

sự mô tả.

• Biểu đồ trình tự UML có thể được sử dụng để thêm chi tiết cho các trường hợp sử dụng bằng cách

hiển thị trình tự xử lý sự kiện tron

**Xác thực yêu cầu**

• Quan tâm đến việc chứng minh rằng các yêu cầu xác định hệ thống mà khách hàng thực sự muốn.

• Chi phí lỗi yêu cầu cao nên việc xác nhận là rất quan trọng

• Sửa lỗi yêu cầu sau khi giao hàng có thể tốn gấp 100 lần chi phí sửa lỗi thực hiện.

**Kiểm tra yêu cầu**

• Hiệu lực. Hệ thống có cung cấp các chức năng hỗ trợ tốt nhất cho

nhu cầu của khách hàng?

• Tính nhất quán. Có bất kỳ xung đột yêu cầu nào không?

• Tính hoàn chỉnh. Có phải tất cả các chức năng theo yêu cầu của khách hàng không?

• Chủ nghĩa hiện thực. Các yêu cầu có thể được thực hiện với ngân sách khả dụng không

Và công nghệ

• Khả năng xác minh. Có thể kiểm tra các yêu cầu không?

**Các kỹ thuật xác nhận yêu cầu**

• Đánh giá yêu cầu

• Phân tích thủ công có hệ thống các yêu cầu.

• Prototyping

• Sử dụng một mô hình thực thi của hệ thống để kiểm tra các yêu cầu.

• Tạo trường hợp thử nghiệm

• Phát triển các bài kiểm tra cho các yêu cầu để kiểm tra khả năng kiểm thử.

**ACTIVITY DIAGRAM**

**Sơ đồ Activity Diagram?**

Một Activity Diagram trong mô hình use-case được sử dụng để nắm bắt các hoạt động và hành động được thực hiện trong use-case

• Về cơ bản, nó là một biểu đồ, thể hiện luồng kiểm soát từ hoạt động này sang hoạt động hoặc hành động khác.

• Nó bao gồm các hành động, các nút và sự chuyển tiếp giữa các hoạt động và trạng thái.

**Activity Diagrams**

• Mô hình quy trình công việc kinh doanh.

• Xác định các trường hợp sử dụng ứng viên, thông qua việc kiểm tra hoạt động kinh doanh

quy trình làm việc.

• Xác định các điều kiện trước và sau cho các use-case

• Mô hình hóa quy trình làm việc giữa / trong các use-case

**The notation | The basics Kí hiệu | Những thứ cơ bản**

**Activities- các hoạt động**

• Một Hoạt động là một quá trình đang được mô hình hóa.

• Hoạt động là một đơn vị công việc cần được thực hiện.

• Bất kỳ Hoạt động nào cũng cần có thời gian.

**Actions-Hành động**

• Một Hành động là một bước trong hoạt động tổng thể.

• Công việc có thể được ghi lại dưới dạng các Hành động trong hoạt động.

Action’s Name

**Transitions-Chuyển tiếp**

• Với các mũi tên chỉ hướng, các đường chuyển tiếp trên một hoạt động biểu đồ hiển thị dòng tuần tự của các hành động trong hoạt động được mô hình hóa.

• Mũi tên sẽ luôn trỏ đến hành động tiếp theo trong hoạt động sự nối tiếp.

**Initial Nodes - Các nút ban đầu**

• Nút ban đầu hiển thị rõ ràng điểm bắt đầu cho chuỗi hành động trong một

sơ đồ hoạt động.

• Nút ban đầu được vẽ dưới dạng một vòng tròn đặc với một đường chuyển tiếp (mũi tên)

kết nối nó với hành động đầu tiên trong chuỗi hành động của hoạt động.

• Có thể có nhiều hơn một nút ban đầu. Trong trường hợp này, việc gọi hoạt động bắt đầu

nhiều luồng, một luồng tại mỗi nút ban đầu.

• Chỉ có thể có một đường chuyển tiếp kết nối nút ban đầu với một hành động.

**Final Nodes -Các nút cuối cùng**

• Nút cuối cùng chỉ ra rằng kết quả hành động của hoạt động hoặc một luồng đã kết thúc.

• Có hai loại nút cuối cùng: nút cuối cùng của hoạt động và nút cuối cùng của luồng.

• Nút cuối cùng của hoạt động được vẽ dưới dạng một vòng tròn bao quanh một vòng tròn đặc nhỏ hơn.

• Nút cuối cùng của luồng được vẽ dưới dạng một vòng tròn với một chữ thập bên trong.

• Một nút cuối cùng của hoạt động dừng tất cả các luồng trong một hoạt động và chấm dứt toàn bộ hoạt động.

• Nút cuối cùng của luồng kết thúc một đường dẫn thông qua một hoạt động, nhưng không kết thúc toàn bộ hoạt động.

• Mỗi sơ đồ hoạt động nên có ít nhất một biểu tượng nút cuối cùng.

• Biểu đồ hoạt động có thể hiển thị nhiều nút cuối cùng. Nói cách khác, hoạt động có thể chấm dứt theo cách cư xử khác nhau.

**The notation | Beyond the basics -Kí hiệu | Ngoài những điều cơ bản**

**Decision nodes- Các nút quyết định**

• Một nút quyết định cho thấy nơi mà quá trình chuyển đổi thoát khỏi một hành động có thể phân nhánh trong các hướng thay thế tùy thuộc vào điều kiện.

• Một quyết định được vẽ như một viên kim cương trên một sơ đồ hoạt động.

• Vì một quyết định sẽ có ít nhất hai kết quả khác nhau, biểu tượng quyết định sẽ

có nhiều đường chuyển tiếp kết nối với các hành động khác nhau.

**Decision nodes > Guard conditions -Các nút quyết định> Điều kiện bảo vệ**

• Một điều kiện bảo vệ cho biết rõ ràng khi nào đi theo đường chuyển tiếp sang hành động tiếp theo.

• Văn bản điều kiện bảo vệ luôn được đặt trong dấu ngoặc.

Ví dụ: [văn bản điều kiện bảo vệ].

• Bảo vệ [else] thường được sử dụng trong sơ đồ hoạt động có nghĩa là "nếu không có

các dòng chuyển tiếp được bảo vệ khác phù hợp với điều kiện thực tế ", sau đó làm theo [else] đường chuyển tiếp.

**Merge nodes- Hợp nhất các nút**

• Một nút hợp nhất tập hợp các luồng thay thế lại với nhau thành một luồng đầu ra duy nhất.

• Một nút hợp nhất sử dụng cùng một biểu tượng hình thoi với nhiều đường dẫn trỏ đến nó, nhưng chỉ với một dòng chuyển tiếp thoát ra khỏi nó.

• Hợp nhất không đồng bộ hóa nhiều luồng đồng thời.

**Synch states -Đồng bộ các trạng thái**

• Một số chuỗi hành động có thể được thực hiện song song.

• Các chuỗi hành động song song được đặt tên chính thức là trạng thái đồng bộ.

• Trạng thái đồng bộ là nơi chuyển tiếp thành nhiều đường dẫn hoặc nhiều đường dẫn được kết hợp thành một quá trình chuyển đổi duy nhất.

**Fork Nodes - Nút ngã ba**

• Ngã ba là nơi con đường phân chia.

• Một nút ngã ba chia luồng đến thành nhiều luồng đồng thời.

**Join Nodes - Tham gia các nút**

• Một phép nối là nơi nhiều đường dẫn đồng thời gặp nhau.

Một kết nối đồng bộ hóa nhiều dòng vào và tạo ra một dòng ra duy nhất. Dòng ra từ một phép nối không thể thực hiện cho đến khi nhận được tất cả các dòng vào.

**Synch states > Example -Đồng bộ hóa các trạng thái> Ví dụ**

**Swimlane**

Swimlane được sử dụng để lập mô hình quy trình kiểm soát theo thủ tục của hoạt động

giữa các đối tượng (cá nhân, tổ chức hoặc người chịu trách nhiệm khác

thực thể) thực sự thực hiện hành động.