



CHƯƠNG 2: PHÂN TÍCH YÊU CẦU PHẦN MỀM VÀ ĐẶC TẢ HỆ THỐNG

2.1 Các yêu cầu phần mềm

2.2. Quy trình xác định yêu cầu

2.3. Các phương pháp mô hình hóa



2.1 Các yêu cầu phần mềm

- ▶ Định nghĩa yêu cầu phần mềm
 - ▶ Yêu cầu chức năng (Functional Requirements)
 - ▶ Yêu cầu phi chức năng (Non-Functional Requirements)
 - ▶ Tầm quan trọng của việc xác định chính xác các yêu cầu
-



Định nghĩa yêu cầu phần mềm

- ▶ Yêu cầu phần mềm là các mô tả chi tiết về các chức năng, tính năng, và ràng buộc mà một hệ thống phần mềm cần phải đáp ứng. Chúng bao gồm tất cả các nhu cầu và mong đợi của người dùng cuối, khách hàng, và các bên liên quan khác. Yêu cầu phần mềm thường được chia thành hai loại :
 - ▶ Yêu cầu chức năng: Mô tả các hành vi và chức năng cụ thể mà hệ thống phải thực hiện. Ví dụ: hệ thống phải cho phép người dùng đăng nhập, tìm kiếm sản phẩm, hoặc gửi email.
 - ▶ Yêu cầu phi chức năng: Mô tả các tiêu chí không liên quan trực tiếp đến chức năng cụ thể, như hiệu suất, bảo mật, khả năng mở rộng, và tính dễ sử dụng.



Yêu cầu chức năng

- Yêu cầu chức năng là các mô tả chi tiết về những gì hệ thống phần mềm phải làm. Chúng xác định các hành vi, dịch vụ, và chức năng mà hệ thống cần cung cấp để đáp ứng nhu cầu của người dùng và các bên liên quan. Yêu cầu chức năng thường được thể hiện dưới dạng các câu lệnh mô tả các tác vụ mà hệ thống phải thực hiện.
-



Yêu cầu chức năng – ví dụ

- ▶ Hệ thống ngân hàng trực tuyến:
 - ▶ Người dùng phải có khả năng đăng nhập vào tài khoản của họ bằng tên người dùng và mật khẩu.
 - ▶ Hệ thống phải cho phép người dùng kiểm tra số dư tài khoản của họ.
 - ▶ Người dùng phải có thể chuyển tiền giữa các tài khoản khác nhau trong cùng một ngân hàng.
-



Yêu cầu chức năng – ví dụ

- ▶ Ứng dụng thương mại điện tử:
 - ▶ Hệ thống phải cho phép người dùng tìm kiếm sản phẩm theo tên, danh mục, hoặc giá.
 - ▶ Người dùng phải có thể thêm sản phẩm vào giỏ hàng và tiến hành thanh toán.
 - ▶ Hệ thống phải gửi email xác nhận đơn hàng cho người dùng sau khi thanh toán thành công.
-



Yêu cầu chức năng – ví dụ

- ▶ Ứng dụng quản lý học sinh:
 - ▶ Giáo viên phải có thể nhập điểm cho từng học sinh trong lớp.
 - ▶ Hệ thống phải tạo báo cáo tổng kết học kỳ cho từng học sinh.
 - ▶ Học sinh phải có thể xem lịch học và điểm số của mình.
-



Tầm quan trọng của việc mô tả các chức năng

- ▶ **Đảm Bảo Đáp Ứng Nhu Cầu Người Dùng:** Mô tả rõ ràng các chức năng giúp đảm bảo rằng hệ thống sẽ đáp ứng đúng nhu cầu và mong đợi của người dùng cuối.
 - ▶ **Hướng Dẫn Phát Triển:** Yêu cầu chức năng cung cấp hướng dẫn chi tiết cho nhóm phát triển về những gì cần được xây dựng, giúp họ tập trung vào việc phát triển các tính năng cần thiết.
 - ▶ **Cơ Sở Để Kiểm Thử:** Các yêu cầu chức năng là cơ sở để thiết kế các trường hợp kiểm thử, đảm bảo rằng hệ thống hoạt động đúng như mong đợi và không có lỗi.
-



Tầm quan trọng của việc mô tả các chức năng

- ▶ Giao Tiếp Giữa Các Bên Liên Quan: Mô tả chức năng rõ ràng giúp cải thiện giao tiếp giữa các bên liên quan, bao gồm khách hàng, nhà phát triển, và quản lý dự án, đảm bảo rằng tất cả mọi người đều có cùng một hiểu biết về những gì cần được phát triển.
 - ▶ Quản Lý Thay Đổi: Khi có yêu cầu thay đổi, các yêu cầu chức năng giúp đánh giá tác động của thay đổi đó đến hệ thống và xác định các điều chỉnh cần thiết
-



Yêu cầu phi chức năng

- Yêu cầu phi chức năng là các tiêu chí không liên quan trực tiếp đến các chức năng cụ thể mà hệ thống phải thực hiện, nhưng lại xác định cách thức hoạt động của hệ thống. Chúng bao gồm các thuộc tính về **chất lượng** của hệ thống như hiệu suất, bảo mật, khả năng bảo trì, và khả năng mở rộng. Yêu cầu phi chức năng thường ảnh hưởng đến **trải nghiệm** người dùng và hiệu quả hoạt động của hệ thống.
-



Yêu cầu phi chức năng – ví dụ

► Hiệu Suất (Performance):

- Hệ thống phải có khả năng xử lý 10.000 giao dịch mỗi giây.
 - Thời gian phản hồi cho bất kỳ yêu cầu nào của người dùng không được vượt quá 2 giây
-



Yêu cầu phi chức năng – ví dụ

► Bảo Mật (Security):

- Dữ liệu người dùng phải được mã hóa khi lưu trữ và truyền tải.
- Hệ thống phải yêu cầu xác thực hai yếu tố cho tất cả các giao dịch tài chính.



Yêu cầu phi chức năng – ví dụ

► Khả Năng Bảo Trì (Maintainability):

- Mã nguồn phải được viết theo các tiêu chuẩn mã hóa để dễ dàng bảo trì và nâng cấp.
 - Hệ thống phải cho phép cập nhật phần mềm mà không cần ngừng hoạt động.
-



Yêu cầu phi chức năng –ví dụ

► Khả Năng Mở Rộng (Scalability):

- Hệ thống phải có khả năng mở rộng để hỗ trợ thêm 100.000 người dùng mà không ảnh hưởng đến hiệu suất.
 - Cơ sở hạ tầng phải có khả năng mở rộng để xử lý gấp đôi lưu lượng truy cập hiện tại trong vòng 6 tháng.
-



Yêu cầu phi chức năng – ví dụ

► Tính Khả Dụng (Availability):

- Hệ thống phải hoạt động 24/7 với thời gian ngừng hoạt động không quá 1 giờ mỗi tháng.

► Tính Dễ Sử Dụng (Usability):

- Giao diện người dùng phải thân thiện và dễ sử dụng, cho phép người dùng hoàn thành các tác vụ chính trong không quá 3 bước.
-



Tầm quan trọng của việc xác định các yêu cầu phi chức năng

- ▶ Cải Thiện Trải Nghiệm Người Dùng: Yêu cầu phi chức năng đảm bảo rằng hệ thống không chỉ hoạt động đúng mà còn hoạt động tốt, mang lại trải nghiệm người dùng tích cực.
 - ▶ Đảm Bảo Chất Lượng Hệ Thống: Chúng giúp xác định các tiêu chuẩn chất lượng mà hệ thống phải đáp ứng, từ đó đảm bảo tính ổn định và tin cậy.
-



Tầm quan trọng của việc xác định các yêu cầu phi chức năng

- ▶ Hỗ Trợ Quản Lý Rủi Ro: Bằng cách xác định các yêu cầu về bảo mật và khả năng phục hồi, các yêu cầu phi chức năng giúp giảm thiểu rủi ro liên quan đến an ninh và sự cố hệ thống.
 - ▶ Hỗ Trợ Quản Lý Tài Nguyên: Yêu cầu về hiệu suất và khả năng mở rộng giúp quản lý tài nguyên hiệu quả, đảm bảo hệ thống có thể đáp ứng nhu cầu hiện tại và tương lai.
-



Tầm Quan Trọng của việc xác định chính xác yêu cầu

- ▶ Giảm Thiểu Rủi Ro: Xác định chính xác yêu cầu từ giai đoạn đầu giúp giảm thiểu rủi ro phát sinh các vấn đề trong quá trình phát triển. Nếu yêu cầu không rõ ràng hoặc không đầy đủ, có thể dẫn đến việc phát triển sai hướng, gây lãng phí thời gian và nguồn lực.
 - ▶ Tiết Kiệm Chi Phí: Sửa chữa lỗi hoặc thay đổi yêu cầu ở giai đoạn sau của dự án thường tốn kém hơn nhiều so với việc làm đúng ngay từ đầu. Việc xác định yêu cầu chính xác giúp tránh các chi phí phát sinh không cần thiết.
-



Tầm Quan Trọng của việc xác định chính xác yêu cầu

- ▶ Cải Thiện Chất Lượng Sản Phẩm: Khi yêu cầu được xác định rõ ràng, nhóm phát triển có thể tập trung vào việc xây dựng các tính năng đáp ứng đúng nhu cầu của người dùng, từ đó cải thiện chất lượng sản phẩm cuối cùng.
 - ▶ Tăng Cường Sự HÀi Lòng của Khách Hàng: Khi sản phẩm cuối cùng đáp ứng đúng và đầy đủ các yêu cầu của khách hàng, sự hài lòng của khách hàng sẽ tăng lên, điều này có thể dẫn đến mối quan hệ lâu dài và cơ hội kinh doanh mới.
-



Tầm Quan Trọng của việc xác định chính xác yêu cầu

- ▶ Quản Lý Dự Án Hiệu Quả Hơn: Yêu cầu rõ ràng giúp lập kế hoạch, ước lượng thời gian và nguồn lực cần thiết cho dự án một cách chính xác hơn. Điều này giúp quản lý dự án hiệu quả và đảm bảo tiến độ.
 - ▶ Giao Tiếp Tốt Hơn Giữa Các Bên Liên Quan: Tài liệu yêu cầu phần mềm là cơ sở để giao tiếp giữa các bên liên quan, bao gồm khách hàng, nhà phát triển, và quản lý dự án. Nó giúp đảm bảo rằng tất cả mọi người đều có cùng một hiểu biết về những gì cần được phát triển.
-



2.2 Quy trình xác định yêu cầu

- ▶ Quy trình xác định yêu cầu là một phần quan trọng trong phát triển phần mềm, nhằm **đảm bảo** rằng sản phẩm cuối cùng **đáp ứng đúng nhu cầu** và mong đợi của người dùng và các bên liên quan.
 - ▶ Quy trình này bao gồm việc thu thập, phân tích, đặc tả, xác nhận và quản lý các yêu cầu phần mềm.
-



2.2 Quy trình xác định yêu cầu

► Vai Trò của Các Bên Liên Quan:

- Khách hàng và người dùng cuối: Cung cấp thông tin về nhu cầu và mong đợi của họ đối với hệ thống.
 - Nhà phân tích hệ thống: Là cầu nối giữa khách hàng và nhóm phát triển, chịu trách nhiệm thu thập và phân tích yêu cầu.
 - Nhóm phát triển: Sử dụng tài liệu yêu cầu để thiết kế và xây dựng hệ thống.
 - Quản lý dự án: Đảm bảo rằng các yêu cầu được xác định và quản lý hiệu quả trong suốt vòng đời dự án.
-



Các giai đoạn xác định yêu cầu

- ▶ Thu thập yêu cầu: Tìm hiểu và ghi nhận các nhu cầu của người dùng.
 - ▶ Phân tích yêu cầu: Làm rõ và xác định tính khả thi của các yêu cầu.
 - ▶ Đặc tả yêu cầu: Tạo tài liệu chi tiết về các yêu cầu đã được xác định.
 - ▶ Xác nhận yêu cầu: Đảm bảo rằng các yêu cầu đã được xác định là chính xác và đầy đủ.
 - ▶ Quản lý yêu cầu: Duy trì và cập nhật các yêu cầu trong suốt vòng đời dự án.
-



Thu Thập Yêu Cầu (Requirements Elicitation)

- ▶ Mục tiêu: Hiểu rõ nhu cầu và mong đợi của các bên liên quan.
 - ▶ Phương pháp: Sử dụng các kỹ thuật như phỏng vấn, khảo sát, hội thảo, và quan sát người dùng.
 - ▶ Kết quả: Danh sách các yêu cầu ban đầu từ các bên liên quan.
-



Phân Tích Yêu Cầu (Requirements Analysis)

- ▶ Mục tiêu: Làm rõ, ưu tiên và xác định tính khả thi của các yêu cầu.
 - ▶ Hoạt động: Xác định các yêu cầu mâu thuẫn, phân loại yêu cầu thành chức năng và phi chức năng.
 - ▶ Kết quả: Tập hợp các yêu cầu đã được phân tích và sắp xếp theo thứ tự ưu tiên.
-



Đặc Tả Yêu Cầu (Requirements Specification)

- ▶ Mục tiêu: Tạo ra tài liệu yêu cầu chi tiết và rõ ràng.
 - ▶ Hoạt động: Viết tài liệu SRS (Software Requirements Specification) bao gồm các mô tả chi tiết về chức năng và phi chức năng.
 - ▶ Kết quả: Tài liệu SRS hoàn chỉnh, làm cơ sở cho thiết kế và phát triển phần mềm.
-



Xác Nhận Yêu Cầu (Requirements Validation)

- ▶ Mục tiêu: Đảm bảo rằng các yêu cầu đã được xác định là chính xác và đáp ứng nhu cầu của các bên liên quan.
 - ▶ Hoạt động: Xem xét tài liệu yêu cầu với các bên liên quan, thực hiện kiểm tra và đánh giá để phát hiện lỗi hoặc thiếu sót.
 - ▶ Kết quả: Các yêu cầu đã được xác nhận và phê duyệt bởi các bên liên quan.
-



Quản Lý Yêu Cầu (Requirements Management)

- ▶ Mục tiêu: Duy trì và quản lý các yêu cầu trong suốt vòng đời dự án.
 - ▶ Hoạt động: Theo dõi thay đổi yêu cầu, cập nhật tài liệu yêu cầu, và quản lý các yêu cầu mới phát sinh.
 - ▶ Kết quả: Các yêu cầu được quản lý hiệu quả, đảm bảo tính nhất quán và khả năng theo dõi.
-



2.3. Các phương pháp mô hình hóa

- ▶ Mô hình hóa giúp chuyển đổi các yêu cầu từ dạng **văn bản** sang các biểu diễn **trực quan**, giúp đảm bảo rằng các yêu cầu được hiểu rõ và thực hiện đúng đắn trong quá trình phát triển.
 - ▶ Lợi Ích của Mô Hình Hóa:
 - ▶ Trực quan hóa hệ thống: Mô hình hóa giúp biểu diễn các yêu cầu và cấu trúc hệ thống một cách trực quan, dễ hiểu hơn so với các mô tả bằng văn bản.
-



2.3. Các phương pháp mô hình hóa

- ▶ Lợi Ích của Mô Hình Hóa (tiếp):
 - ▶ Phát hiện sớm vấn đề: Mô hình hóa cho phép phát hiện sớm các vấn đề tiềm ẩn trong thiết kế và yêu cầu, giúp tiết kiệm thời gian và chi phí sửa chữa sau này.
 - ▶ Hỗ trợ thiết kế và phát triển: Các mô hình cung cấp cơ sở cho việc thiết kế và phát triển hệ thống, đảm bảo rằng các yêu cầu được thực hiện đúng đắn.
 - ▶ Quản lý phức tạp: Mô hình hóa giúp quản lý các hệ thống phức tạp bằng cách chia nhỏ chúng thành các phần dễ quản lý hơn.
-



2.3. Các phương pháp mô hình hóa

- ▶ Các phương pháp:
 - ▶ Mô Hình Hóa Chức Năng (Functional Modeling)
 - ▶ Mô Hình Hóa Dữ Liệu (Data Modeling)
 - ▶ Mô Hình Hóa Hành Vi (Behavioral Modeling)
 - ▶ Mô Hình Hóa Kiến Trúc (Architectural Modeling)
-



Mô Hình Hóa Chức Năng

- ▶ Sơ đồ Use Case (Use Case Diagrams): Biểu diễn các chức năng của hệ thống và các tương tác giữa người dùng và hệ thống. Sơ đồ use case giúp xác định các yêu cầu chức năng và các kịch bản sử dụng.
 - ▶ Sơ đồ Hoạt Động (Activity Diagrams): Mô tả luồng công việc hoặc các quy trình trong hệ thống, giúp hiểu rõ các bước thực hiện một chức năng cụ thể.
-



Mô Hình Hóa Dữ Liệu

- ▶ Sơ đồ Thực Thể - Quan Hệ (Entity-Relationship Diagrams - ERD): Biểu diễn các thực thể trong hệ thống và mối quan hệ giữa chúng. ERD giúp xác định cấu trúc dữ liệu và các ràng buộc.
 - ▶ Sơ đồ Lớp (Class Diagrams): Mô tả các lớp trong hệ thống và mối quan hệ giữa chúng, thường được sử dụng trong thiết kế hướng đối tượng.
-



Mô Hình Hóa Kiến Trúc

- ▶ Sơ đồ Thành Phần (Component Diagrams): Biểu diễn cấu trúc của hệ thống dưới dạng các thành phần và mối quan hệ giữa chúng.
 - ▶ Sơ đồ Triển Khai (Deployment Diagrams): Mô tả cách các thành phần phần mềm được triển khai trên phần cứng, giúp hiểu rõ cấu trúc vật lý của hệ thống.
-