

## CHƯƠNG 3

# KHẢO SÁT - PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU

Với sản phẩm phần mềm được xây dựng, việc hiểu đầy đủ các đặc điểm của nó là điều không dễ. Quá trình xác định các chức năng và các ràng buộc của hệ thống gọi là tìm hiểu và xác định yêu cầu. Để có được điều này thì cần phải trả lời câu hỏi "cái gì-what" chứ không phải là "như thế nào-how". Tìm hiểu, xác định và phân tích yêu cầu là bước hình thành bài toán, do vậy các yêu cầu của bài toán cần phải được tìm hiểu và phân tích theo chiều rộng (ngang) và theo chiều sâu (dọc).

### 3.1. TÌM HIỂU, XÁC ĐỊNH YÊU CẦU

#### 3.1.1. Khảo sát, tìm hiểu yêu cầu

Khi một công ty muốn ký một hợp đồng cho một dự án phát triển một phần mềm, công ty sẽ phát biểu các yêu cầu ở mức trừu tượng để không bắt buộc định nghĩa trước các giải pháp. Các yêu cầu phải được viết sao cho các nhà phát triển phần mềm có thể đưa ra các giải pháp khác nhau. Sau khi đã trúng thầu và ký hợp đồng, yêu cầu phải được làm rõ hơn để khách hàng có thể hiểu và đánh giá được phần mềm. Cả hai tài liệu nói trên đều gọi là tài liệu yêu cầu người dùng.

Theo mức độ chi tiết có thể chia ra các loại tài liệu yêu cầu:

- + Xác định yêu cầu: đây là một khẳng định, bằng ngôn ngữ tự nhiên hơn là các sơ đồ, về các dịch vụ hệ thống cần cung cấp và các ràng buộc mà hệ thống phải tuân theo. Tài liệu này cung cấp cho các thành phần: người quản lý của bên khách hàng, người dùng cuối của hệ thống, kỹ sư của khách hàng, người quản lý ký kết hợp đồng, các kiến trúc sư hệ thống.

- + Đặc tả yêu cầu: là tài liệu được cấu trúc mô tả hệ thống các dịch vụ chi tiết hơn. Đôi khi tài liệu này được gọi là đặc tả chức năng. Đây có thể coi là hợp đồng ký kết giữa người mua và kẻ bán phần mềm. Tài liệu này cung cấp cho các thành phần: người dùng cuối của hệ thống, kỹ sư của khách hàng, các kiến trúc sư hệ thống, người phát triển phần mềm.

- + Đặc tả phần mềm: là mô tả trừu tượng hơn của phần mềm làm cơ sở cho thiết kế và triển khai. Tài liệu này cung cấp cho các thành phần: kỹ sư của khách hàng, các kiến trúc sư hệ thống, người phát triển phần mềm.

*Xác định yêu cầu:* là mô tả trừu tượng các dịch vụ mà hệ thống được mong đợi phải cung cấp và các ràng buộc mà hệ thống phải tuân thủ khi vận hành. Nó chỉ có các đặc tả phẩm hạnh bên ngoài của hệ thống mà không liên quan đến các đặc tính thiết kế. Nó phải được viết sao cho người ta có thể hiểu được mà không cần một kiến thức chuyên môn đặc biệt nào.

Các yêu cầu của phần mềm được chia thành hai loại:

- 1) *Các yêu cầu hệ thống chức năng*: các dịch vụ mà hệ thống phải cung cấp.
- 2) *Các yêu cầu không chức năng*: các ràng buộc mà hệ thống phải tuân theo.

Về nguyên tắc các yêu cầu của một hệ thống phải là vừa đầy đủ, vừa tránh kiện. Đầy đủ có nghĩa là mọi yêu cầu đều phải được đặc tả. Tránh kiện có nghĩa là các yêu cầu không gây ra mâu thuẫn. Thực tế đối với các hệ lớn và phức tạp thì thực là không thể đạt được tính đầy đủ và tính tránh kiện cho phiên bản đầu của tư liệu yêu cầu phần mềm. Vấn đề là khi duyệt lại hoặc trong các pha sau này của vòng đời phần mềm, người ta phát hiện ra các sự không thỏa mãn đó thì tư liệu yêu cầu phải được chỉnh lý lại.

Về bản chất, chúng ta phải hiểu và xác định rõ những yêu cầu của khách hàng. Tuy nhiên, thường bài toán được khách hàng phát biểu bằng ngôn ngữ tự nhiên cộng thêm với việc dùng các bảng các biểu đồ cho các người dùng dễ hiểu (xem là người dùng không biết các khái niệm chuyên môn công nghệ thông tin). Không may là ngôn ngữ được dùng này lại thường là không chính xác và mơ hồ, đôi khi có sự lầm lẫn giữa các biểu thị khái niệm và các biểu thị chi tiết làm cho việc mô tả chứa các thông tin hỗ lớn được biểu diễn ở nhiều mức chi tiết khác nhau.

Ở đây, chúng ta cần chú ý rằng người đặt hàng có thể vì không hiểu biết gì về tin học nên họ không thể phát biểu chính xác và đầy đủ các yêu cầu của họ, đôi lúc thì những cái mà người sử dụng yêu cầu và những cái mà họ cần là không giống nhau. Thêm vào đó, chúng ta lại không hiểu biết đầy đủ về đối tượng, địa bàn cho nên không thể thu thập đầy đủ và chính xác các thông tin của đối tượng và đây chính là một trong những mâu thuẫn giữa khách hàng và chúng ta. Vì vậy, trong thực tế, đối với các hệ thống lớn và phức tạp, rất khó có thể đạt được tính đầy đủ và thống nhất của tài liệu yêu cầu.

Các yêu cầu được tìm hiểu còn chứa các mâu thuẫn:

- Thiếu rõ ràng: Rất khó sử dụng ngôn ngữ tự nhiên mô tả chính xác không nhầm lẫn mà không làm khó khăn cho người đọc.
- Nhầm lẫn yêu cầu: Các yêu cầu chức năng, các ràng buộc, mục đích của hệ thống và các thông tin thiết kế không được phân biệt rõ ràng.
- Trộn lẫn yêu cầu: Một số các yêu cầu khác nhau có thể được thể hiện như là một yêu cầu đơn.

Giải quyết mâu thuẫn này, chúng ta phải: trên cơ sở nghiên cứu kỹ lĩnh vực ứng dụng và thảo luận với người sử dụng để định nghĩa chính xác các yêu cầu của bài toán. Xác định rõ và đầy đủ bài toán là yếu tố quan trọng góp phần đảm bảo thành công của dự án. Nhiệm vụ của giai đoạn này là xây dựng được các hồ sơ mô tả chi tiết về các yêu cầu, nhiệm vụ, chức năng của hệ thống dự kiến.

### 3.1.2. Đánh giá các yêu cầu

Đánh giá các yêu cầu phần mềm liên quan với việc cho biết các yêu cầu thực sự định nghĩa hệ thống đáp ứng đòi hỏi của khách hàng. Nếu việc đánh giá này không chính xác, các lỗi trong phần yêu cầu sẽ truyền tới thiết kế hệ thống và triển khai hệ thống. Chi phí sửa chữa lỗi sẽ rất lớn. Sự thay đổi về yêu cầu ngụ ý rằng việc thiết kế

và triển khai cũng phải thay đổi theo. Một số khía cạnh của yêu cầu cần phải kiểm chứng:

- Giá trị: người dùng có thể nghĩ rằng hệ thống cần một số chức năng, tuy nhiên sau một số phân tích, có thể xác định các chức năng khác cần được đưa vào. Do hệ thống có nhiều loại người sử dụng nên có các yêu cầu khác nhau và không thể tránh khỏi sự thỏa hiệp các nhu cầu đó.
- Chắc chắn: mọi yêu cầu không được mâu thuẫn với các yêu cầu khác
- Hoàn chỉnh: định nghĩa cần phải bao gồm mọi chức năng và các ràng buộc
- Hiện thực: không có các yêu cầu đặc biệt đến mức phi hiện thực. Có thể dự đoán trước các phát triển phần cứng, tuy nhiên phát triển phần mềm thì khó dự đoán hơn.
- Mẫu: là một mô hình chạy được của hệ thống được trình bày với người sử dụng. Đây là một kỹ thuật đánh giá yêu cầu hiệu quả. Nó cho phép người dùng thử nghiệm với hệ thống. Việc đánh giá lại yêu cầu không nên được coi là công việc tiếp theo của tư liệu hóa yêu cầu sau khi đã hoàn thành. Các xem xét về yêu cầu định kỳ liên quan với người dùng và kỹ sư phần mềm luôn cần thiết.

Các xem xét yêu cầu có thể là hình thức hoặc phi hình thức. Xem xét phi hình thức liên quan việc các người ký hợp đồng thảo luận các yêu cầu với khách hàng. Nhiều vấn đề có thể được giải quyết dễ dàng bất ngờ khi tham khảo trực tiếp với khách hàng. Đối với yêu cầu xem xét chính thức, đội phát triển phải dẫn dắt khách hàng thông qua các yêu cầu hệ thống, giải thích các triển khai của mỗi yêu cầu. Nhóm rà soát phải kiểm tra mỗi yêu cầu về độ thống nhất, hoàn chỉnh cho toàn bộ tài liệu. Họ có thể phải kiểm tra:

- Có khả năng kiểm tra: tài liệu có thể kiểm tra thực tế được không?
- Khả năng hiểu biết: tài liệu có được khách hàng hiểu biết thấu đáo hay không?
- Lưu vết: nguồn gốc của tài liệu có được xác định rõ ràng hay không? Rất có thể phải quay lại nguồn gốc ban đầu để đánh giá ảnh hưởng của sự thay đổi.
- Tính thích hợp: các yêu cầu đã phù hợp hay chưa? Có thể thay đổi các yêu cầu mà không làm ảnh hưởng lớn đến toàn bộ hệ thống không.

### 3.2. PHÂN TÍCH YÊU CẦU

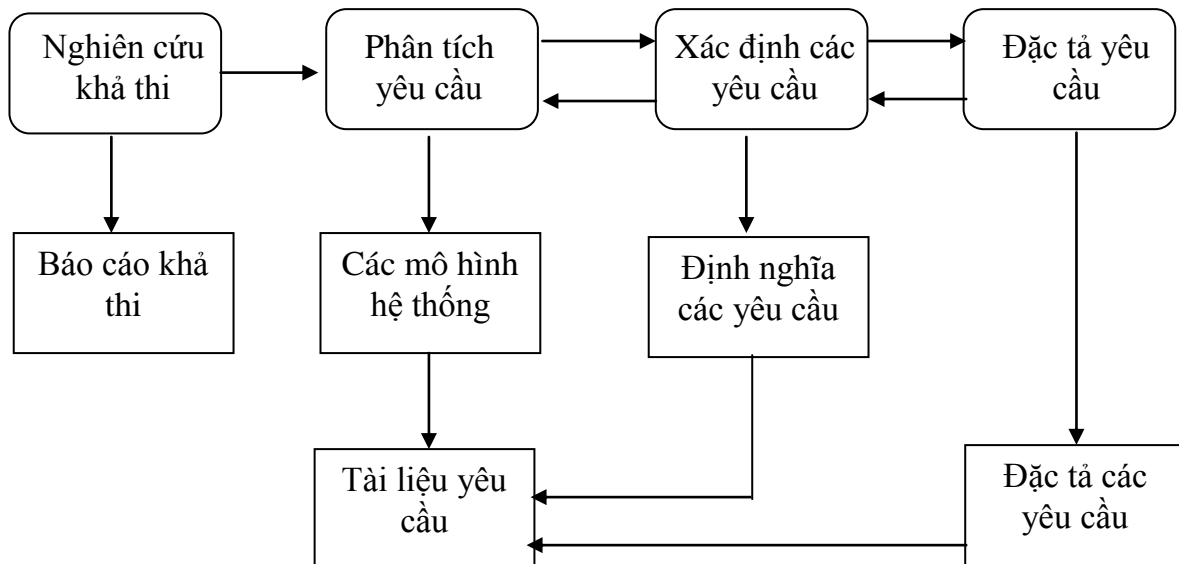
Nghiên cứu kỹ các yêu cầu của người sử dụng và của hệ thống phần mềm để xây dựng các đặc tả về hệ thống là cần thiết, nó sẽ xác định hành vi của hệ thống. Nhiệm vụ của giai đoạn này là phải trả lời được các câu hỏi sau:

- Đầu vào của hệ thống là gì
- Những quá trình cần xử lý trong hệ thống, hay hệ thống phần mềm sẽ phải xử lý những cái gì.
- Đầu ra: kết quả xử lý của hệ thống là gì
- Những ràng buộc trong hệ thống, chủ yếu là mối quan hệ giữa đầu vào và đầu ra như thế nào.

Trả lời được câu hỏi trên, nghĩa là phải xác định được chi tiết các yêu cầu làm cơ sở để đặc tả hệ thống. Đó là kết quả của sự trao đổi, thống nhất giữa người đầu tư, người sử dụng với người xây dựng hệ thống. Mục tiêu là xây dựng các hồ sơ mô tả chi

tiết các yêu cầu của bài toán nhằm nêu bật được hành vi, chức năng cần thực hiện của hệ thống dự kiến.

Như vậy, phân tích yêu cầu là quá trình suy luận các yêu cầu hệ thống thông qua quan sát hệ thống hiện tại, thảo luận với các người sử dụng, phân tích công việc. Việc này có thể liên quan với việc tạo một hay nhiều mô hình khác nhau. Nó giúp các phân tích viên hiểu biết hệ thống. Các mẫu hệ thống cũng có thể được phát triển để mô tả các yêu cầu. Ta có quy trình để có các chức năng của hệ thống:



Trong quá trình phân tích cần lưu ý đến tính khả thi của dự án.

+ Khả thi về kinh tế: chi phí phát triển phải cân xứng với lợi ích mà hệ thống đem lại, gồm có:

➤ Chi phí:

- Mua sắm: thiết bị, vật tư (phần cứng), tư vấn, cài đặt thiết bị, quản lý và phục vụ,...
- Chi phí cho khởi công: phần mềm phục vụ cho hệ thống, hệ thống liên lạc (truyền dữ liệu), nhân sự ban đầu: đào tạo - huấn luyện, cải tổ tổ chức cho phù hợp,...
- Chi phí liên quan: chi phí nhân công phục vụ thu nhập dữ liệu, sửa đổi, cập nhật hệ thống, chuẩn bị tài liệu,...
- Chi phí liên tục là tổn kém nhất, gồm: bảo trì, thuê bao, khấu hao phần cứng, chi phí phục vụ cho vận hành,...

➤ Lợi nhuận do sử dụng hệ thống

- Nhiệm vụ xử lý thông tin: giảm chi phí do xử lý tự động, tăng độ chính xác và kết quả tốt hơn, thời gian trả lời rút ngắn,...
- Có được từ hệ thống: thu thập và lưu trữ dữ liệu tự động, đầy đủ, dữ liệu được chuẩn hóa, bảo đảm an toàn và an ninh dữ liệu, tương thích và chuyển đổi giữa các bộ phận, truy cập và tìm kiếm nhanh, kết nối và trao đổi diện rộng,...

- + Khả thi về kỹ thuật: đây là vấn đề cần lưu ý vì các mục tiêu, chức năng và hiệu suất của hệ thống theo một cách nào đó là còn "mơ hồ" do vậy xét:
  - Rủi ro xây dựng: các phần tử hệ thống (chức năng, hiệu suất) khi thiết kế và phân tích có tương đương hay không?
  - Có sẵn tài nguyên: có sẵn con người và tài nguyên cần thiết để phát triển hệ thống?
  - Công nghệ: các công nghệ liên quan cho việc phát triển hệ thống đã có sẵn hay chưa?
- + Khả thi về hợp pháp: có sự xâm phạm, vi phạm hay khó khăn nào gây ra khi xây dựng hệ thống hay không.
- + Các phương án: đánh giá về phương án tiếp cận đến việc xây dựng hệ thống.

### 3.3. ĐẶC TẢ YÊU CẦU

#### 3.3.1. Đặc tả yêu cầu

Khi đã xác định rõ bài toán thì bước tiếp theo là tìm hiểu xem hệ thống dự kiến sẽ yêu cầu làm cái gì. Điều quan trọng ở đây là phải xây dựng được danh sách các yêu cầu của người sử dụng. Dựa trên những yêu cầu của người sử dụng, người phát triển đưa ra các đặc tả cho hệ thống.

Người xây dựng hệ thống phải trả lời được các yêu cầu sau đây:

- Đầu ra của hệ thống là cái gì
- Hệ thống sẽ phải làm cái gì để có kết quả mong muốn, nghĩa là phải xử lý những cái gì
- Những tài nguyên mà hệ thống yêu cầu là gì

Hiểu rõ nguồn gốc, các dạng thông tin cần cung cấp cho hệ thống hoạt động. Hệ thống sẽ phải giải quyết những vấn đề gì, những kết quả cần phải có là gì. Xác định được mối quan hệ giữa cái vào và cái ra cho quá trình hoạt động của hệ thống. Các đặc tả chi tiết phục vụ cho việc xây dựng và trải nghiệm về hệ thống để kiểm tra xem những nhiệm vụ đã đặt ra có hoàn tất được hay không.

Ở đây, chúng ta cần chú ý là trong một số trường hợp, sẽ nảy sinh những yêu cầu mới mà có thể là ta phải xây dựng lại hệ thống, tất nhiên điều này sẽ làm chậm tiến trình xây dựng và làm tăng giá thành do một vài lý do để không thể hoàn chỉnh các đặc tả đối với các hệ thống như:

- Các hệ thống phần mềm lớn luôn đòi hỏi cải tiến từ hiện trạng. Mặc dù các khó khăn của hệ thống hiện tại có thể xác định được nhưng các ảnh hưởng và hiệu ứng của hệ thống mới khó có thể dự đoán trước được.
- Hệ thống lớn thường có nhiều cộng đồng sử dụng khác nhau. Họ có các yêu cầu và ưu tiên khác nhau. Các yêu cầu hệ thống cuối cùng không tránh khỏi các thỏa hiệp.
- Người trả tiền cho hệ thống và người sử dụng thường khác nhau. Các yêu cầu đưa ra do ràng buộc của các tổ chức và tài chính có thể tranh chấp với yêu cầu của người sử dụng.

Do các đặc tả yêu cầu thêm các thông tin vào định nghĩa yêu cầu nên các đặc tả thường được biểu diễn cùng với các mô hình hệ thống được phát triển trong quá trình phân tích yêu cầu. Nó cần bao gồm mọi thông tin cần thiết về yêu cầu chức năng và ràng buộc của hệ thống. Phân tích yêu cầu được tiếp tục xác định và đặc tả khi các yêu cầu mới nảy sinh. Đây là tài liệu thường xuyên thay đổi và nên được kiểm soát chặt chẽ.

Ngôn ngữ tự nhiên không hoàn toàn thuận tiện cho các thiết kế viên hoặc các hợp đồng giữa các khách hàng và cán bộ phát triển hệ thống vì có một số lý do như sau:

- Nhầm lẫn do cách hiểu các khái niệm khác nhau giữa hai bên.
- Đặc tả yêu cầu ngôn ngữ tự nhiên quá mềm dẻo. Một vấn đề có thể được mô tả bằng quá nhiều cách khác nhau.
- Các yêu cầu không được phân hoạch tốt, khó tìm các mối quan hệ,...

Do vậy người ta thường dùng các thay thế khác để đặc tả các yêu cầu như:

- Ngôn ngữ tự nhiên có cấu trúc,
- Ngôn ngữ mô tả thiết kế, giống ngôn ngữ lập trình nhưng có mức trừu tượng cao hơn,
- Ngôn ngữ đặc tả yêu cầu,
- Ghi chép graphic,
- Đặc tả toán học,...

Có thể chia đặc tả yêu cầu ra làm hai loại: đặc tả phi hình thức (ngôn ngữ tự nhiên) và đặc tả hình thức (dựa trên kiến trúc toán học).

### *1. Đặc tả phi hình thức*

Đặc tả phi hình thức là đặc tả sử dụng ngôn ngữ tự nhiên. Tuy nó không được chặt chẽ bằng đặc tả hình thức nhưng được nhiều người biết và có thể dùng để trao đổi với nhau để làm chính xác hóa các điểm chưa rõ, chưa thống nhất giữa các bên phát triển hệ thống.

### *2. Đặc tả hình thức*

Đặc tả hình thức là đặc tả mà ở đó các từ ngữ, cú pháp, ngữ nghĩa được định nghĩa hình thức dựa vào toán học. Đặc tả hình thức có thể coi là một phần của hoạt động đặc tả phần mềm. Các đặc tả yêu cầu được phân tích chi tiết. Các mô tả trừu tượng của các chức năng chương trình có thể được tạo ra để làm rõ yêu cầu.

Đặc tả phần mềm hình thức là một đặc tả được trình bày trên một ngôn ngữ bao gồm: từ vựng, cú pháp và ngữ nghĩa được định nghĩa. Định nghĩa ngữ nghĩa đảm bảo ngôn ngữ đặc tả không phải là ngôn ngữ tự nhiên mà dựa trên toán học. Các chức năng nhận các đầu vào trả lại các kết quả. Các chức năng có thể định ra các điều kiện tiền tố và hậu tố. Điều kiện tiền tố là điều kiện cần thỏa mãn để có dữ liệu vào, điều kiện hậu tố là điều kiện cần thỏa mãn sau khi có kết quả.

Có hai hướng tiếp cận đặc tả hình thức để phát triển các hệ thống tương đối phức tạp.

- + Tiếp cận đại số, hệ thống được mô tả dưới dạng các toán tử và các quan hệ.
- + Tiếp cận mô hình, mô hình hệ thống được cấu trúc sử dụng các thực thể toán học như là các tập hợp và các thứ tự.

Sử dụng đặc tả hình thức, ta có các thuận lợi:

- Cho phép chúng ta thấy và hiểu được bản chất bên trong của các yêu cầu, đây là cách tốt nhất để làm giảm các lỗi, các thiếu sót có thể xảy ra và giúp cho công việc thiết kế được thuận lợi.
- Do chúng ta sử dụng toán học cho việc đặc tả nên có thể dựa vào các công cụ toán học khi phân tích và điều này làm tăng thêm tính chắc chắn và tính đầy đủ của hệ thống.
- Đặc tả hình thức, bản thân nó cho chúng ta một cách thức cho việc kiểm tra hệ thống sau này.

Tuy vậy, đặc tả hình thức cũng bộc lộ một vài khó khăn:

- Quản lý phần mềm có tính bảo thủ cố hữu của nó, không sẵn sàng chấp nhận các kỹ thuật mới.
- Chi phí cho việc đặc tả hình thức thường cao hơn so với các đặc tả khác (tuy phần cài đặt sẽ thấp hơn), nên khó để chứng minh rằng chi phí tương đối cao cho đặc tả sẽ làm giảm tổng chi phí dự án.
- Phần lớn, những người đặc tả hệ thống không được đào tạo một cách chính quy về việc sử dụng đặc tả hình thức cho việc đặc tả hệ thống mà dựa trên thói quen của họ.
- Thông thường, nhiều thành phần của hệ thống là khó cho việc đặc tả bằng ngôn ngữ hình thức. Thêm vào đó là khách hàng không thể hiểu được nó.
- Khách hàng không thích các đặc tả toán học.

### 3.3.2. Nguyên lý đặc tả

Nguyên lý 1: Phân tách chức năng với cài đặt: đặc tả là mô tả điều mong muốn chứ không phải cách thức thực hiện (cài đặt). Kết quả thu được theo dạng cái gì chứ không phải là thế nào.

Nguyên lý 2: Cần tới ngôn ngữ đặc tả hệ thống hướng tiến trình: cần thiết đặc biệt trong trường hợp môi trường là động và sự thay đổi của nó ảnh hưởng tới hành vi của thực thể nào đó tương tác với môi trường nào đó.

Nguyên lý 3: Đặc tả phải bao gồm hệ thống có phần mềm là một thành phần: vì hệ thống bao gồm các thành phần tương tác với nhau, chỉ bên trong hoàn cảnh của toàn bộ hệ thống và tương tác giữa các thành phần của nó thì hành vi của một thành phần mới có thể xác định.

Nguyên lý 4: Đặc tả phải bao gồm cả môi trường mà hệ thống vận hành.

Nguyên lý 5: Đặc tả hệ thống phải là một mô hình nhận thức: không phải là mô hình thiết kế hay cài đặt. Nó phải mô tả một hệ thống như cộng đồng người sử dụng cảm nhận thấy. (Các sự vật mà nó thao tác phải tương ứng với các sự vật của lĩnh vực

đó; các tác nhân phải mô hình cho các cá nhân, tổ chức, trang thiết bị; các hành động họ thực hiện thì mô hình cho những hoạt động thực tế xuất hiện trong lĩnh vực;...)

Nguyên lý 6: Đặc tả phải vận hành: phải đầy đủ và hình thức để có thể được dùng trong việc xác định liệu một cài đặt được đề nghị có thỏa mãn đặc tả trong những trường hợp kiểm thử tùy ý hay không.

Nguyên lý 7: Đặc tả hệ thống phải dung sai về tính không đầy đủ và tính nâng cao. Đặc tả không thể hoàn toàn đầy đủ do môi trường phức tạp

+ Đặc tả là mô hình - sự trừu tượng hóa - của tình huống thực nên không đầy đủ.

+ Đặc tả sẽ tồn tại ở nhiều mức chi tiết

+ Các công cụ phân tích được sử dụng để giúp cho đặc tả và kiểm thử đặc tả phải có khả năng xử lý với tính không đầy đủ.

Nguyên lý 8: Đặc tả phải được cục bộ hóa và được ghép lồng léo: Đặc tả làm cơ sở cho thiết kế và cài đặt, không phải tĩnh mà là một sự vật động, đang trải qua thay đổi đáng kể nên nội dung và cấu trúc phải phù hợp. Sự thay đổi khi cần sửa đổi là tối thiểu, chỉ một phần nhỏ các thành phần có thể thâm vào hay loại bớt một cách dễ dàng.

### 3.4. TƯ LIỆU HÓA YÊU CẦU PHẦN MỀM

Các yêu cầu hệ thống được trình bày trong tài liệu các yêu cầu phần mềm cho biết những thứ cần bộ phát triển hệ thống cần biết. Tài liệu này bao gồm các định nghĩa về yêu cầu và các đặc tả về các yêu cầu. Trong một số trường hợp, chúng không được trình bày riêng biệt mà được tích hợp làm một. Đôi khi, định nghĩa yêu cầu được trình bày như là một giới thiệu tới đặc tả yêu cầu. Cách tiếp cận hiệu quả nhất là trình bày các đặc tả chi tiết như là phụ lục của yêu cầu.

Tài liệu yêu cầu phần mềm không phải tài liệu đặc tả. Nó cần phải mô tả cái hệ thống cần phải làm chứ không phải làm thế nào. Tài liệu này cần dễ dàng được đặc tả và ánh xạ sang các phần tương ứng của thiết kế hệ thống. Nếu các dịch vụ, ràng buộc và các đặc tả thuộc tính trong tài liệu yêu cầu phần mềm được thỏa mãn bởi thiết kế thì thiết kế này được coi là giải pháp thích hợp với vấn đề.

Về nguyên tắc, các yêu cầu cần được hoàn chỉnh và chắc chắn. Mọi chức năng hệ thống cần được đặc tả và các yêu cầu không được mâu thuẫn. Tuy nhiên các thiếu sót là không thể tránh khỏi, do vậy tài liệu nên được cấu trúc để cho việc thay đổi. Nội dung nên được chia thành các chương. Sáu yêu cầu cần được thỏa mãn là:

- Nó cần mô tả các hành vi hệ thống bên ngoài
- Nó cần mô tả các ràng buộc về thực hiện
- Nó cần phải dễ thay đổi
- Nó phải là công cụ tham chiếu cho người bảo trì hệ thống
- Nó cần ghi được vòng đời của hệ thống
- Nó cần biểu thị được các đáp ứng chấp nhận được với các sự kiện không dự kiến



**Cấu trúc chung của tài liệu yêu cầu phần mềm gồm các phần như sau:**

+ Giới thiệu: mô tả sự cần thiết của hệ thống. Nó cần sự mô tả sơ lược các chức năng của mình và giải thích cách làm việc với các hệ thống khác. Nó cũng cần mô tả làm thế nào hệ thống đáp ứng được toàn bộ các mục tiêu chiến lược và nghiệp vụ.

+ Thuật ngữ: nó cần định nghĩa các khái niệm kỹ thuật được sử dụng trong tài liệu này. Không được giả định người đọc đã có kinh nghiệm.

+ Mô hình hệ thống: phần này lập một hoặc nhiều mô hình hệ thống cho biết các quan hệ giữa các cấu thành hệ thống với hệ thống và môi trường của nó. Nó cần bao gồm các mô hình đối tượng, mô hình luồng dữ liệu và ngữ nghĩa dữ liệu.

+ Định nghĩa yêu cầu chức năng: các dịch vụ cung cấp cho người dùng cần được mô tả trong mục này. Mô tả có thể dùng ngôn ngữ tự nhiên, sơ đồ hoặc các dạng ghi chép khác cho phép khách hàng có thể hiểu được.

Các dịch vụ cung cấp cho người dùng cần được mô tả trong mục này. Mô tả có thể dùng ngôn ngữ tự nhiên, sơ đồ hoặc các dạng ghi chép khác cho phép khách hàng có thể hiểu được.

+ Định nghĩa yêu cầu phi chức năng: các ràng buộc về phần mềm và các hạn chế đối với thiết kế cần phải được mô tả trong phần này. Nó có thể bao gồm các chi tiết của biểu diễn dữ liệu, thời gian đáp ứng và yêu cầu bộ nhớ,... Các tiêu chuẩn về sản phẩm và quy trình cần tuân thủ cũng được mô tả.

+ Tiến triển hệ thống: phần này mô tả các giả thiết căn bản làm cơ sở cho hệ thống và dự đoán các thay đổi về phát triển phần cứng, yêu cầu người dùng

+ Đặc tả yêu cầu: mô tả các yêu cầu cơ bản chi tiết hơn. Nếu cần các chi tiết hơn có thể được thêm vào các yêu cầu phi chức năng, ví dụ giao diện với các hệ thống có thể được định nghĩa.

- + Ngoài ra, tài liệu yêu cầu phần mềm có thể bao gồm thêm các phần sau:
  - Phần cứng: nếu hệ thống được phát triển trên một phần cứng đặc biệt, phần cứng này và giao diện cần được mô tả. Nếu phần cứng bán sẵn được sử dụng, các cấu hình cực tiểu và cực đại phải được mô tả.
  - Yêu cầu dữ liệu: tổ chức logic của dữ liệu được sử dụng bởi hệ thống và các quan hệ giữa chúng được mô tả, có thể dùng sơ đồ thực thể liên kết.
  - Chỉ mục có thể được cung cấp. Ví dụ chỉ mục theo chữ cái, chỉ mục theo chương, theo chức năng....

Do hệ thống được vận hành trong thời gian dài, nên môi trường hệ thống và mục đích nghiệp vụ có thể thay đổi. Khi đó tài liệu yêu cầu cũng cần phải thay đổi. Với mục đích tiến triển, tài liệu yêu cầu thường được chia theo hai phân loại:

- Các yêu cầu ổn định: được suy dẫn từ các hoạt động cốt lõi của tổ chức tương đối liên quan trực tiếp tới miền hệ thống.

- Các yêu cầu bất thường: các yêu cầu có thể thay đổi khi phát triển hệ thống sau này như: các yêu cầu xuất hiện như là sự hiểu biết của khách hàng về sự phát triển của hệ thống trong quá trình xây dựng hệ thống, các yêu cầu được sinh ra do sự xuất hiện của việc tin học hóa làm thay đổi các quy trình nghiệp vụ,...

### 3.5. ĐẶC TÍNH DỮ LIỆU VÀ CÁC KỸ THUẬT ĐỂ THU THẬP DỮ LIỆU

#### 3.5.1. Đặc tính dữ liệu

Một ứng dụng thành công là một ứng dụng đáp ứng được đầy đủ các yêu cầu của người sử dụng. Trong quá trình xác định yêu cầu, các dữ liệu thu được của bài toán chứa một số tính chất mà ta gọi là đặc tính dữ liệu như:

- Tính định hướng thời gian,
- Tính cấu trúc,
- Tính đầy đủ,
- Nhập nhằng,
- Ngữ nghĩa,
- Độ lớn của dữ liệu,...

Mỗi yếu tố trên đều quan trọng trong việc xác định các đặc tả của ứng dụng bởi vì chúng mang đến các chỉ dẫn cho kỹ sư phần mềm biết số lượng và kiểu thông tin nên được chọn. Cũng vậy, các kiểu dữ liệu khác nhau có liên quan tới các loại ứng dụng khác nhau và đòi hỏi các kỹ thuật khai thác thông tin khác nhau. Không chú ý tới các đặc tính dữ liệu sẽ gây lỗi phân tích và thiết kế. Mỗi liên hệ giữa các kiểu ứng dụng và các đặc tính dữ liệu của nó được thể hiện ở bảng sau:

<b>Đặc tính dữ liệu Loại ứng dụng</b>	<b>Tính thời gian</b>	<b>Cấu trúc</b>	<b>Mức đầy đủ</b>	<b>Nhập nhằng</b>	<b>Ngữ nghĩa</b>	<b>Độ lớn</b>
<b>TPS</b>	Hiện tại	Cấu trúc	Đầy đủ	Thấp	Cố định	Các loại
<b>CSDL</b>	Quá khứ hiện tại	Cấu trúc	Đầy đủ	Thấp	Cố định	Các loại
<b>DSS</b>	Các loại	Cấu trúc	Thay đổi	Thấp, trung bình	Thay đổi	Trung bình cao
<b>GDSS</b>	Hiện tại tương lai	Phi cấu trúc	Không đầy đủ	Trung bình cao	Thay đổi	Thấp
<b>EIS</b>	Tương lai	Phi cấu trúc	Không đầy đủ	Trung bình cao	Thay đổi	Thấp trung bình
<b>ES</b>	Hiện tại (dựa quá khứ)	Bán cấu trúc	Không đầy đủ	Trung bình cao	Thay đổi	Thấp

Hệ xử lý giao dịch bao gồm kiến thức định trước, thông tin đầy đủ, có cấu trúc, hiện thời. Do hệ xử lý giao dịch là các ứng dụng thao tác của công ty nên để điều khiển và bảo trì các bản ghi của thao tác hiện thời, bạn phải có thông tin đầy đủ, hiện thời.

Các ứng dụng hỏi đáp có đặc tính tương tự hệ xử lý giao dịch với đặc điểm khác mà chúng có thể tập trung vào các thông tin lịch sử thêm vào thông tin hiện tại. Truy vấn là các câu hỏi được đặt ra bởi dữ liệu để tìm thấy các vấn đề và giải pháp, để phân tích, tổng kết và báo cáo trên dữ liệu. Để tạo tổng kết và báo cáo với sự tin tưởng, dữ liệu phải có cấu trúc, đầy đủ và được diễn giải không nhầm lẫn và có ngữ nghĩa nhất định.

Hệ hỗ trợ quyết định là các công cụ phân tích thống kê cho phép phát triển các thông tin giúp đỡ việc ra quyết định. Kiểu dữ liệu xác định hệ hỗ trợ quyết định luôn có thể được biểu diễn lại, có thể chưa hoàn chỉnh, nhập nhằng, có ngữ nghĩa thay đổi từ trung bình tới nhiều về độ lớn.

Hệ hỗ trợ quyết định theo nhóm là công cụ hỗ trợ họp nhóm cho nhóm người. Các công cụ hệ hỗ trợ quyết định theo nhóm thao tác có cấu trúc trên đầy đủ và còn các nhập nhằng về ngữ nghĩa. Bản thân các công cụ thì đầy đủ, không nhập nhằng và mạnh nhưng các thông tin họp nhóm mà nó thực hiện thì lại không như vậy.

Hệ thông tin điều hành là các ứng dụng hướng tương lai cho phép duyệt qua môi trường và xác định khuynh hướng, cơ hội kinh doanh, hoặc các hoạt động công nghiệp khác ảnh hưởng tới hoạt động của công ty. Hệ thông tin điều hành giải quyết phần lớn với các dữ liệu “hỗn độn” không có cấu trúc, không đầy đủ, nhập nhằng, và chứa ngữ nghĩa thay đổi.

Hệ chuyên gia quản lý và suy luận thông qua các dữ liệu bán cấu trúc, không đầy đủ, nhập nhằng và ngữ nghĩa thay đổi. Các chuyên gia lấy các thông tin ngẫu nhiên và không cấu trúc sau đó tạo cấu trúc cho nó. Họ suy luận bằng cách làm thế nào diễn đạt dữ liệu để loại trừ mức độ nhập nhằng và cố định ngữ nghĩa. Do đó, mặc dù các dữ liệu đầu vào ứng dụng có các đặc tính mờ, quá trình xử lý dữ liệu phải thực sự được cấu trúc cao.

### ***3.5.1.1. Tính định hướng thời gian***

Tính hướng thời gian của dữ liệu đề cập tới quá khứ, hiện tại hoặc các đòi hỏi tương lai của ứng dụng đã đề ra.

Các dữ liệu quá khứ: có thể mô tả công việc đã được biến đổi thế nào qua thời gian, các quy định ảnh hưởng thế nào tới nhiệm vụ, vị trí của nó trong tổ chức và nhiệm vụ. Các thông tin quá khứ là chính xác, đầy đủ và xác đáng.

Các thông tin hiện tại: là các thông tin về cái gì đang xảy ra. Ví dụ, thông tin ứng dụng hiện tại liên quan tới quá trình hoạt động của công ty, số lượng của các lệnh được thực hiện trong ngày hoặc số lượng các hàng hoá được sản xuất, các chính sách, sản phẩm, đòi hỏi nghiệp vụ, yêu cầu pháp quy hiện tại hoặc các ràng buộc khác cũng rất cần thiết cho phát triển ứng dụng. Các thông tin hiện tại nên được tự động hoá theo

cách thích hợp với đội ngũ phát triển để tăng trí thức của họ về ứng dụng và phạm vi bài toán.

Các đòi hỏi trong tương lai: liên quan tới các sự thay đổi sẽ xảy ra, chúng không chính xác và rất khó kiểm tra. Ví dụ: các dự đoán kinh tế, khuynh hướng tiếp thị, bán hàng,...

#### **3.5.1.2. Tính cấu trúc**

Cấu trúc của thông tin định hướng về phần mở rộng theo đó thông tin có thể được phân loại theo cách nào đó. Cấu trúc có thể tham chiếu tới các hàm, môi trường hoặc dạng dữ liệu hay dạng xử lý. Các thông tin thay đổi từ phi cấu trúc cho tới cấu trúc mà phần cấu trúc được xác định bởi kỹ sư phần mềm. Cấu trúc là đặc biệt quan trọng bởi vì thiếu nó ta có thể tạo ứng dụng sai.

#### **3.5.1.3. Tính đầy đủ**

Tính đầy đủ thể hiện ở chỗ các thông tin cần thiết phải được biểu diễn. Một kiểu ứng dụng đòi hỏi một mức độ đầy đủ khác nhau. Các hệ thống xử lý giao dịch luôn tiếp cận các thông tin đầy đủ và chính xác, trong khi các hệ hỗ trợ quyết định đòi hỏi thông tin ít đầy đủ hơn. Các hệ thông tin điều hành, hệ chuyên gia, hoặc là các ứng dụng trí tuệ nhân tạo có mức độ cao nhất về tính không đầy đủ trong phạm vi của ứng dụng.

Đối với các ứng dụng phải giải quyết các thông tin không đầy đủ, một thách thức đối với nhóm phát triển là phải quyết định thông tin đã đủ để sử dụng hay chưa. Đôi khi quyết định này được tiến hành từ phía người dùng, đôi khi nó được tiến hành bên trong ứng dụng và cần phải có luật để xác định mức độ đầy đủ.

#### **3.5.1.4. Nhập nhằng**

Tính nhập nhằng là một thuộc tính của dữ liệu, thể hiện ở chỗ không trong sáng về nghĩa hoặc có nhiều nghĩa một cách hữu ý. Tính này liên quan nhiều đến mức độ ngữ nghĩa. Vấn đề này nảy sinh khi gặp một vấn đề có thể được hiểu theo nhiều cách - ví dụ câu phát biểu: "Ông cụ già đi mau quá!". Để giải quyết tính nhập nhằng cần căn cứ vào ngữ cảnh.

#### **3.5.1.5. Ngữ nghĩa**

Ngữ nghĩa là một tập hợp các định nghĩa được chia sẻ cho biết các thuật ngữ, chính sách hoặc các hành động được hiểu như thế nào cho mọi người trong một tổ chức nào đó.

Ngữ nghĩa rất quan trọng trong phát triển ứng dụng và đối với bản thân ứng dụng. Nếu mọi người dùng chung một thuật ngữ nhưng có quan niệm khác nhau sẽ xuất hiện sự không hiểu và không trao đổi thông tin được. Đối với bản thân ứng dụng nếu dữ liệu bị nhập nhằng về ý nghĩa có thể sẽ không bao giờ được xử lý cho đến khi người sử dụng hiểu được ý nghĩa của dữ liệu. Các ứng dụng sẽ có ngữ nghĩa cố định với các mục dữ liệu được định tính thông qua việc đào tạo và quá trình sử dụng lâu

dài. Khi đánh mất ngữ nghĩa của thông tin có thể gây tổn thất rất lớn đối với các bên liên quan.

#### **3.5.1.6. Độ lớn của dữ liệu**

Độ lớn của dữ liệu là số lượng các sự kiện nghiệp vụ hệ thống phải tiến hành trong vài chu kỳ nào đó. Độ lớn của tạo mới hoặc thay đổi khách hàng được tiến hành theo tháng hoặc năm, trong khi độ lớn của giao dịch nghiệp vụ được tiến hành theo ngày hoặc giờ và độ lớn tối đa. Độ lớn tối đa là số lượng các giao dịch hoặc các sự kiện nghiệp vụ được xử lý trong thời kỳ bận nhất. Thời kỳ cao điểm có thể theo năm hoặc cuối vài tháng, ví dụ chuẩn bị cho báo cáo nộp thuế. Độ lớn của dữ liệu là một nguồn thông tin phức tạp bởi vì số lượng thời gian cần thiết xử lý một giao dịch đơn có thể trở thành rất quan trọng đối với lượng lớn dữ liệu cần xử lý.

#### **3.5.2. Các kỹ thuật để thu thập dữ liệu cho ứng dụng**

Do mỗi giai đoạn phát triển hệ thống đều đòi hỏi có sự trao đổi giữa nhà phát triển và người dùng để nhận được thông tin có ích. Mỗi giai đoạn cần tìm kiếm một dải rộng các câu hỏi về ứng dụng. Khi phân tích tính khả thi, các câu hỏi tương đối rộng và tổng quát như: đâu là phạm vi của vấn đề, cách tốt nhất để tự động hoá là gì, công ty có cố gắng phát triển ứng dụng này không, công ty có thể hỗ trợ việc phát triển ứng dụng không?...thì khi phân tích yêu cầu chúng ta tìm hiểu các thông tin liên quan ứng dụng là gì, như: các dữ liệu cần thiết là gì, các xử lý nào được tiến hành và các thông tin chi tiết liên quan?... Như vậy, làm thế nào để chúng ta thu thập đầy đủ các thông tin để phục vụ cho phát triển ứng dụng?

##### **3.5.2.1. Một số kỹ thuật để thu thập dữ liệu**

###### *1. Phỏng vấn*

Phỏng vấn là việc tập hợp một số lượng ít người - thường với một hoặc hai người - cho một thời gian cố định với một mục đích cụ thể. Trong quá trình phỏng vấn, các câu hỏi có thể được thay đổi. Bạn có thể đánh giá được cảm nhận của họ, động cơ và thói quen đối với các phòng ban, quá trình quản lý, ứng dụng hoặc các thực thể khác đáng chú ý. Kiểu phỏng vấn được xác định bởi kiểu thông tin yêu cầu. Phỏng vấn nên được dẫn dắt sao cho cả hai bên tham gia đều cảm thấy thoải mái với kết quả. Được chuẩn bị kỹ đồng nghĩa là hiểu rõ người đang được phỏng vấn, do đó bạn không làm họ bối rối và bạn có thể có vài câu hỏi ban đầu được chuẩn bị cho dù không phải là tất cả.

Một phỏng vấn tốt có bắt đầu, đoạn giữa và kết thúc:

- Lúc bắt đầu, bạn tự giới thiệu và đặt các câu hỏi đơn giản. Nên bắt đầu với các câu hỏi tổng quát và không đòi hỏi các trả lời về động cơ cá nhân. Hãy chú ý tới các câu trả lời để tìm ra đâu mới các câu hỏi khác và tính trung thực, thái độ của người được phỏng vấn.

- Vào giữa buổi nên tập trung vào chủ đề. Hãy lấy mọi thông tin bạn cần lưu ý, sử dụng các kỹ thuật các bạn đã chọn ban đầu. Nếu xuất hiện một vài thông tin quan trọng, hãy hỏi xem bạn có thể được thảo luận nó sau này.
- Vào lúc kết thúc, hãy tóm tắt tất cả các thứ bạn đã nghe và nói cho các phòng vẫn cái gì sẽ xảy ra tiếp. Bạn có thể ghi chép và đề nghị người được hỏi xem xét lại. Tốt nhất là trong thời gian khoảng 48 giờ và có sự chấp thuận của người dùng theo ngày xác định.

Phòng vẫn sử dụng hai kiểu câu hỏi:

- Câu hỏi mở là câu hỏi cho nhiều câu trả lời khác nhau. Câu hỏi mở thích hợp cho các mô tả các chức năng ứng dụng hiện tại cũng như đang đề nghị và cho việc xác định cảm nhận ý kiến, và mong đợi về ứng dụng được đề ra. Một ví dụ là: “ Ông có thể nói cho tôi về...” “Ông nghĩ thế nào về...” “Ông có thể mô tả làm thế nào...”
- Câu hỏi đóng là câu hỏi mà câu trả lời là “có” hoặc “không” hoặc một câu trả lời cụ thể. Các câu hỏi đóng tốt cho khai thác thông tin thực tế hoặc bắt người dùng tập trung vào phỏng vấn. Ví dụ cho câu hỏi đóng là: “Bạn có dùng các báo cáo hàng tháng không”. Các câu trả lời “Có” có thể được tiếp nối bởi các câu hỏi mở “Ông có thể giải thích...”

### ***Các bước thường được tiến hành cho cuộc phỏng vấn***

⇒ Tiến hành đặt cuộc hẹn phù hợp với thời gian của phỏng vấn

⇒ Chuẩn bị tốt, tìm hiểu kỹ về người được phỏng vấn

⇒ Đúng giờ

⇒ Có kế hoạch cho cuộc phỏng vấn

- Giới thiệu bản thân và mục đích
- Sử dụng câu hỏi mở để bắt đầu
- Luôn chú ý vào trả lời
- Có kế hoạch cho nội dung chính
- Kết hợp câu hỏi đóng và mở
- Luôn bám sát các trình bày và phát triển chi tiết
- Luôn cung cấp các thông tin phản hồi, ví dụ: “ Cho phép tôi trình bày lại điều ông vừa nói...”
- Hạn chế ghi chép nếu thấy không tiện
- Có kế hoạch kết thúc
- Tóm tắt nội dung, yêu cầu hiệu chỉnh
- Yêu cầu làm chính xác, đánh giá lại ghi chép
- Cho biết ngày tháng họ sẽ nhận được báo cáo
- Thống nhất lại ngày lấy lại bản hiệu chỉnh
- Xác nhận lại lịch làm việc

Các câu hỏi có thể được đưa ra theo kiểu câu trúc hoặc phi cấu trúc.

- Phỏng vấn có cấu trúc là phỏng vấn trong đó người phỏng vấn đã có danh mục các mục cần duyệt qua, các câu hỏi xác định và các thông tin cần biết xác định.

- Phỏng vấn không cấu trúc là phỏng vấn được định hướng bởi câu trả lời. Các câu hỏi phần lớn là câu hỏi mở. Không có một kế hoạch ban đầu, do vậy người phỏng vấn biết các thông tin cần thiết sẽ dùng các câu trả lời từ các câu hỏi mở để phát triển mọi câu hỏi chi tiết hơn về chủ đề.

Phỏng vấn có cấu trúc thích hợp khi bạn biết về các thông tin cần thiết trước khi phỏng vấn. Ngược lại phỏng vấn không cấu trúc thích hợp khi bạn không đoán trước được chủ đề.

Các trường hợp điển hình của phỏng vấn là người khách hàng bắt đầu với phỏng vấn phi cấu trúc để cho bạn nhận thức về miền bài toán. Sau đó bài toán dần dần trở thành có cấu trúc và tập trung vào các thông tin bạn cần để hoàn chỉnh phân tích.

Các thuận lợi và khó khăn của phỏng vấn có cấu trúc và phỏng vấn không có cấu trúc được thể hiện như sau:

	<b>Phỏng vấn có cấu trúc</b>	<b>Phỏng vấn không có cấu trúc</b>
<b>Ưu điểm</b>	<ul style="list-style-type: none"> <li>- Dùng dạng chuẩn cho nhiều câu hỏi</li> <li>- Dễ quản lý và đánh giá</li> <li>- Đánh giá được nhiều mục đích</li> <li>- Không cần đào tạo nhiều</li> <li>- Có kết quả trong các phỏng vấn</li> </ul>	<ul style="list-style-type: none"> <li>- Có khả năng mềm dẻo nhất</li> <li>- Cần chăm chú nghe và có kỹ năng mở rộng câu hỏi</li> <li>- Có thể bao hết các thông tin chưa biết</li> <li>- Đòi hỏi có thực hành</li> </ul>
<b>Nhược điểm</b>	<ul style="list-style-type: none"> <li>- Chi phí chuẩn bị lớn</li> <li>- Tính cấu trúc có thể không thích hợp cho mọi tình huống</li> <li>- Giảm tính chủ động của các người phỏng vấn</li> </ul>	<ul style="list-style-type: none"> <li>- Lãng phí thời gian phỏng vấn</li> <li>- Người phỏng vấn có thể định kiến với các câu hỏi</li> <li>- Tốn thời gian lựa chọn và phân tích thông tin</li> </ul>

Một kỹ năng tốt là phát triển các sơ đồ như là một phần của tài liệu phỏng vấn. Khi bắt đầu lần phỏng vấn mới, bạn nên bàn bạc các sơ đồ và đưa cho họ bản ghi chép để họ có thể kiểm tra sau này. Bạn nhận ngay ý kiến phản hồi về tính chính xác của sơ đồ và hiểu biết của bạn về ứng dụng. Lợi ích của cách tiếp cận này thể hiện cả mặt kỹ năng và tâm lý. Từ khía cạnh kỹ thuật, bạn thường xuyên được kiểm tra lại các vấn đề bạn được nghe. Cho tới khi thời gian phân tích kết thúc, cả bạn và khách hàng đều tin chắc rằng quá trình xử lý ứng dụng là đúng và đầy đủ. Từ khía cạnh tâm lý, bạn làm tăng niềm tin của khách hàng vào khả năng phân tích bằng cách trình bày các hiểu biết của mình. Mỗi khi bạn cải thiện sơ đồ và đi sâu vào phân tích, bạn cũng làm tăng niềm tin của người sử dụng rằng bạn sẽ xây dựng được ứng dụng đáp ứng được nhu cầu của họ.

Các kết quả sử dụng người phỏng vấn nên luôn được trao đổi lại tới người được phỏng vấn trong một thời gian ngắn. Các người được phỏng vấn nên được báo cho một thời hạn đối với việc phỏng vấn. Tuy nhiên bạn có thể xin bố trí bổ sung phỏng vấn trong trường hợp còn nhiều điều cần hỏi hoặc còn nhiều người cần gặp.

Ưu điểm của phương pháp phỏng vấn:

- Phỏng vấn thích hợp cho việc nhận các thông tin đảm bảo cả số lượng lẫn chất lượng:
- Các kiểu thông tin định tính là các ý kiến, niềm tin, thói quen, chính sách, và mô tả.
- Các kiểu về thông tin định lượng bao gồm tần suất, số lượng, định lượng các mục được dùng trong ứng dụng.

Nhược điểm của phương pháp phỏng vấn:

- Phỏng vấn và các dạng khác của thu thập dữ liệu có thể làm bạn lạc lối, thiếu chính xác, hoặc thông tin không liên quan, không thích hợp.
- Bạn cần học cách đọc ngôn ngữ cử chỉ và thói quen để quyết định được các điều cần thiết cho cùng một thông tin.
- Đòi hỏi kỹ năng
- Có thể có kết quả thiên vị
- Đòi hỏi ba người để kiểm tra kết quả và thường không thích hợp với số lượng lớn người

Do vậy, khi sử dụng phương pháp phỏng vấn để thu thập dữ liệu, người phỏng vấn phải hiểu các hành động của người được phỏng vấn để có thể có các hành động tương ứng cho phù hợp. Sau đây là bảng tham khảo các hành vi của người được phỏng vấn và hoạt động tương ứng của người phỏng vấn.

Hành vi của người được phỏng vấn	Đáp ứng của người phỏng vấn
⇒ Đoán các câu trả lời chứ không thừa nhận không biết	⇒ Sau phỏng vấn, kiểm tra chéo các câu trả lời
⇒ Cố nói những điều lọt tai người phỏng vấn, sai sự thật	⇒ Tránh hỏi các câu dễ đoán câu trả lời, kiểm tra chéo các câu trả lời
⇒ Cho thông tin không đầy đủ	⇒ Kiên trì hỏi để đạt mục đích
⇒ Dừng trình bày khi người phỏng vấn ghi chép	⇒ Ghi nhanh nhất có thể, chỉ hỏi các câu quan trọng
⇒ Vội vã	⇒ Đề nghị bố trí buổi khác
⇒ Không muốn thay đổi môi trường hiện tại	⇒ Động viên cải thiện môi trường hiện tại và so sánh hai khuynh hướng
⇒ Thể hiện không quan tâm, trả lời ngắt quãng	⇒ Nói chuyện vui sau đó hướng về chủ đề chính
⇒ Không hợp tác, từ chối trả lời	⇒ Lấy nguồn tin khác, và hỏi “Ông có quan tâm về những điều người khác nói về ông hay không?”. Nếu câu trả lời là không thì thôi phỏng vấn
⇒ Phàn nàn về vị trí công tác, lương,...	⇒ Tìm ra mấu chốt vấn đề. Cố dẫn dắt về chủ đề chính. Ví dụ “Dường như cơ quan ông có rất nhiều vấn đề, có thể ứng dụng mới được đề xuất sẽ giải quyết mọi vấn đề trên”
⇒ Là người ưa chuộng công nghệ	⇒ Chọn lọc các thông tin cần thiết, không để lời cuốn vào các chiến lược công nghệ



## 2. *Họp nhóm*

Họp nhóm là việc tập hợp ba hoặc nhiều hơn một số người cho một thời hạn nhất định để thảo luận một số chủ đề. Họp nhóm có thể vừa bổ sung và thay thế phỏng vấn. Nó bổ sung phỏng vấn bằng cách cho phép nhóm kiểm tra lại các kết quả phỏng vấn cá nhân. Nó có thể thay thế cho phỏng vấn bằng cách cung cấp một diễn đàn cho các người sử dụng cùng tìm ra các đòi hỏi và các giải pháp cho ứng dụng.

Họp nhóm cũng có thể làm lãng phí thời gian. Nói chung, họp nhóm càng lớn thì càng ít ý kiến nhất trí và thời gian quyết định càng kéo dài. Do vậy, nên có một kế hoạch ban đầu cho họp nhóm. Lịch trình nên được cung cấp trước cho các thành viên. Số lượng các chủ đề nên giữ ở mức độ 1 đến 5 và không mời những người không thích họp để tránh lãng phí thời gian. Họp nhóm nên có một thời gian cố định và có điểm thống nhất cụ thể với các quyết định cần thiết, không nên kéo dài hơn 2 giờ để có thể đảm bảo được sự chú ý của thành viên. Họp nhóm phù hợp với việc đòi hỏi các tổng quát, tìm kiếm đồng lòng, và tiếp nhận cả thông tin chi tiết và thông tin tóm tắt.

Vậy, ưu điểm của phương pháp họp nhóm:

- Có thể tạo ra quyết định,
- Nhận được cả thông tin tổng hợp và chi tiết,
- Phương pháp tốt cho các yêu cầu bên ngoài,
- Tập hợp được nhiều người dùng liên quan,...

Nhược điểm của phương pháp họp nhóm:

- Nếu số đại biểu nhiều sẽ tốn thời gian cho việc ra quyết định,
- Tốn thời gian,
- Các ngắt quãng làm phân tán sự chú ý,
- Mời không đúng thành viên dẫn tới chậm có kết quả,
- Dễ chuyển sang các chủ đề không liên quan như là chính trị, thể thao,...

## 3. *Quan sát*

Quan sát là hoạt động có thể tiến hành thủ công hoặc tự động như sau:

+ Theo cách thủ công, người quan sát ngồi tại một chỗ và ghi chép các hoạt động, các bước xử lý công việc. Ghi chép hoặc băng ghi hình được dùng để phân tích cho các sự kiện, các mô tả động từ chính, hoặc các hoạt động chỉ rõ lý do, công việc hoặc các thông tin về công việc.

+ Theo cách tự động, máy tính sẽ lưu giữ các chương trình thường trú, lưu lại vết của các chương trình được sử dụng, email và các hoạt động khác được xử lý bởi máy. Các file nhật ký của máy sẽ được phân tích để mô tả công việc.

Quan sát có các nhược điểm:

- Thời gian của quan sát có thể không biểu diễn cho các công việc diễn ra thông thường,
- Ý nghĩ là đang bị quan sát có thể làm thay đổi thói quen thường ngày của người bị quan sát,
- Tốn thời gian.

Tuy nhiên, quan sát có các ưu điểm:

- Kỹ sư phần mềm có thể nhận được sự hiểu biết tốt về môi trường công tác hiện tại và quá trình xử lý thông qua quan sát.
- Kỹ sư phần mềm có thể tập trung vào vấn đề, mà không bị ảnh hưởng bởi người khác.
- Các ngăn cách giữa kỹ sư phần mềm và các người được phỏng vấn sẽ được vượt qua bởi quan sát.

Để quan sát có hiệu quả, nên xác định cái gì sẽ được quan sát và xác định thời gian cần thiết cho việc quan sát. Đồng thời, hãy xin sự chấp thuận của cả người quản lý và cá nhân trước khi tiến hành quan sát.

#### *4. Ấn định công việc tạm thời*

Với một công việc tạm thời, ta có được nhận thức đầy đủ hơn về các nhiệm vụ, đồng thời nó cung cấp cho ta kinh nghiệm thực tế. Đầu tiên, ta học các thuật ngữ và ngữ cảnh sử dụng nó. Thời gian tiến hành thường từ 2 tuần đến 1 tháng đủ dài để bạn có thể quen thuộc với phần lớn các công việc thông thường và các tình huống ngoại lệ nhưng không được quá dài để trở thành chuyên gia thực sự đối với công việc.

Công việc tạm thời cho ta cơ sở hình thức hoá các câu hỏi về các chức năng nào của phương pháp hiện thời của công việc sẽ được lưu giữ lại và cái nào sẽ bị loại trừ hoặc thay đổi. Nó cũng là cách thức để trả lời các câu hỏi không thể thực hiện được

Bất lợi của công việc tạm thời là tốn thời gian và sự lựa chọn về thời gian có thể làm tối thiểu hoá vấn đề. Thêm vào đó, người kỹ sư phần mềm có thể có thiên kiến về quá trình xử lý công việc, nội dung làm ảnh hưởng tới công việc thiết kế sau này.

#### *5. Điều tra qua câu hỏi*

Điều tra qua câu hỏi là xây dựng các câu hỏi để phỏng vấn trên giấy hoặc máy tính. Các câu hỏi được dùng để nhận các thông tin từ số lượng lớn người sử dụng và thường ở dạng khả năng lựa chọn, người trả lời chỉ việc đánh dấu. Các mục câu hỏi, như là phỏng vấn, có thể là câu hỏi mở hoặc câu hỏi đóng nhưng không chỉ rõ tên, dẫn đến các câu trả lời trung thực hơn nhiều phỏng vấn.

Ưu điểm của phương pháp điều tra:

- Các trả lời không cần biết tên nên quan điểm và cảm nhận thu được là trung thực,
- Có thể tiến hành với nhiều người,
- Thích hợp với các câu hỏi đóng và hữu hạn,
- Phù hợp với công ty đa văn hoá và có thể tùy biến với quy ước địa phương,...

Hạn chế của phương pháp điều tra:

- Khó thực hiện lại được,
- Các câu hỏi không có trả lời có nghĩa là không thu được thông tin,
- Các câu hỏi có thể khó hiểu,
- Thực hiện và đánh giá có thể chậm,
- Không thể thêm các thông tin khi đã tiến hành công việc,

- Thông tin thu được hạn chế trong một phạm vi hẹp,
- Chỉ dùng nó như một phương pháp bổ sung,...

#### 6. Xem xét tài liệu

Khái niệm tài liệu ám chỉ tới các cảm nang, quy định, các thao tác chuẩn mà tổ chức cung cấp như là hướng dẫn cho các nhà quản lý và nhân viên.

Các tài liệu thực sự hữu ích cho các kỹ sư phần mềm để học về các lĩnh vực mà họ chưa từng có kinh nghiệm. Nó có thể hữu ích cho việc xác định các câu hỏi về quá trình thao tác và sản xuất. Tài liệu đưa ra các thông tin khách quan.

Tuy nhiên, các tài liệu không phải luôn nằm trong công ty, nó có thể là các ấn phẩm kỹ thuật, các báo cáo nghiên cứu,... nên khó cho việc tìm kiếm. Nó còn gây thành kiến và không cung cấp để có thể nhận biết được quan điểm, động cơ,...

#### 7. Xem xét phần mềm

Khi các ứng dụng cũ phải được thay thế các phần mềm mới, việc nghiên cứu các phần mềm đã tồn tại cung cấp cho chúng ta các thông tin về quá trình xử lý công việc hiện thời và các mở rộng có ràng buộc bởi thiết kế phần mềm.

Khuyết điểm chính của việc nhận thông tin từ quan sát phần mềm là tài liệu có thể không chính xác hoặc kịp thời. Thêm vào đó, thời gian có thể bị lãng phí nếu ứng dụng đang bị xóa bỏ.

#### **3.5.2.2. Đánh giá tính phù hợp của các kỹ thuật thu thập dữ liệu đối với đặc tính của dữ liệu**

Phỏng vấn và họp nhóm là phù hợp với mọi loại kiểu dữ liệu do đó chúng thường xuyên được sử dụng.

Quan sát chỉ cung cấp các định hướng thô về mặt độ lớn, nhưng bị hạn chế bởi thời gian. Bởi vì thông tin từ quan sát không có cấu trúc, một vài kỹ năng được dùng cho kỹ sư phần mềm để tạo nên cấu trúc phù hợp với các tình huống. Thông tin có thể không được đầy đủ.

Kỹ thuật đặt câu hỏi có thể hỏi các câu hỏi có cấu trúc. Nếu câu hỏi là mở thì độ đầy đủ có thể thấp. Mức độ nhập nhằng của kỹ thuật đặt câu hỏi nên thấp nhưng ngữ nghĩa câu hỏi có thể không được biên dịch bởi người trả lời. Câu hỏi về độ lớn tại phòng ban hoặc tổ chức thường không thích hợp. Các thông tin về độ lớn của giao dịch hoặc thời gian cho xử lý giao dịch cho các nhân viên nên nhận được các thông tin có nghĩa.

Làm việc tạm thời tương tự quan sát ở chỗ có mức độ không chắc chắn cao kết hợp với các thông tin nhận được. Các thông tin có khuynh hướng hiện tại, phi cấu trúc, và không đầy đủ phụ thuộc vào các thời kỳ công tác. Các nhập nhằng thay đổi từ mức thấp đến mức trung bình phụ thuộc mức độ công việc được tổ chức và định nghĩa tốt đến mức nào. Nội dung ngữ nghĩa có thể thay đổi phụ thuộc các định nghĩa dùng chung trong nhóm làm việc.

Tài liệu cung cấp các thông tin không đầy đủ, phi cấu trúc. Thời gian nghiên cứu thay đổi phụ thuộc vào tài liệu nằm bên trong hay bên ngoài công ty. Các tài liệu bên trong thường liên quan tới các thông tin cũ trong khi tài liệu bên ngoài thường hướng về thông tin hiện tại và tương lai. Các tài liệu bên ngoài mang tính phổ quát và thống nhất trong khi tài liệu bên trong thường thay đổi theo phòng ban.

Phần mềm cung cấp các thông tin cũ, đôi khi là hiện tại được cấu trúc bởi vì đã được tự động hoá. Mức độ nhập nhằng có thể từ thấp đến trung bình. Các thông tin về độ lớn có thể được biểu diễn nhưng nên có kiểm tra chéo khi sử dụng các phương pháp khác.

Tính phù hợp của các kỹ thuật thu thập dữ liệu đối với đặc tính của dữ liệu được tóm tắt ở bảng sau

<b>Đặc tính dữ liệu</b> <b>Kỹ thuật</b>	<b>Tính thời gian</b>	<b>Cấu trúc</b>	<b>Mức đầy đủ</b>	<b>Nhập nhằng</b>	<b>Ngữ nghĩa</b>	<b>Độ lớn</b>
Phòng vấn	Các loại	Các loại	Các loại	Các loại	Thay đổi	Các loại
Hợp nhóm	Các loại	Các loại	Các loại	Các loại	Thay đổi	Các loại
Quan sát	Hiện tại	Phi cấu trúc	Không đầy đủ	Thay đổi	Thay đổi	Thô
Câu hỏi	Các loại	Cấu trúc	Đầy đủ	Thấp	Cố định (phụ thuộc cách hiểu)	Cá nhân
Công việc tạm thời	Hiện tại	Phi cấu trúc	Không đầy đủ	Thấp, trung bình	Thay đổi	Phụ thuộc cách hiểu
Tài liệu	Quá khứ hiện tại	Phi cấu trúc	Không đầy đủ	Thấp, trung bình	Thay đổi, hoặc cố định	Có thể, hoặc không
Xem xét phần mềm	Quá khứ Hiện tại	Cấu trúc	Đầy đủ	Thấp, trung bình	Cố định	Có thể

### 3.5.2.3. Đánh giá tính phù hợp của các kỹ thuật thu thập dữ liệu đối với các kiểu ứng dụng

+ Hệ xử lý giao dịch và các ứng dụng hỏi đáp có thể dùng mọi kỹ thuật.

Hợp nhóm và phỏng vấn có ưu thế vượt trội bởi vì chúng gợi ra một phạm vi rộng nhất các đáp ứng trong thời gian ngắn nhất.

Quan sát và ấn định công việc tạm thời hữu dụng cho việc nhận các thông tin nền tảng về vấn đề hiện thời, nhưng cần phải được sử dụng với chỉ định để không gây định kiến với thiết kế ứng dụng.

Các câu hỏi thích hợp với số lượng nhân viên tương đối lớn - lớn hơn 50 - và việc nhận ra các tính chất của người dùng đã xác định, ví dụ, yêu cầu đào tạo của người dùng trong khi phân tích các đặc tính của tổ chức. Nó yêu cầu về màn hình, ví

dụ, các kiểu màu sắc khác nhau, các câu hỏi có thể thích hợp cho việc biểu diễn một tập nhỏ các lựa chọn cho người dùng.

+ Hệ hỗ trợ quyết định cũng được coi là có thể dùng cho mọi kỹ thuật, nhưng không phải mọi kỹ thuật đều thích hợp cho mọi tình huống. Hệ hỗ trợ quyết định nói chung được phát triển dành cho các công việc mang tính chất riêng rẽ. Do đó, quan sát hoặc làm việc cùng một hoặc hai người đại diện có thể dẫn tới cách nhìn thành kiến về yêu cầu ứng dụng. Đối với tài liệu là các báo cáo thống kê có thể phù hợp cung cấp cho các ví dụ của các kiểu phân tích cần thiết của hệ hỗ trợ quyết định. Các tài liệu khác, như là chính sách, thủ tục nói chung là không thích hợp. Đối với các hệ hỗ trợ quyết định mục đích chung cùng với số lượng lớn người dùng, các câu hỏi là cách thích hợp để xác định phạm vi của vấn đề và các kỹ thuật phân tích cần thiết cho hệ hỗ trợ quyết định. Các thông tin này có thể được bổ sung bởi phỏng vấn và họp nhóm để xác định thêm các chi tiết.

+ Hệ hỗ trợ quyết định theo nhóm thường xuyên là một hệ cho khách hàng xây dựng các gói phần mềm cung cấp các kiểu hỗ trợ khác nhau cho các họp nhóm tự động. Các kỹ sư phần mềm làm việc trên môi trường hỗ trợ quyết định theo nhóm cần biết các kiểu dữ liệu, số lượng người tham gia, cũng như các kiểu suy luận và kỹ thuật thống nhất ý kiến của nhóm.

Các khối hỗ trợ quyết định theo nhóm không phải là các kiến thức chung mà cũng không được sử dụng thường xuyên. Bạn có thể xây dựng một hệ hỗ trợ quyết định theo nhóm cho một người. Do đó, một thời gian đáng kể phải dùng để tìm ra thị trường, người bán, và các khối hệ hỗ trợ quyết định theo nhóm. Các tài liệu bên ngoài về sản phẩm thích hợp cho các câu hỏi phát triển dẫn dắt tới các thông tin cần thiết. Sau khi kiến thức về thị trường đã được thu thập thì phỏng vấn và họp nhóm thích hợp cho xác định các yêu cầu đặc trưng và để xem xét lại với người dùng hệ hỗ trợ quyết định theo nhóm có thể làm và không làm cái gì. Các phương pháp khác có thể có giá trị hữu hạn. Ví dụ, quan sát có thể giúp kỹ sư phần mềm biết các công cụ sẽ làm việc thế nào. Các tài liệu bên trong cung cấp các thông tin về họp nhóm mà hệ hỗ trợ quyết định theo nhóm được chờ đợi sẽ cung cấp cũng có thể được dùng.

+ Hệ thông tin điều hành tương tự hệ hỗ trợ quyết định theo nhóm về sự khan hiếm và thiếu tổng quát về các thông tin - tri thức - liên quan. Hệ thông tin điều hành thường được xây dựng cho một nhóm người dùng tương đối ít nên không phù hợp với việc đặt câu hỏi. Hệ thông tin điều hành là môi trường khá chuyên biệt nên các tài liệu và phần mềm cũ không có giá trị nhiều. Quan sát không có tác dụng nhiều bởi vì việc thực hiện không thích hợp cho quan sát. Công việc tạm thời không dùng được vì bạn không thể làm việc được chỉ trong một hoặc hai tuần. Do vậy, các phương pháp phỏng vấn, họp nhóm, xem tài liệu là các kỹ thuật phù hợp nhất.

Tính phù hợp của các kỹ thuật thu thập dữ liệu đối với các kiểu ứng dụng được chỉ ra ở bảng sau:

<b>Loại ứng dụng</b> <b>Kỹ thuật</b>	<b>TPS</b>	<b>CSDL</b>	<b>DSS</b>	<b>GDSS</b>	<b>EIS</b>	<b>ES</b>
Phỏng vấn	Tốt	Tốt	Tốt	Tốt	Tốt	Tốt
Họp nhóm	Tốt	Tốt	Tốt	Tốt	Tốt	Tốt
Quan sát	Tốt	Tốt	Tốt	Hạn chế	Hạn chế	
Câu hỏi	Tốt	Tốt	Tốt			
Công việc tạm thời	Tốt	Tốt	Tốt			
Xem tài liệu	Tốt	Tốt	Tốt			
Xem xét phần mềm	Tốt	Tốt	Tốt	Hạn chế	Hạn chế	Hạn chế

### Câu hỏi

1. Hãy chỉ rõ vai trò quan trọng của việc xác định đúng đắn yêu cầu của bài toán.
2. Hãy phân tích để chỉ rõ các sai lầm thường gặp trong việc xác định yêu cầu.
3. Các công việc phải làm để có được một tài liệu yêu cầu chính xác.
4. Hãy nêu và phân tích các đặc tính dữ liệu của yêu cầu.
5. Các kỹ thuật để thu thập dữ liệu cho bài toán. Khi xây dựng một ứng dụng, bạn đã sử dụng những kỹ thuật nào, vì sao?
6. Hãy xây dựng tài liệu yêu cầu cho các hệ thống phần mềm mà các anh chị đã tìm hiểu.