

.....o0o.....

## **Bài giảng: Công nghệ phần mềm**

Tai lieu.vn

## Lời nói đầu

Trong hơn ba chục năm qua người ta đã chứng kiến sự lớn mạnh về số lượng cũng như mức độ quan trọng trong việc ứng dụng cơ sở dữ liệu. Các cơ sở dữ liệu là thành phần cơ bản trong hệ thống thông tin, dùng trong cả máy lớn lẫn máy nhỏ. Việc thiết kế cơ sở dữ liệu được coi là hoạt động thông dụng, có hiệu quả đối với cán bộ chuyên môn lẫn người dùng không chuyên.

Từ cuối những năm 60, khi các cơ sở dữ liệu xuất hiện lần đầu trên thị trường phần mềm, người thiết kế xoay sở như thợ thủ công, họ dùng sơ đồ khối, các cấu trúc bản ghi và thiết kế cơ sở dữ liệu thường bị lẫn với việc cài đặt cơ sở dữ liệu, tình huống này đã thay đổi, các phương pháp và các mô hình thiết kế cơ sở dữ liệu đã tiến hoá song song với quá trình công nghệ hệ thống cơ sở dữ liệu. Khi làm việc với cơ sở dữ liệu quan hệ người ta sử dụng các ngôn ngữ hỏi mạnh, công cụ phát triển ứng dụng và giao diện người dùng thân thiện. Công nghệ cơ sở dữ liệu đã có nền lý thuyết, gồm lý thuyết quan hệ về dữ liệu, xử lý câu hỏi và tối ưu, điều khiển tương tranh, quản lý giao tác và khôi phục sai sót...

Khi công nghệ cơ sở dữ liệu đã tiến lên, công nghệ thiết kế và các kỹ thuật cũng phát triển. Người ta đã chia quá trình thành các pha, đặt mục đích chính cho mỗi pha, và các kỹ thuật đạt được các pha đó.

Tuy nhiên các phương pháp luận thiết kế cơ sở dữ liệu không thông dụng; hầu hết các tổ chức và các nhà thiết kế cá nhân ít tuân theo các phương pháp luận thiết kế, và điều đó cũng dễ dẫn đến sai lầm trong việc phát triển các hệ thống thông tin. Bên cạnh việc thiếu tiếp cận cấu trúc về thiết kế cơ sở dữ liệu, thời gian và tài nguyên cần cho đề án cơ sở dữ liệu thường bị đánh giá thấp. Do vậy các cơ sở dữ liệu phát triển là không hoàn thiện và không hiệu quả so với nhu cầu của các ứng dụng; các tư liệu không đầy đủ và bảo trì còn nhiều vấn đề vướng mắc.

Nhiều bài toán đã không rõ ràng và không trong sáng về bản chất chính xác của dữ liệu tại mức khái niệm và mức triu tượng. Trong nhiều trường hợp, dữ liệu được mô tả từ khi bắt đầu đề án trong cấu trúc dữ liệu lưu trữ; chứ không tập trung vào việc hiểu thuộc tính có cấu trúc của dữ liệu. Các dữ liệu cần đọc lập với việc cài đặt.

Vì vậy để xây dựng một hệ thống thông tin cần có các mô hình thiết kế cụ thể để tránh những thiếu sót đã nêu trên. Một trong những mô hình được áp dụng rộng rãi và có hiệu quả là mô hình thác nước. Ngày nay, nhiều nghiên cứu đã cho ra đời nhiều mô hình tiến bộ hơn, có thể xây dựng được các hệ thống thông tin lớn như: mô hình phát triển tiến hoá, mô hình xoắn ốc của Bochor... Tuy nhiên mô hình thác nước là một mô hình đơn giản và thích hợp với những bài toán không quá lớn.

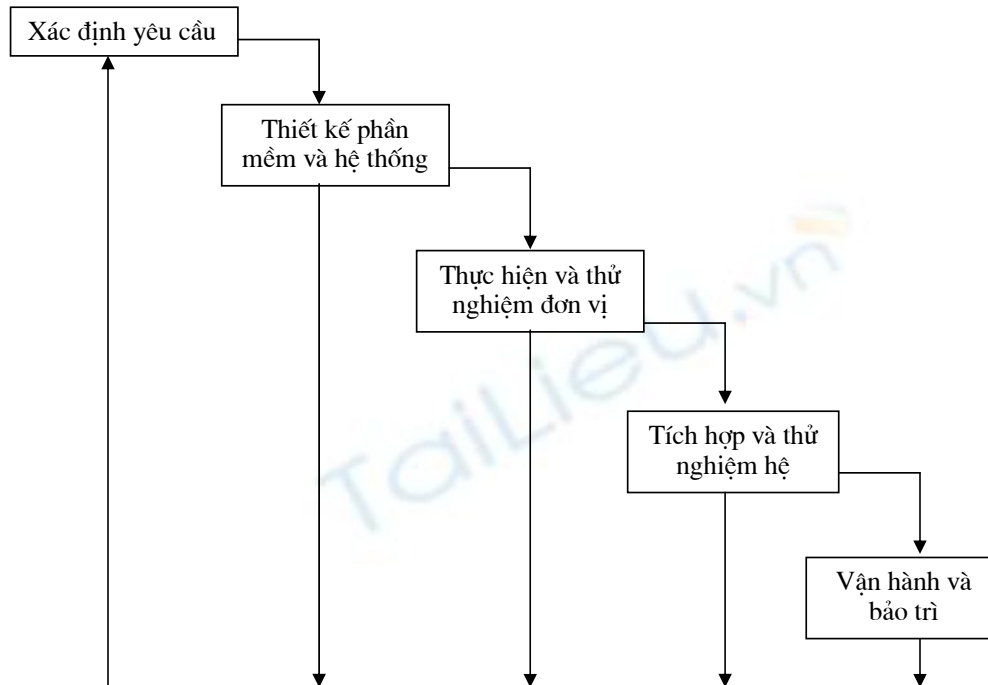
Trong khuôn khổ của một buổi **cenema**, tôi muốn giới thiệu qua với các bạn các khái niệm, các bước cơ bản để xây dựng một hệ thống thông tin bằng mô hình thác nước, những thuận lợi và khó khăn trong từng bước xây dựng, cũng như những tiến bộ và hạn chế khi sử dụng mô hình này để xây dựng hệ thống thông tin.

Vì thời gian và khả năng có hạn nên tôi không thể mô tả một cách chi tiết từng bước trong quá trình thiết kế. Rất mong nhận được những ý kiến đóng góp ở các bạn để chương trình ngày càng hoàn thiện hơn. Xin cảm ơn tất cả mọi người.

## Mô hình thác nước

Mô hình đầu tiên trong việc xử lý và phát triển phần mềm bắt nguồn từ những công nghệ xử lý khác nhau. Mô hình này đã được các nhà hoạch định dự án chấp thuận. Nó bao gồm các tiến trình phát triển rõ ràng và cụ thể, kế cận nhau giống như thác nước. Vì vậy nó có tên gọi là mô hình thác nước

Có 5 tiến trình trong một chu trình của mô hình thác nước:



### 1. Xác định yêu cầu:

Để xây dựng được một phần mềm mang tính thực tiễn cao, trước hết cần phải trả lời được câu hỏi phần mềm được xây dựng làm gì, ứng dụng vào lĩnh vực nào? Hay ngược lại, các nhà sản xuất phần mềm cần phải biết được các nhà phát triển và người sử dụng muốn gì trong sản phẩm của họ. Nói cách khác cần phải có sự trao đổi giữa các nhà sản xuất với các nhà phát triển và người sử dụng, để từ đó rút ra được một bản danh sách các yêu cầu một cách đầy đủ và chi tiết. Đây là một bước công phu và nhiều khó khăn.

### 2. Thiết kế phần mềm và hệ thống:

Quy trình thiết kế hệ thống chia các yêu cầu thành hệ thống phần mềm và phần cứng. Nó được bao trùm bởi một cấu trúc hệ thống. Thiết kế phần mềm liên quan đến việc sử dụng các ngôn ngữ lập trình để viết các hàm phần mềm theo hệ thống các yêu cầu từ đó có thể dịch sang một hay nhiều chương trình có tính thực thi.

### 3. Thực hiện và thử nghiệm đơn vị:

Trong bước này, việc thiết kế phần mềm thực chất là thiết lập một chương trình hay các Modul chương trình riêng lẻ. Việc thử nghiệm từng modul chương trình liên quan đến việc thẩm tra mỗi modul bằng cách đưa vào các chi tiết trong yêu cầu kỹ thuật của nó.

#### 4. Tích hợp và chạy thử cả hệ thống:

Dùng modul chương trình hay các chương trình đơn lẻ được tích hợp lại với nhau và chạy thử như một hệ thống hoàn chỉnh để đảm bảo rằng các yêu cầu phần mềm đã được đáp ứng. Sau khi kiểm thử, hệ thống phần mềm được giao lại cho khách hàng.

#### 5. Vận hành và bảo trì:

Thông thường, đây là bước có chu kỳ dài nhất, hệ thống đã hoàn tất và được đưa vào sử dụng. Bảo trì là việc sửa chữa lại các sai lầm không được phát hiện sớm ở các bước trước đó, phát hiện các tính năng của từng modul trong hệ thống và nâng cao khả năng phục vụ hệ thống khi các yêu cầu mới được phát triển thêm.

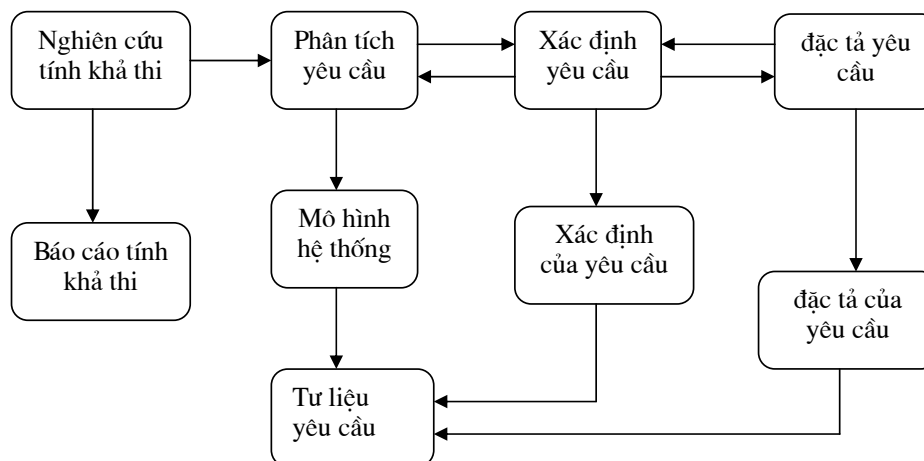
Trong thực tế, các bước này không độc lập với nhau chúng đan xen lẫn nhau và bổ xung thông tin cho nhau trong quá trình thiết kế hệ thống. Xử lý phần mềm không phải là một dạng đường đơn giản mà nó liên quan đến một hệ thống lặp lại tuần tự của các hoạt động phát triển. Trong giai đoạn cuối cùng, phần mềm được đưa vào sử dụng, các lỗi và những thiếu sót trong yêu cầu phần mềm được phát hiện, cần phải định nghĩa thêm một số chức năng mới. Việc sửa đổi trở nên cần thiết để những phần mềm còn sai sót trở nên hữu ích hơn. Thực hiện các cải tiến có thể liên quan đến việc lặp lại các bước trước đó.

Đáng tiếc rằng, một mô hình bao gồm nhiều bước lặp lại tuần tự khó có thể nhận ra một cách rõ ràng và nhanh chóng các sai lầm và thiếu sót cho việc lập kế hoạch và báo cáo. Tuy nhiên sau một số ít bước lặp lại, công việc này có thể được tiến hành một cách dễ dàng hơn và có thể bổ xung thông tin vào bước đặc tả yêu cầu. Sự cố định sớm các yêu cầu có thể làm cho hệ thống không thực hiện được các công việc mà người dùng cần đến. Nó cũng có thể dẫn đến việc xây dựng các hệ thống không tốt, có thể dẫn đến tình trạng hệ thống bị phá vỡ khi thực hiện.

Hạn chế của mô hình thác nước là không linh hoạt trong việc phân chia các đối tượng thành các bước riêng biệt. Đôi khi việc cứu vãn hệ thống là không thể khi nó không đáp ứng được các yêu cầu mới của khách hàng. Tuy nhiên, mô hình thác nước phản ánh công nghệ theo lối tư duy tự nhiên quen thuộc và nó thích hợp với các hệ thống không quá lớn. Do vậy nó vẫn còn được sử dụng rộng rãi.

### Quá trình xây dựng yêu cầu

Muốn xây dựng được hệ thống trước tiên ta phải khảo sát được các yêu cầu của hệ thống. Thông thường để xây dựng một yêu cầu ta cần phải thực hiện các bước như sơ đồ sau:



Yêu cầu là những gì được phát biểu ra, thường là văn bản những gì mà khách hàng muốn có. Trên thực tế giữa yêu cầu thật sự với những gì được phát biểu có sự khác nhau. Nhu cầu là những gì mà người sử dụng muốn, nhưng yêu cầu đôi khi không thoả mãn hết được các nhu cầu.

Yêu cầu hệ thống và mục tiêu của hệ thống thường được đưa ra bằng những khái niệm mang tính định lượng có thể được kiểm chứng.

Khi có một kí kết hợp đồng người ta thường đưa ra yêu cầu hệ thống chứ không đưa ra nhu cầu và mục tiêu của hệ thống. Bởi hai yếu tố này có thể được kiểm chứng đánh giá. Công việc phân tích và nắm bắt nhu cầu luôn luôn gặp khó khăn. Bởi những khách hàng nắm bắt được các kiến thức chuyên ngành nhưng không biết chuyên môn tin học nên không thể đưa ra được yêu cầu thích hợp cho hệ thống và ngược lại người làm hệ thống lại không nắm được các kiến thức chuyên ngành. Do vậy, các đặc tả ban đầu thường không tốt.

Có bốn bước quan trọng trong quá trình xây dựng yêu cầu:

- Nghiên cứu tính khả thi :

Công việc ước lượng đánh giá được bắt đầu từ việc nhận biết nhu cầu của người sử dụng. Có thể các phần mềm hiện tại đã làm hài lòng người sử dụng, việc nghiên cứu tính khả thi sẽ quyết định hệ thống được đề xuất có hiệu quả cao hơn theo quan điểm của người kinh doanh hay không và nó có thể được cung cấp ngân sách để tiến hành hay không. Nghiên cứu tính khả thi sẽ làm giảm thời gian và chi phí cho việc sản xuất phần mềm. Kết quả sẽ thông tin đến người ra quyết định với bản phân tích chi tiết hơn.

- Phân tích yêu cầu: Đây là một quá trình xác định các yêu cầu hệ thống thông qua việc theo dõi sự tồn tại của hệ thống, thảo luận với người sử dụng và người ra quyết định. Qui trình này có thể liên quan đến sự phát triển của một hay nhiều mô hình hệ thống khác nhau. Quá trình phân tích giúp ta hiểu được những yêu cầu mà hệ thống đã nêu ra. Những nguyên mẫu hệ thống cũng có thể được phát triển để giúp ta hiểu rõ hơn các yêu cầu.
- Xác định yêu cầu: Qui trình này tập hợp các yêu cầu đã thu thập được thành một tài liệu. Nó phản ánh chính xác điều mà khách hàng muốn, được thể hiện bằng ngôn ngữ tự nhiên cùng với các bảng biểu thể hiện được thông tin chung nhất với người sử dụng. Tài liệu là thông tin do phía khách hàng cung cấp, tài liệu này không dùng để kí kết hợp đồng.
- Đặc tả yêu cầu: Các yêu cầu thường có cấu trúc rõ ràng, tỉ mỉ. Đây là cơ sở cho phía khách hàng và nhà cung cấp kí kết hợp đồng. Việc tạo ra tài liệu này thường được thực hiện song song với việc thiết kế mức cao. Việc thiết kế và các hoạt động yêu cầu ảnh hưởng lẫn nhau trong quá trình thực hiện. Trong quá trình tạo ra tài liệu này các lỗi của bước xác định yêu cầu sẽ được khám phá. Đây là mức mô tả trừu tượng các dịch vụ của hệ thống được sử dụng làm cơ sở cho việc thiết kế và thực hiện phần mềm.

Trên đây là 4 bước quan trọng trong quá trình xây dựng yêu cầu. Bây giờ ta sẽ đi sâu phân tích một số bước cụ thể:

## **I/ Phân tích yêu cầu:**

Sau khi nghiên cứu tính khả thi, bước quan trọng đầu tiên là phân tích hoặc suy luận các yêu cầu. Các chuyên gia phát triển phần mềm làm việc với các khách hàng và người sử dụng cuối

cùng để tìm ra các lĩnh vực ứng dụng mà cấp hệ thống phần mềm phải đáp ứng, việc thực hiện các yêu cầu hệ thống do phần mềm và các yếu tố khác quy định, chẳng hạn như phần cứng, các thiết bị ngoại vi ...

Phân tích yêu cầu là một bước quan trọng, tính khả thi của hệ thống sau này phụ thuộc nhiều vào quá trình trao đổi giữa nhu cầu của khách hàng và nhà cung cấp với các công việc được tự động hoá. Nếu việc phân tích không đáp ứng nhu cầu thực của khách hàng, hệ thống sau khi hình thành sẽ không đáp ứng được các yêu cầu.

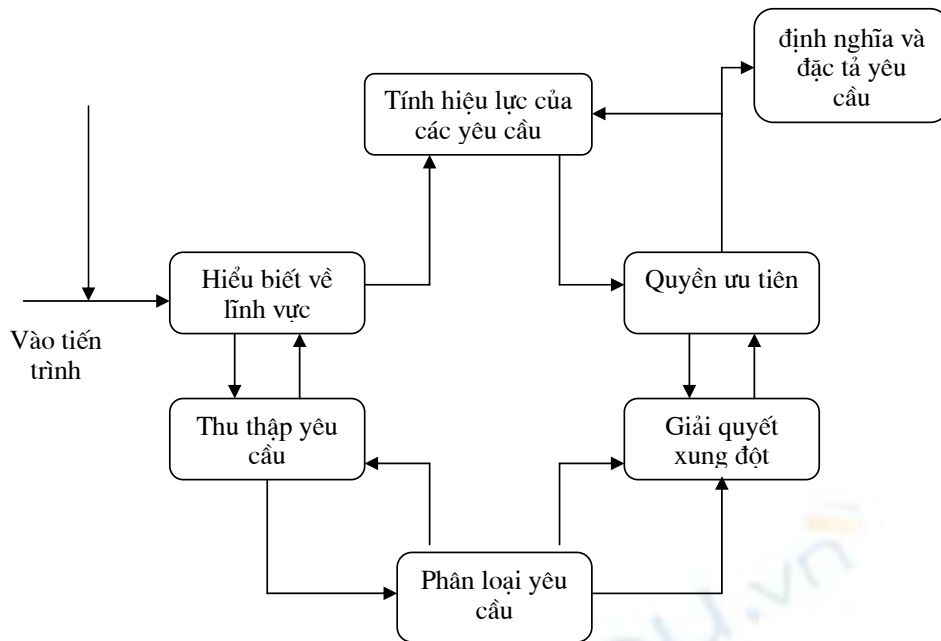
Phân tích yêu cầu có thể còn liên quan đến sự đa dạng của các cấp bậc chức vụ khác nhau của các nhân viên trong cùng một tổ chức. Hay nói cách khác, vấn đề bảo mật gây ra rất nhiều khó khăn trong phân tích yêu cầu. Điều này ảnh hưởng tới người tác động cuối cùng vào hệ thống, người tổ chức và thiết đặt hệ thống. Các nhà đầu tư có ảnh hưởng trực tiếp hoặc gián tiếp tới những yêu cầu của hệ thống.

Bước phân tích yêu cầu là khó bởi một số lý do sau:

- Trong hầu hết các trường hợp, các nhà đầu tư không biết thực sự họ muốn gì ở hệ thống máy tính. Thậm chí khi họ có một định hướng rõ ràng họ mới có thể biết hệ thống cần phải làm gì, và rất khó khăn để kết hợp các yêu cầu lại với nhau. Họ có thể làm sai lệch nhu cầu bởi họ không biết giá trị của câu hỏi.
- Các nhà đầu tư trong một hệ thống thường bộc lộ các yêu cầu với kiến thức ngầm định trong công việc của họ. Còn người kỹ sư không có nhiều kinh nghiệm trong các lĩnh vực của khách hàng, do vậy cần phải hiểu được những yêu cầu và dịch chúng sang một dạng tài liệu chấp nhận được.
- Các nhà đầu tư khác nhau có những yêu cầu khác nhau và họ có thể tiến hành theo những cách rất khác nhau. Người kỹ sư phải nhận biết được những điểm chung và những điểm khác biệt đó.
- Việc phân tích lấy một không gian trong khung cảnh của tổ chức, những nhân tố chính trị (chẳng hạn như việc phân quyền) có thể ảnh hưởng đến yêu cầu của hệ thống, những nhân tố này có thể không rõ ràng mạch lạc tới người sử dụng cuối cùng. Ví dụ như người lãnh đạo thường có quyền cao hơn đối với hệ thống.
- Việc phân tích lấy khía cạnh thương mại và kinh tế làm động lực. Nó chắc chắn sẽ thay đổi trong quá trình phân tích. Từ đây, những yêu cầu riêng lẻ có thể thay đổi, những yêu cầu mới có thể xuất hiện từ phía các nhà đầu tư mới. Các nhà đầu tư mới sẽ không phải thăm dò lại từ đầu mà chỉ phải bổ xung thêm thông tin hay yêu cầu.

Để tiến hành phân tích yêu cầu, ta phải dựa vào sự hiểu biết nhất định trong các lĩnh vực cụ thể, khi đó mô hình của tiến trình phân tích chắc chắn sẽ đơn giản hơn.

Mô hình sau đây nêu lên một số bước quan trọng trong tiến trình phân tích.



Trong sơ đồ trên các bước bổ xung cho nhau, chu kỳ bắt đầu từ hiểu biết về các lĩnh vực và cuối cùng là bản phê chuẩn các yêu cầu. Những hiểu biết của quá trình phân tích sẽ dần được cải thiện sau mỗi chu trình.

- **Hiểu biết về lĩnh vực ứng dụng:** Phân tích phải được phát triển qua sự hiểu biết về các lĩnh vực ứng dụng. Giả sử có một hệ thống siêu thị, quy trình phân tích phải tìm ra được những điểm chung nhất, khái quát nhất giữa các siêu thị.
- **Thu thập yêu cầu:** Quy trình này liên quan đến các nhà đầu tư, người xây dựng hệ thống phải thông qua họ để khám phá ra các yêu cầu. Từ đó sẽ nâng cao hiểu biết để phát triển và xây dựng hệ thống.
- **Phân loại yêu cầu:** Quá trình này lấy các yêu cầu một cách ngẫu nhiên, không có cấu trúc và sắp xếp chúng một cách có hệ thống.
- **Giải quyết xung đột:** Chắc chắn trong quá trình đưa ra yêu cầu, các nhà đầu tư do không có chuyên môn nên xung đột có thể xảy ra. Công việc của người xây dựng hệ thống là cần phải tìm ra và giải quyết được các xung đột đó.
- **Quyền ưu tiên:** Trong một tập hợp các yêu cầu, bao giờ cũng phải có những yêu cầu quan trọng, cấp bách hơn yêu cầu khác, vì vậy ta phải tác động đến các nhà đầu tư để khám phá ra các yêu cầu cần thiết nhất, từ đó có kế hoạch xây dựng hệ thống.
- **Làm cho các yêu cầu trở nên có hiệu lực:** Các yêu cầu phải đảm bảo tính kiên định và phù hợp với nhu cầu thực của các nhà đầu tư. Từ đó mới có thể xây dựng được một hệ thống hữu ích.

Trong quá trình phân tích yêu cầu, một vài mô hình hệ thống khác nhau có thể được sinh ra, mô hình là hình ảnh trù tượng của hệ thống, các mô hình khác nhau sẽ có các thông tin khác nhau. Sự khác nhau này xuất phát từ nguồn gốc các yêu cầu phục vụ khác nhau.

## II/ Xác định yêu cầu:

Xác định yêu cầu là mô tả trù tượng các dịch vụ mà hệ thống phải cung cấp cũng như các ràng buộc mà hệ thống phải tuân theo khi thực hiện. Đặc điểm của nó là tư liệu được viết theo kiểu hướng khách hàng, do vậy nó phải được viết bằng ngôn ngữ để khách hàng có thể hiểu được. Nó chỉ đặc tả hành vi bên ngoài của hệ thống mà không mô tả chi tiết thiết kế. Hệ thống các yêu cầu được chia làm hai loại:

- Các yêu cầu chức năng: Là các dịch vụ mà hệ thống phải cung cấp. Do vậy các yêu cầu chức năng sẽ tác động trở lại các dữ liệu vào và có ảnh hưởng đến một số tính hướng đặc biệt. Trong một số trường hợp, yêu cầu chức năng cũng có thể có các trạng thái không phải làm việc.
- Yêu cầu phi chức năng: Đó là các ràng buộc mà hệ thống phải tuân theo. Nó bao gồm sự ràng buộc về thời gian, các chuẩn công nghệ trong quá trình xử lý...

Trong thực tế, xác định yêu cầu của hệ thống vừa phải hoàn thiện, vừa phải tránh kiện, hoàn thiện có nghĩa là mọi yêu cầu của hệ thống được nêu lên đều phải có trong xác định yêu cầu, tính tránh kiện nghĩa là trong một xác định yêu cầu không thể có các yêu cầu phủ định nhau, mâu thuẫn nhau. Nói cách khác, nó phải đảm bảo được tính thống nhất.

Đối với các hệ thống phức tạp, ta khó có thể tìm được một xác định yêu cầu vừa đầy đủ, vừa tránh kiện. Một phần là do tính phức tạp cố hữu của hệ thống, một phần là do quan điểm khác nhau của nhu cầu người sử dụng. Tính không kiên định sẽ không rõ ràng khi các yêu cầu được đưa ra lần đầu, vấn đề là ta phải phát hiện ra nó ở các bước phân tích sâu hơn.

Xác định yêu cầu được viết bằng ngôn ngữ tự nhiên, từ đó nó sẽ nảy sinh ra các vấn đề sau:

- Tính thiếu rõ ràng: Do viết bằng ngôn ngữ tự nhiên, nên đôi khi mơ hồ dài dòng, khó hiểu.
- Sự lẫn lộn yêu cầu: Không phân biệt được đâu là yêu cầu chức năng và đâu là yêu cầu phi chức năng.
- Sự trộn lẫn trong các yêu cầu: Các yêu khác nhau có thể biểu diễn thành một yêu cầu chung.

Yêu cầu bao gồm cả khái niệm và thông tin chi tiết, nó đưa ra các khái niệm có cấu hình điều khiển thuận lợi được cung cấp như một phần cố định của APSE. Tuy nhiên nó cũng có những thuận lợi trong việc truy nhập tới đối tượng trong một nhóm người sử dụng một tên chưa hoàn chỉnh. Một vài tổ chức cố gắng sản xuất ra một đặc tả đơn để vừa hoạt động như một bản xác định yêu cầu vừa như một bản đặc tả yêu cầu. Khi một bản xác định yêu cầu được kết hợp với một bản đặc tả yêu cầu, ta thường nhầm lẫn giữa khái niệm và chi tiết.

Mô hình đầu tiên trong việc xác định yêu cầu là việc kết hợp ba loại yêu cầu khác nhau:

- Một nhận thức các trạng thái yêu cầu chức năng mà hệ thống soạn thảo sẽ cung cấp một mạng lưới. Nó đưa ra tính hợp lý cho vấn đề này.
- Một yêu cầu phi chức năng đưa ra những thông tin chi tiết về các đơn vị lưới(cm hay inc)
- Một yêu cầu phi chức năng của giao diện người dùng xác định người dùng bật hay tắt lưới.(chiều)

### III/ Đặc tả yêu cầu:

Đặc tả yêu cầu là đưa thêm các thông tin vào bản xác định yêu cầu. Đặc tả yêu cầu thường được dùng với các mô hình hệ thống, phát triển trong suốt quá trình phân tích yêu cầu. Đặc tả cùng và mô hình sẽ được mô tả trong hệ thống để thiết kế và thực hiện. Nó cũng bao gồm tất cả những thông tin cần thiết mà hệ thống phải thực hiện.



Ngôn ngữ tự nhiên thường được dùng để đặc tả yêu cầu. Tuy nhiên, đặc tả bằng ngôn ngữ tự nhiên không phải là cơ sở tốt cho một thiết kế hoặc cho một hợp đồng giữa khách hàng và người phát triển hệ thống. Sau đây là một vài lý do:

- Ngôn ngữ tự nhiên có thể hiểu được là dựa vào việc đọc bản đặc tả và khi viết sử dụng những từ giống nhau cho những khái niệm giống nhau. Điều này rất khó hiểu bởi các từ trong ngôn ngữ tự nhiên đôi khi rất mơ hồ.
- Một bản đặc tả yêu cầu bằng ngôn ngữ tự nhiên rất linh hoạt. bạn có thể nói về cùng một vấn đề bằng những cách khác nhau, nó làm cho người đọc khó tìm ra những yêu cầu giống nhau. đó là lỗi nghiêng về phía tiến trình.
- Các yêu cầu không được phân chia một cách có hiệu quả bằng ngôn ngữ đó, do đó khó tìm ra được các yêu cầu quan hệ. Để khám phá ra một sự thay đổi bạn phải xem xét tất cả các yêu cầu, chứ không chỉ trong nhóm các yêu cầu quan hệ.

Bởi tất cả các lý do trên, đặc tả yêu cầu được viết bằng ngôn ngữ tự nhiên có thể bị hiểu sai, điều này thường không được phát hiện cho đến giai đoạn thiết kế hoặc thực hiện các giai đoạn của tiến trình phần mềm. Vì vậy có một cách tốt hơn là dùng luận phiên các kí hiệu để tránh một vài vấn đề về không hạn chế được bằng ngôn ngữ tự nhiên, đó là đưa cấu trúc vào đặc tả. Sau đây là một số phương pháp:

- Ngôn ngữ tự nhiên có cấu trúc: Cách tiếp cận này dựa trên việc định nghĩa các chuẩn mẫu hoặc các chuẩn tạm thời để xây dựng đặc tả yêu cầu.
- Ngôn ngữ mô tả thiết kế: Cách tiếp cận này dựa vào việc sử dụng các ngôn ngữ giống như ngôn ngữ lập trình nhưng có nhiều điểm trừu tượng hơn để phân loại yêu cầu bằng việc đưa ra một mô hình có thể thực hiện được của hệ thống.
- Ngôn ngữ đặc tả yêu cầu: Một số ngôn ngữ được thiết kế cho những mục đích đặc biệt để xây dựng các yêu cầu phần mềm. Ví dụ như: PSL/PSA (Tiechrow và Hershey, 1977) và RSL (Alford, 1977; Bell et al.1977; Alford, 1985). Những tiến bộ của cách tiếp cận này là việc cung cấp các công cụ có mục đích đặc biệt được phát triển.
- Các kí hiệu đồ họa: Hệ thống kí hiệu đồ họa tốt nhất có lẽ là SADT (Ross, 1977; Schoman và Ross, 1977). SADT có một bộ kí hiệu đồ họa quen thuộc, vì vậy chủ yếu được các nhà đầu tư sử dụng. Nó trở nên thân thiện trong việc sử dụng để phân tích và đặc tả yêu cầu.
- Đặc tả toán học: Dùng các kỹ thuật dựa trên các khái niệm toán học có cơ sở như: tập hợp, máy hỗn hợp trạng thái hoặc lưới để đặc tả yêu cầu.  
Ưu điểm: loại bỏ hoàn toàn tính mơ hồ trong đặc tả.  
Nhược điểm: khó khăn cho bên khác hàng bởi họ không hiểu được các kí hiệu toán học.  
Khắc phục: thêm những chú giải bằng ngôn ngữ thử nghiệm ở những chỗ thích hợp.  
Tính dò theo được: khi viết ra các đặc tả yêu cầu, một điểm quan trọng là các yêu cầu có liên quan với nhau phải tham khảo chéo nhau được. Dò theo được là một thuộc tính của đặc tả yêu cầu phản ánh tính dễ tìm ra các yêu cầu có liên quan.

Davis (1990) đã tóm tắt và so sánh một vài cách tiếp cận khác nhau để đặc tả yêu cầu. Trong hai cách tiếp cận đầu, chủ yếu là ngôn ngữ thử nghiệm có cấu trúc và ngôn ngữ mô tả thiết kế. Việc làm thích ứng các ngôn ngữ yêu cầu không được biết hay sử dụng rộng rãi. Các kí hiệu đồ họa tương tự như các kí hiệu được sử dụng để xây dựng các mô hình hệ thống.

Khi viết đặc tả yêu cầu một điều rất quan trọng là các yêu cầu quan hệ phải phù hợp khi thay đổi các yêu cầu hoặc các yêu cầu khác được tìm thấy. Một vài mối quan hệ giữa các yêu cầu là rất tế nhị. Đó là lý do mà ta không thể vạch ra cụ thể các yêu cầu. Tuy nhiên, có một vài phương thức đơn giản có thể áp dụng cho việc xác định và đặc tả yêu cầu:

- Tất cả các yêu cầu nên được phân chia
- Các yêu cầu cần phải xác định rõ quan hệ.
- Mỗi tài liệu yêu cầu phải chứa một ma trận để chỉ ra các yêu cầu quan hệ.

Các ma trận khác nhau có thể được phát triển cho các loại quan hệ khác nhau.

Một kỹ thuật đơn giản là các công cụ CASE được sử dụng để cung cấp các phân tích và đặc tả yêu cầu. Một vài công cụ có các tiện ích đơn giản như tìm các yêu cầu sử dụng trong cùng thời kỳ, kết nối các tiện ích từ một yêu cầu đến một yêu cầu có quan hệ khác có thể được cung cấp.

## Thiết kế phần mềm

Một bản thiết kế tốt là chìa khoá dẫn đến thành công. Tuy nhiên, ta không thể chính thức hoá tiến trình xử lý trong bất kỳ quy tắc kỹ thuật nào. Thiết kế là sáng tạo ra một tiến trình nhìn thấu các yêu cầu và khả năng trong từng phần của thiết kế. Nó phải thực hành, học hỏi ở những người có kinh nghiệm và nghiên cứu các hệ thống đang tồn tại.

Các vấn đề thiết kế phải được khôi phục trong ba giai đoạn:

1. Nghiên cứu và hiểu được vấn đề: Nếu không hiểu được vấn đề, hiệu quả của công việc thiết kế phần mềm sẽ rất thấp. Chúng ta nên xem xét từ những khía cạnh, những quan điểm khác nhau trong các yêu cầu thiết kế.
2. Xác định toàn bộ các đặc trưng của giải pháp có thể: Điều này rất có lợi cho việc xác định các giải pháp và xem xét đánh giá chúng. Sự lựa chọn giải pháp phụ thuộc vào kinh nghiệm của người thiết kế, tính có giá trị của các thành phần có thể dùng lại được và sự đơn giản của giải pháp. Người thiết kế thường thích dùng các giải pháp quen thuộc mặc dù nó không phải là giải pháp tối ưu vì họ hiểu được những thuận lợi và khó khăn.
3. Các mô tả mang tính trừu tượng hoá được sử dụng trong giải pháp: Trước khi tạo ra một tài liệu chính thức, người thiết kế có thể viết một bản mô tả các thông tin thiết kế. Điều này có thể được phân tích trong quá trình thiết kế chi tiết. Các lỗi và các thiếu sót trong thiết kế mức cao sẽ được khám phá trong quá trình phân tích. Chúng sẽ được khắc phục khi làm thành tài liệu.

Tiến trình giải quyết vấn đề được lặp lại sau mỗi lần xác định các thực thể mang tính trừu tượng trong thiết kế. Tiến trình cải tiến còn tiếp tục cho tới khi một bản của mỗi thực thể mang tính trừu tượng được chuẩn bị.

### I/ Tiến trình thiết kế:

Một mô hình tổng thể của một bản thiết kế phần mềm là một sơ đồ có hướng. Mục tiêu của tiến trình thiết kế là tạo ra một sơ đồ không có mâu thuẫn. Cần chú ý rằng sơ đồ này được đưa ra những thực thể được thiết kế như các tiến trình, chức năng hoặc kiểu. Việc thiết lập các mối quan hệ giữa các thực thể thường xuyên được thực hiện.

Người thiết kế phần mềm không tới ngay điểm kết thúc của sơ đồ thiết kế ngay lập tức mà phát triển thêm bằng cách lặp lại các thiết kế qua một số phiên bản khác. Tiến trình thiết kế sẽ đưa thêm thông tin và chi tiết hoá khi bản thiết kế được phát triển với sự quay lui nhất định để sửa lỗi