

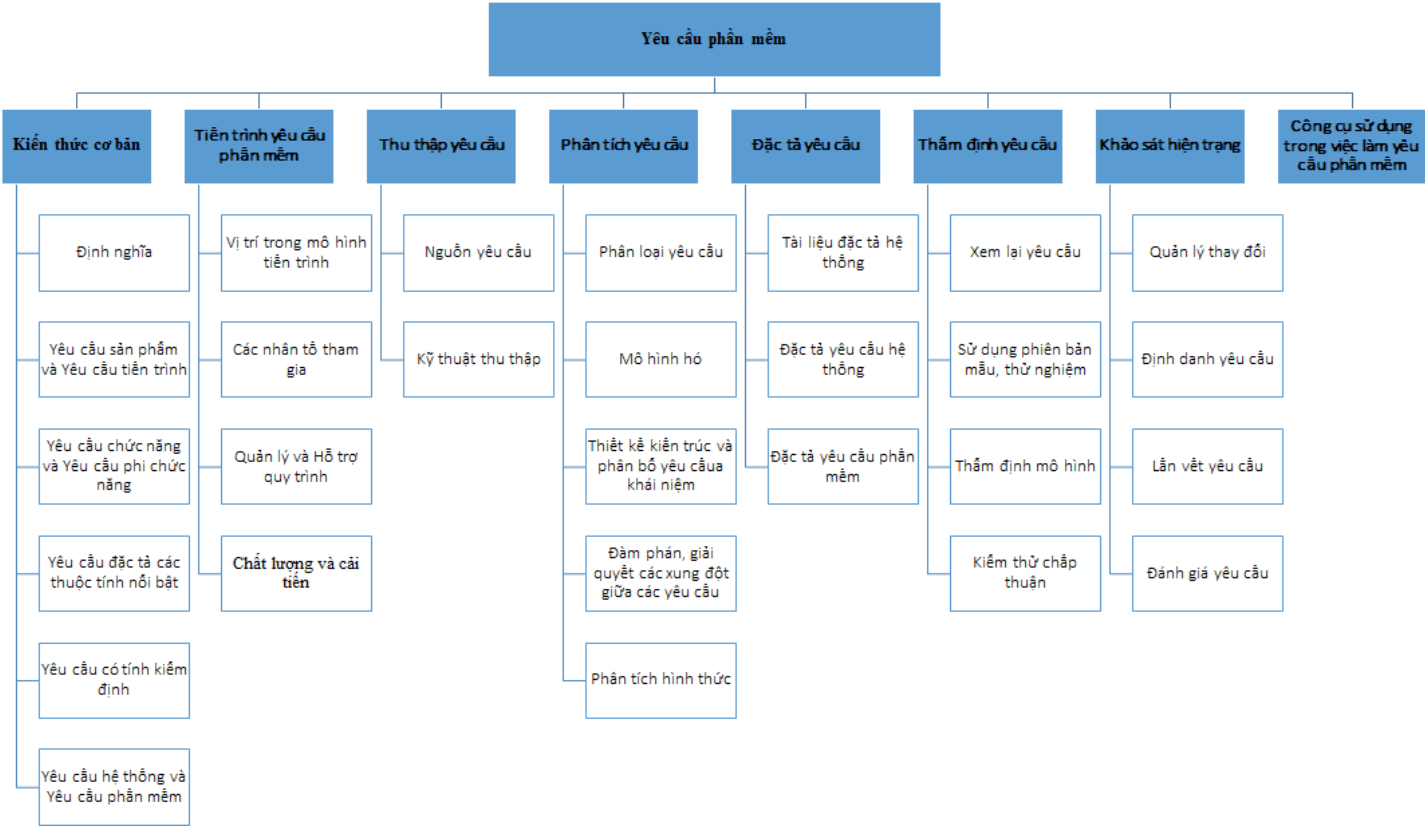
Chương 1: Yêu cầu phần mềm

Thuật ngữ thường dùng

- **CIA**: Confidentiality, Integrity, and Availability(Độ tin cậy, Tính toàn vẹn, tính khả dụng)
- **DAG**: Directed Acyclic Graph (Đồ thị có hướng và không có chu trình)
- **FSM**: Functional Size Measurement (Thước đo chức năng)
- **INCOSE**: International Council on Systems Engineering ()
- **UML**: Unified Modeling Language (Ngôn ngữ mô hình hóa)
- **SysML**: Systems Modeling Language (Ngôn ngữ mô hình hóa hệ thống)

1.1. Kiến thức cơ bản

Tổng quan



Định nghĩa

- Yêu cầu cho 1 phần mềm cụ thể là tổng hợp những yêu cầu từ nhiều người khác nhau về tổ chức, mức độ chuyên môn và mức độ tham gia, tương tác với phần mềm trong môi trường hoạt động của nó.
- Có thể kiểm chứng một cách riêng rẽ ở mức chức năng(yêu cầu chức năng) hoặc mức hệ thống(yêu cầu phi chức năng)
- Cung cấp các chỉ số đánh giá độ ưu tiên về các mặt khi cân nhắc về nguồn tài nguyên.
- Cung cấp các giá trị trạng thái để theo dõi tiến độ của dự án.

Phân loại

- Theo sản phẩm và tiến trình
 - Yêu cầu sản phẩm: là những đòi hỏi hay ràng buộc mà phần mềm phải thực hiện.

- Yêu cầu tiến trình: là những ràng buộc liên quan đến việc phát triển phần mềm đó (quy trình, đối tác kiểm thử, phân tích, kĩ thuật sử dụng,...).

Ví dụ: Trong Project 1:

Yêu cầu sản phẩm là xây dựng trang web Trường học điện tử với các tính năng như Giáo viên quản lý câu hỏi, đề thi; Học sinh tham gia làm bài; Admin duyệt câu hỏi của giáo viên trước khi đăng,...

Yêu cầu tiến trình: Phải thực hiện theo mô hình Agile. Sản phẩm cuối cùng bao gồm cả sản phẩm và backlog cho từng Sprint.

- Theo chức năng
 - Yêu cầu chức năng: đặc tả các chức năng mà phần mềm phải thực hiện.
 - Yêu cầu phi chức năng: là các ràng buộc về giải pháp và chất lượng (hiệu năng, việc bảo trì, độ an toàn, bảo mật,...).
 - Yêu cầu đặc tả các thuộc tính nổi bật: là các đặc tả cho các thuộc tính phụ thuộc vào sự vận hành,... đặc biệt là kiến trúc hệ thống. Các thuộc tính này không thể xác định được cho từng thành phần đơn lẻ.
- Theo tính kiểm định
 - Mơ hồ, không thể kiểm định
 - Rõ ràng, định lượng và có thể kiểm định được.
- Theo phạm vi đặc tả
 - Yêu cầu hệ thống: đặc tả các cấu hình, cơ sở hạ tầng, phần cứng, phần mềm, con người, kỹ thuật,... của toàn bộ hệ thống.
 - Yêu cầu phần mềm: đặc tả các chức năng, giao diện,... của các cấu phần phần mềm.

1.2. Tiến trình yêu cầu phần mềm

Vị trí trong mô hình tiến trình

- Xuất phát từ giai đoạn khởi tạo dự án, cho tới khi hoàn thiện các tiến trình trong vòng đời phát triển của phần mềm. Không chỉ là các hoạt động bề nổi nhìn thấy được.
- Được nhận biết như phần cấu hình của mọi việc, và được quản lí như việc quản lí các cấu hình; hoặc là sản phẩm của các quá trình trong vòng đời phát triển.
- Được thiết kế theo từng tổ chức và hoàn cảnh của từng dự án.

Các nhân tố tham gia

- Người dùng: vận hành phần mềm.
- Khách hàng: gồm những người được hưởng mục tiêu cuối cùng của phần mềm, những giá trị thương mại mà phần mềm hướng đến.
- Nhà phân tích thị trường: phân tích nhu cầu thị trường và tương tác với bên đại diện khách hàng.
- Đại diện pháp lý: kiểm soát việc tuân thủ quy định, quy ước chung, quy định pháp lý.
- Kỹ sư phần mềm: thu lợi nhuận hợp pháp từ việc phát triển phần mềm.

Quản lý và Hỗ trợ quy trình

- Quản lí nguồn tài nguyên được sử dụng trong các tiến trình.
- Cân đối nguồn nhân lực, tài chính, đào tạo và công cụ.

Chất lượng và cải tiến

- Xác định vai trò của tiến trình xây dựng yêu cầu về các mặt chi phí, thời gian và sự thoả mãn của khách hàng với sản phẩm.
- Định hướng tiến trình theo các chuẩn chất lượng và xây dựng mô hình cải tiến cho phần mềm và hệ thống.
- Bao gồm:
 - Độ bao phủ theo các mô hình và chuẩn cải tiến.
 - Việc đo đạc và đánh giá tiến trình.
 - Việc thực hiện và lên kế hoạch cải tiến.
 - Việc cài đặt và lên kế hoạch cho an ninh.

1.3. Thu thập yêu cầu

Là giai đoạn đầu tiên trong việc xây dựng sự hiểu biết về sản phẩm phần mềm và các vấn đề cần thiết phải giải quyết (ví dụ cần hiểu biết về các chức năng của phần mềm). Đây cũng là giai đoạn mà các bên liên quan (stakeholders) được xác định. Thiết lập các mối quan hệ giữa các nhóm phát triển và khách hàng.

Một trong những nguyên tắc cơ bản của quá trình thu thập yêu cầu là sự trao đổi giữa các bên liên quan. Sự trao đổi liên tục qua toàn bộ vòng đời phát triển phần mềm (SDLC), quá trình trao đổi với các bên liên quan khác nhau tại mỗi các thời điểm khác nhau. Trước khi bắt đầu phát triển, các chuyên gia thu thập yêu cầu có thể tạo ra các kênh cho sự giao tiếp này. Họ sẽ là trung gian giữa khách hàng và kỹ sư phần mềm.

Một số lợi ích của thu thập yêu cầu:

- Tạo được niềm tin của khách hàng khi họ được tham gia vào giai đoạn thu thập yêu cầu.
- Giảm việc phải làm lại trong quá trình phát triển
- Quá trình phát triển sẽ nhanh hơn, giảm được những chi phí cho những yêu cầu không cần thiết.
- Hạn chế phạm vi hệ thống bị phình rộng.

1.3.1. Nguồn yêu cầu - Requirements Sources

Các yêu cầu có rất nhiều nguồn trong đặc thù phần mềm và điều quan trọng là tất cả các nguồn tiềm năng cần được xác minh và đánh giá. Phần này nhằm nâng cao nhận thức của các nguồn khác nhau của yêu cầu phần mềm và những framework để quản lý chúng.

Những điểm chính của nguồn yêu cầu bao gồm:

- Mục tiêu - Goal. Các mục tiêu về giá trị và giá thành thường mơ hồ, không rõ ràng. Kỹ sư phần mềm cần chú ý để xác định rõ các mục tiêu đó. Nghiên cứu tính khả thi là sẽ giúp giảm giá thành của quá trình phát triển. Ví dụ kỹ sư phần mềm cần xác định chi phí xây dựng server với chi phí đi mua cái nào sẽ tối ưu hơn để lựa chọn.
- Hiểu biết về các lĩnh vực. Các kỹ sư phần mềm cần có kiến thức về các lĩnh vực như: mua sắm, ngân hàng, chăm sóc sức khỏe,... lĩnh vực mà phần mềm được sử dụng. Việc hiểu biết về các lĩnh vực sẽ giúp cho người thu thập yêu cầu thu thập được những thông tin chính xác cao.
- Các bên liên quan. Nhiều phần mềm đã được chứng minh không đạt yêu cầu vì nó chỉ tập trung vào yêu cầu của một số bên mà bỏ qua các bên khác. Do đó phần mềm đã giao rất khó để sử dụng hoặc phá vỡ văn hóa hoặc tổ chức chính trị của tổ chức khách hàng. Các kỹ sư phần mềm cần phải xác định, miêu tả và quản lý các yêu cầu của các bên liên quan. Ví dụ phần mềm cho người không chuyên thì sử dụng chuột và các menu chọn, nhưng với người thành thạo thì cần có các hot-key để rút ngắn thời gian tương tác
- Nguyên tắc kinh doanh. Là những điều kiện hoặc các ràng buộc được xác định để các doanh nghiệp hoạt động được. "Một sinh viên không thể đăng ký vào các khóa học học kỳ tiếp theo nếu vẫn còn một số môn chưa thanh toán học phí" sẽ là một ví dụ của nguyên tắc kinh doanh đó cho các phần mềm đăng ký môn học của trường đại học.
- Môi trường vận hành. Các yêu cầu sẽ được bắt nguồn từ môi trường mà trong đó phần mềm sẽ được thực thi. Ví dụ như ràng buộc thời gian trong phần mềm thời gian thực hoặc ràng buộc hiệu năng trong môi trường kinh doanh.
- Môi trường tổ chức. Phần mềm thường có thể bị ràng buộc bởi cấu trúc, văn hóa và tổ chức chính trị. Các kỹ sư phần mềm cần phải hiểu biết về chúng, phần mềm không nên ép buộc thay đổi ngoài ý muốn trong quá trình kinh doanh.

1.3.2. Kỹ thuật thu thập- Elicitation Techniques

Một khi các nguồn yêu cầu được xác định, các kỹ sư phần mềm có thể bắt đầu thu thập thông tin yêu cầu từ chúng. Phần này tập trung vào các kỹ thuật để thu thập các thông tin cần thiết từ các bên liên quan.

Các kỹ sư phần mềm cần phải linh hoạt với các sự việc xảy ra ví dụ: người dùng gặp khó khăn trong việc mô tả yêu cầu của họ, có thể thông tin quan trọng không được nói ra hoặc có thể không muốn hoặc không thể hợp tác.

Thu thập không phải là hoạt động thụ động, ngay cả khi các bên liên quan sẵn sàng hợp tác các kỹ sư phần mềm phải cố gắng để thu thập thông tin chính xác nhất.

Một số kỹ thuật thu thập yêu cầu như:

- Phỏng vấn. Phỏng vấn là cách truyền thống để thu thập yêu cầu. Ưu điểm: Thu thập được những thông tin trực tiếp các thông tin có chất lượng cao, tính chân thực và độ tin cậy có thể kiểm nghiệm được trong quá trình phỏng vấn. Nhược điểm: Đòi hỏi người phỏng vấn phải có trình độ cao, am hiểu, có kỹ năng xử lý các tình huống. Khó triển khai được trên quy mô rộng. Tiếp cận khách hàng là việc tương đối khó vì cần hẹn trước.
- Kịch bản. Kỹ sư phần mềm cung cấp một hệ thống các câu hỏi bằng việc sử dụng các câu hỏi như "What if" và "How is this done". Các loại kịch bản phổ biến được sử dụng là mô tả các trường hợp sử dụng. Ví dụ: Điều gì sẽ xảy ra với phần mềm khi bị mất kết nối mạng,...
- Các bản mẫu. Kỹ thuật này sẽ làm rõ các yêu cầu không rõ ràng. Có thể sử dụng mockup hoặc màn hình thiết kế hoặc các bản thử nghiệm để xác minh những yêu cầu từ khách hàng. Ví dụ: Thu thập các yêu cầu về thiết kế hoặc chức năng của phần mềm.
- Cuộc hội họp. Một nhóm người sẽ mang lại nhiều cái nhìn sâu sắc hơn vào các yêu cầu phần mềm. Mọi người sẽ cùng suy nghĩ, tinh chỉnh các yêu cầu. Lợi thế của phương pháp này là các yêu cầu mâu thuẫn sẽ dễ dàng xử lý hơn.

1.4. Phân tích yêu cầu

Nhằm mục đích:

- Phát hiện và giải quyết xung đột giữa các yêu cầu
- Tìm ra những giới hạn của phần mềm và cách phần mềm tương tác với tổ chức và môi trường hoạt động của nó.
- Nghiên cứu các yêu cầu hệ thống để lấy được các yêu cầu phần mềm.

1.4.1. Mô hình hóa khái niệm

Xây dựng các mô hình trong một vấn đề thực tế là chìa khóa của phân tích yêu cầu phần mềm. Mục đích của nó là để hiểu rõ về những vấn đề xảy ra cũng như miêu tả được giải pháp của vấn đề.

Do đó mô hình khái niệm bao gồm các mô hình của các thực thể từ miền vấn đề, cấu hình để phản ánh các mối quan hệ trong thế giới thực và ràng buộc.

Có rất nhiều loại mô hình có thể được phát triển bao gồm biểu đồ use case, mô hình luồng dữ liệu, mô hình trạng thái, mô hình dựa trên mục tiêu, tương tác người dùng, mô hình dữ liệu, mô hình đối tượng...

Các yếu tố ảnh hưởng đến sự lựa chọn mô hình bao gồm:

- Vấn đề tự nhiên. Một số loại phần mềm đòi hỏi một số khía cạnh được phân tích đặc biệt nghiêm ngặt. ví dụ mô hình trạng thái và mô hình tham số của phần mềm thời gian thực quan trọng hơn so với hệ thống thông tin.
- Sự thành thạo của kỹ sư phần mềm. Kỹ sư phần mềm có kinh nghiệm sẽ lựa chọn các mô hình hay phương pháp để được kết quả tốt hơn.
- Các yêu cầu về quy trình của khách hàng. Khách hàng có thể áp đặt các ký hiệu ưa thích của họ hoặc phương pháp hoặc ngăn cản bất cứ cái gì mà họ thấy không quen thuộc. Nhân tố này có thể xung đột với các nhân tố trước đó.

1.4.2. Thiết kế kiến trúc và phân bổ yêu cầu

Thiết kế kiến trúc là điểm mà tại đó quá trình yêu cầu trùng lặp với phần mềm hoặc các hệ thống thiết kế. Trong nhiều trường hợp, các hành vi kỹ sư phần mềm như là kiến trúc sư phần mềm bởi vì quá trình phân tích và xây dựng các yêu cầu đòi hỏi rằng các thành phần kiến trúc / thiết kế đó sẽ chịu trách nhiệm đáp ứng các yêu cầu được xác định.

Phân bổ là quan trọng để cho phép phân tích chi tiết các yêu cầu. Do đó, ví dụ, khi một bộ các yêu cầu đã được phân bổ cho một thành phần, các yêu cầu cá nhân có thể được phân tích thêm để khám phá thêm các yêu cầu về cách thành phần tương tác với thành phần khác để đáp ứng các yêu cầu được giao.

Trong các dự án lớn, phân bổ thúc đẩy một vòng mới của phân tích cho mỗi hệ thống. Thiết kế kiến trúc được xác định chặt chẽ với các mô hình khái niệm.

1.4.3. Đàm phán, giải quyết các xung đột giữa các yêu cầu

Điều này liên quan đến việc giải quyết vấn đề giữa hai yêu cầu của các bên liên quan cùng các tính năng không tương thích, giữa các yêu cầu và nhân lực hoặc giữa yêu cầu chức năng và yêu cầu phi chức năng.

Trong tất cả các trường hợp kỹ sư phần mềm không được tự đưa ra các quyết định mà cần thiết tham khảo từ các bên liên quan để đạt được một sự đồng thuận trên sự thỏa hiệp thích hợp.

Tuy nhiên, thường rất khó khăn để có được thông tin thực. Ngoài ra, các yêu cầu thường phụ thuộc vào nhau, và có ưu tiên tương đối. Trong thực tế, các kỹ sư phần mềm thực hiện các yêu cầu ưu tiên thường xuyên mà không biết về tất cả các yêu cầu. Nó cũng bao gồm một phân tích từ các kỹ sư phần mềm ước tính chi phí thực hiện từng yêu cầu hoặc liên quan đến các yêu cầu khác.

1.4.4. Phân tích hình thức - Formal Analysis

Formal Analysis đã có một tác động trên một số lĩnh vực ứng dụng, đặc biệt là các hệ thống toàn vẹn cao. Các hình thức thể hiện của các yêu cầu đòi hỏi một ngôn ngữ với ngữ nghĩa định nghĩa chính thức(vd: Ngôn ngữ Z).

Việc sử dụng một phân tích hình thức cho các yêu cầu biểu hiện có hai lợi ích. Đầu tiên, nó cho phép các yêu cầu thể hiện bằng ngôn ngữ được xác định một cách chính xác và rõ ràng, do vậy tránh được khả năng hiểu sai.

Thứ hai, yêu cầu có thể được lý giải trên, cho phép đặc tính mong muốn của phần mềm cụ thể để chứng minh.

Phân tích hình thức nhất là tập trung vào giai đoạn khá muộn của phân tích yêu cầu.

1.5. Đặc tả yêu cầu

Đặc tả yêu cầu là một mô tả của hệ thống phần mềm được phát triển, đưa ra các yêu cầu chức năng và phi chức năng, và có thể bao gồm một tập hợp các ca sử dụng (use cases) để mô tả tương tác giữa người dùng với phần mềm.

Đặc tả yêu cầu tạo cơ sở cho một thỏa thuận giữa khách hàng và nhà cung cấp về những gì phần mềm đã làm được tốt cũng như

những gì chưa được như mong đợi. Nó cũng cung cấp một cơ sở thực tế để ước tính giá thành sản phẩm, rủi ro và lịch trình. Đối với các hệ thống phức tạp có 3 loại tài liệu được tạo ra là: định nghĩa hệ thống, yêu cầu hệ thống và các yêu cầu phần mềm. Đối với sản phẩm phần mềm đơn giản chỉ cần 1 trong 3 tài liệu.

1.5.1. Tài liệu đặc tả hệ thống.

Còn được biết như là tài liệu yêu cầu người dùng hay là tài liệu vận hành ghi lại những yêu cầu hệ thống. Nó xác định yêu cầu hệ thống ở mức cao với cách nhìn từ domain. Độc giả của tài liệu bao gồm hệ thống người dùng hoặc khách hàng. Vì vậy nội dung của nó phải được diễn đạt bằng những từ ngữ của những lĩnh vực riêng. Tài liệu sẽ liệt kê các yêu cầu hệ thống cùng với các thông tin cơ bản về đối tượng hệ thống, môi trường mục tiêu của nó, giả định và các yêu cầu phi chức năng. Nó có thể bao gồm mô hình khái niệm được thiết kế để minh họa cho ngữ cảnh hệ thống, sử dụng kịch bản, và các miền thực thể chính, cũng như luồng công việc.

1.5.2. Đặc tả yêu cầu hệ thống

Người phát triển những dự án phần mềm có những thành phần thuần túy là software và những phần non-software - ví dụ như máy bay hiện đại thường tách biệt yêu cầu hệ thống với yêu cầu phần mềm. Theo quan điểm này, yêu cầu hệ thống được quy định, các yêu cầu phần mềm có nguồn gốc từ các yêu cầu hệ thống, và sau đó các yêu cầu đối với các thành phần phần mềm được xác định.

1.5.3. Đặc tả yêu cầu phần mềm

Đặc tả yêu cầu phần mềm tạo cơ sở cho việc thỏa thuận giữa khách hàng và nhà thầu hoặc các nhà cung cấp về những gì sản phẩm phần mềm có làm việc đúng như mong muốn không. Nó cho phép một đánh giá nghiêm ngặt các yêu cầu trước khi có thể bắt đầu vào việc thiết kế và làm giảm việc thiết kế lại. Nó cũng cần cung cấp một cơ sở thực tế để ước tính giá thành sản phẩm, rủi ro, và lịch trình.

Các tổ chức cũng có thể sử dụng một tài đặc tả yêu cầu phần mềm làm cơ sở để phát triển kế hoạch kiểm tra và xác minh. Đặc tả yêu cầu phần mềm cung cấp một cơ sở thông báo cho chuyển một sản phẩm phần mềm cho người dùng mới hoặc các nền tảng phần mềm. Cuối cùng, nó có thể cung cấp một cơ sở để nâng cao phần mềm. Yêu cầu phần mềm thường được viết bằng ngôn ngữ tự nhiên, nhưng đặc tả yêu cầu phần mềm có thể được bổ sung bằng các mô tả chính thức hoặc gần chính thức. Lựa chọn các ký hiệu thích hợp và các khía cạnh của kiến trúc phần mềm cụ thể được mô tả chính xác hơn so với ngôn ngữ tự nhiên. Các nguyên tắc chung là ký hiệu nên được sử dụng cho phép các yêu cầu để được mô tả là chính xác càng tốt. Điều này đặc biệt quan trọng đối với các phần mềm an toàn cao và một số loại phần mềm đáng tin cậy khác. Tuy nhiên, sự lựa chọn của các ký hiệu thường được hạn chế bởi việc đào tạo, kỹ năng, và sở thích của các tác giả và độc giả.

Một số chỉ tiêu chất lượng đã được phát triển có thể được sử dụng liên quan đến chất lượng của đặc tả yêu cầu phần mềm như chi phí, hài lòng, hiệu quả, đúng tiến độ, và tái sản xuất. Chỉ tiêu chất lượng cho đặc tả yêu cầu của cá nhân bao gồm mệnh lệnh, chỉ thị, các pha yếu, tùy chọn, và sự duy trì. Các chỉ số cho các tài liệu đặc tả yêu cầu phần mềm bao gồm kích thước, dễ đọc, đặc tả kỹ thuật, chiều sâu và cấu trúc văn bản.

u đặc tả yêu cầu phần mềm bao gồm kích thước, dễ đọc, đặc tả kỹ thuật, chiều sâu và cấu trúc văn bản.

1.6. Thẩm định yêu cầu

Tất cả tài liệu yêu cầu cần được thông qua quá trình thẩm định và kiểm duyệt. Vậy Thẩm định yêu cầu là j?

Khái niệm

- Thẩm định yêu cầu quan tâm đến việc chứng tỏ rằng các yêu cầu định nghĩa được hệ thống mà khách hàng thực sự muốn. Các yêu cầu phải được thẩm định để đảm bảo rằng người thực thi, người lập trình hiểu được yêu cầu.
- Việc thẩm định phải đảm bảo dễ hiểu, nhất quán và hoàn thiện
- Thẩm định yêu cầu được quan tâm đến quá trình kiểm tra tài liệu yêu cầu có đảm bảo đầu ra là 1 phần mềm hoàn chỉnh, đúng đắn.

Các kĩ thuật thẩm định yêu cầu

- xem lại yêu cầu
- sử dụng phiên bản mẫu, thử nghiệm
- thẩm định mô hình
- kiểm thử chấp thuận

1.6.1. Xem lại yêu cầu

Có lẽ phương tiện phổ biến nhất của việc thẩm định là sự kiểm tra hoặc xem lại các tài liệu yêu cầu. Một nhóm người được giao nhiệm vụ tìm các lỗi, sai sót, thiếu sự rõ ràng, và độ lệch so với tiêu chuẩn. Thành phần của nhóm này đặc biệt quan trọng (ít nhất cần có đại diện khách hàng để có được định hướng đúng đắn), và họ có thể cung cấp sự hướng dẫn trong việc tìm kiếm thông tin chuẩn xác. Việc nhận xét có thể được thành lập sau khi hoàn thành các tài liệu định nghĩa hệ thống, các tài liệu đặc tả hệ thống, đặc tả cơ bản cho 1 phiên bản mới sắp phát hành, hoặc bất kì bước nào khác trong quá trình làm yêu cầu. Ví dụ. Công ty đưa ra 1 tiêu chuẩn chung khi làm tài liệu, các kí hiệu, mô hình đối tượng, cần phải được thống nhất, Nhưng 1 nhân viên mới vào chưa nắm rõ được các tiêu chuẩn này nên làm sai 1 số chỗ. Phương pháp xem lại yêu cầu sẽ giúp tìm ra sai sót này giúp đồng nhất trong quá trình viết tài liệu.

1.6.2. Sử dụng phiên bản mẫu, thử nghiệm

Sử dụng phiên bản mẫu, thử nghiệm là dùng một mô hình chạy được của hệ thống để kiểm tra các yêu cầu. Ưu điểm của phương pháp này nhằm giúp dễ dàng trong việc giải thích những yêu cầu của phần mềm, giúp khách hàng có những phản hồi kịp thời để làm rõ hệ thống đang sai ở đâu, cũng như phát hiện ra những yêu cầu mới. Ví dụ Việc sử dụng mô hình giao diện người dùng (Mockup,...) giúp người lập trình cũng như khách hàng dễ hiểu, tiết kiệm hơn việc miêu tả đơn thuần dùng văn bản hoặc mô hình đồ họa. Sự biến động hay sự thay đổi yêu cầu sau khi dùng bản thử nghiệm là rất thấp bởi vì có sự thống nhất giữa người lập trình, khách hàng và các bên liên quan. Bên cạnh những ưu điểm đó, phương pháp dùng phiên bản mẫu cũng có một số nhược điểm như sau. Dùng phiên bản thử nghiệm làm phân tán sự tập trung của người dùng. Ví dụ, Phiên bản mẫu là phiên bản chưa hoàn thiện về mặt thẩm mỹ cũng như chức năng. Điều đó khiến người dùng nhầm tưởng rằng chất lượng sản phẩm có chất lượng không tốt. Dùng phiên bản mẫu có thể tốn kém hơn cho việc phát triển. Ví dụ, khách hàng quá tập trung vào chức năng của phiên bản mẫu sẽ dẫn đến lỗi phát sinh, người làm yêu cầu lại phải sửa lỗi đó. Do đó việc giải thích đây chỉ là bản mẫu, thử nghiệm là đặc biệt quan trọng. vì vậy sẽ tránh lãng phí nguồn nhân lực.

1.6.3. Thẩm định mô hình

Thẩm định model là thẩm định lại chất lượng các model(information model, behavior model, structure model) đã được phát triển trong suốt quá trình phân tích. Ví dụ trong mô hình đối tượng, chúng ta phải kiểm tra xem liên kết giữa các đối tượng, giữa sự trao đổi dữ liệu giữa các đối tượng có chuẩn xác không. Các model phải đủ các tiêu chí: Hoàn thiện, nhất quán và chuẩn xác.

1.6.4. Kiểm thử chấp thuận

Một đặc điểm quan trọng của yêu cầu phần mềm là nó có thể kiểm định rằng sản phẩm cuối cùng phải thỏa mãn các yêu cầu. Viết các testcase dành cho các yêu cầu để kiểm tra khả năng đáp ứng được các yêu cầu end-user. Sản phẩm cuối cùng phải thỏa mãn các testcase

1.7. Khảo sát hiện trạng

Thực trạng có rất nhiều thay đổi, do đó quản lí những thay đổi và duy trì những yêu cầu là chìa khóa quyết định sự thành công của phần mềm. Ví dụ : không phải tổ chức nào cũng có thói quen làm tài liệu cũng như quản lý yêu cầu đặc biệt với những công ty mới thành lập hoặc những công ty có nguồn nhân lực hạn chế. Khi những công ty này mở rộng, số lượng khách hàng trở lên lớn hơn, sản phẩm của họ bắt đầu phát triển, thì lúc này việc tìm lại những yêu cầu và thúc đẩy thêm nhiều tính năng để đáp ứng nhu cầu thị trường và sự thay đổi của môi trường là rất cần thiết. Do đó, tài liệu yêu cầu và quản lí những thay đổi là chìa khóa cho sự thành công của bất kì quá trình yêu cầu nào.

1.7.1. Quản lý thay đổi

Quản lý thay đổi là trung tâm của quản lý yêu cầu. Yêu cầu phần mềm luôn luôn thay đổi: +Môi trường doanh nghiệp và kĩ thuật thay đổi: Phần cứng mới => giao diện mới Ví dụ: Màn hình thay đổi, sắc nét hơn nên giao diện cũng phải chăm chút hơn. Luật thay đổi, nhu cầu doanh nghiệp thay đổi => thay đổi chức năng Ví dụ: Luật doanh nghiệp không cho tiết lộ thông tin người dùng. Do đó hệ thống quản lí user cần phải bảo mật hơn

- Khách hàng, người sử dụng thay đổi => Thay đổi chức năng Ví dụ: Khách hàng muốn tạo chức năng đặt lịch gửi mail. Do đó phần mềm cần phải có chức năng email schedual.
- Xung đột giữa các yêu cầu mới nảy sinh, và giữa yêu cầu mới với yêu cầu cũ => Quản lý thay đổi là trung tâm và đặc biệt quan trọng của quản lý các yêu cầu.

1.7.2. Định danh yêu cầu

- Yêu cầu bao gồm không chỉ đặc tả hệ thống mà còn những thông liên quan, giúp dễ dàng quản lí và diễn giải yêu cầu
- Định danh yêu cầu cần được định nghĩa, ghi nhận và cập nhật như phần phần mềm trong quá trình phát triển và bảo trì.

- Bao gồm:
 - việc phân loại yêu cầu Ví dụ: functional requirement and non-functional requirement
 - phương pháp xác minh Ví dụ: . xác minh bằng cách xem lại yêu cầu . xác minh bằng sử dụng phiên bản mẫu, thử nghiệm
 - thăm định mô hình
 - kiểm thử chấp thuận
 - kế hoạch kiểm thử
 - thuộc tính duy nhất để thuận tiện cho việc tham chiếu giữa các yêu cầu và lần vết. Ví dụ: thông tin của yêu cầu duy nhất để tiện cho việc thêm và chỉnh sửa sau khi có sự thay đổi.

1.7.3. Lần vết yêu cầu

- Lần vết yêu cầu có liên quan với việc khôi phục nguồn của yêu cầu và dự đoán những ảnh hưởng của các yêu cầu.
- Truy vết để thực hiện phân tích những ảnh hưởng khi yêu cầu thay đổi.
- một yêu cầu nên được truy về những yêu cầu khác và các bên liên quan để thúc đẩy nó(ví dụ: từ yêu cầu phần mềm về yêu cầu hệ thống)
- Ngược lại, một yêu cầu nên được truy đến những yêu cầu khác nhằm thỏa mãn nó(ví dụ: từ yêu cầu hệ thống đến yêu cầu phần mềm)

1.7.4. Đánh giá yêu cầu

- Đánh giá kích thước của sự thay đổi yêu cầu.
 - Ví dụ: đánh giá độ khó khăn của việc triển khai thay đổi yêu cầu, đánh giá xem nó ảnh hưởng tới những phân hệ nào trong hệ thống.
- Đánh giá chi phí cho việc phát triển hoặc duy trì 1 yêu cầu
 - Ví dụ: đánh giá nguồn lực, chi phí, mandate cho việc thay đổi yêu cầu.

1.8. Công cụ sử dụng trong việc làm yêu cầu phần mềm

- Công cụ cho việc vẽ model (CASE tool , starUML)
- Công cụ cho việc quản lí yêu cầu (spreadsheet, cơ sở dữ liệu,...)