

Giáo trình

PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG HƯỚNG ĐỐI TƯỢNG SỬ DỤNG UML

Biên soạn

ThS. Phạm Nguyễn Cương

TS. Hồ Tường Vinh

Giới thiệu

Hệ thống phần mềm càng ngày càng trở nên phức tạp. Các ứng dụng hôm nay có những yêu cầu và kiến trúc đòi hỏi phức tạp hơn rất nhiều so với quá khứ. Các kỹ thuật, công cụ, và phương pháp luận phát triển hệ thống phần mềm đang thay đổi một cách nhanh chóng. Các phương pháp phát triển phần mềm chúng ta sẽ sử dụng trong tương lai có lẽ sẽ khác so với các phương pháp hiện hành đang sử dụng. Tuy nhiên, một điều hiển nhiên là phát triển hướng đối tượng và các khái niệm cơ bản của nó đang được sử dụng rộng rãi. Nhiều trường học đã nhận ra được điều này và đã tạo ra những khoá học phát triển hệ thống hướng đối tượng như một phần chính yếu của hệ thống thông tin tin học hoá và các chương trình khoa học máy tính. Giáo trình này dự kiến sẽ cung cấp một kiến thức nền tảng về phát triển các hệ thống hướng đối tượng cho các đối tượng sinh viên những năm cuối. Mục tiêu của giáo trình là cung cấp một mô tả rõ ràng về các khái niệm nền tảng phát triển hệ thống hướng đối tượng. Trong đó, nhấn mạnh đến tính đơn giản của tiếp cận giúp sinh viên có kiến thức về UML có thể dễ dàng nắm bắt để phát triển một hệ thống hướng đối tượng.

Mục tiêu

Sau khi học xong môn học này sinh viên có thể:

- Hiểu các nguyên lý nền tảng của kỹ thuật hướng đối tượng và các khái niệm về sự trừu tượng, tính bao bọc, tính thừa kế, và tính đa hình.
- Hiểu về một số quy trình phát triển hệ thống, nội dung các giai đoạn cơ bản của một quy trình phát triển, và một số phương pháp phân tích thiết kế hướng đối tượng.
- Tiếp cận toàn bộ quy trình phát triển hệ thống sử dụng các kỹ thuật hướng đối tượng
- Sử dụng UML như là một công cụ mô hình hoá trong quá trình phát triển hệ thống
- Phát triển hệ thống từ các mô hình use case được xem như là một mô hình phân tích nhằm biểu diễn đầy đủ yêu cầu hệ thống.
- Áp dụng một quy trình lặp, tập trung vào kiến trúc để phát triển một mô hình thiết kế đủ chi tiết, đủ mạnh đáp ứng với các nhu cầu:
 - o Phù hợp với các yêu cầu hệ thống đã được thống nhất qua mô hình use case trong giai đoạn phân tích.
 - o Tái sử dụng.
 - o Dễ dàng để cài đặt hệ thống trong một ngôn ngữ và môi trường cụ thể.

PHẦN 1: TỔNG QUAN

Chương 1

GIỚI THIỆU VỀ PHƯƠNG PHÁP VÀ PHƯƠNG PHÁP LUẬN PHÁT TRIỂN HỆ THỐNG HƯỚNG ĐỐI TƯỢNG

Giới thiệu về phương pháp phát triển hướng chức năng

Đây là phương pháp cận truyền thống của ngành công nghiệp phần mềm trong đó quan điểm về phần mềm như là một tập hợp các chương trình (hoặc chức năng) và dữ liệu giả lập. Vậy chương trình là gì? Theo Niklaus Wirth, người tạo ra ngôn ngữ lập trình Pascal thì: “Chương trình = thuật giải + cấu trúc dữ liệu”. Điều này có nghĩa rằng có hai khía cạnh khác nhau của hệ thống được tiếp cận, hoặc tập trung vào các chức năng của hệ thống hoặc tập trung vào dữ liệu. Chúng ta chia hướng tiếp cận này thành hai thời kỳ: thời kỳ vào những năm thập niên 70, tiếp cận phân tích và thiết kế hệ thống theo phương pháp gọi là Descartes. Ý tưởng chính trong cách tiếp cận này là một quá trình lặp phân rã hệ thống thành các chức năng và ứng dụng phương pháp lập trình cấu trúc đơn thể chương trình, việc phân rã kết thúc khi một chức năng được phân rã có thể lập trình được. Trong thời kỳ này, người ta chưa quan tâm đến các thành phần không được tin học hoá mà chỉ xoay quanh đến các vấn đề trong hệ thống để lập trình, tập trung vào chức năng và ít tập trung vào dữ liệu (vì thời kỳ này đang chuẩn hoá và phát triển về cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu)

Thời kỳ vào những thập niên 80, tiếp cận phân tích thiết kế theo phương pháp gọi là *hệ thống*. Quan điểm chính của phương pháp này là tiếp cận hệ thống theo 2 thành phần, thành phần xử lý (thành phần động) và thành phần dữ liệu (thành phần tĩnh) của hệ thống. Cách tiếp cận của các phương pháp trong giai đoạn này tuân theo hai tính chất: tính toàn thể: tiếp cận hệ thống qua việc mô tả các hệ thống con và sự tương tác giữa chúng; tính đúng đắn: tìm kiếm sự phân rã, kết hợp các hệ thống con sao cho hành vi của nó tiêu biểu nhất của hệ thống trong môi trường tác động lên hệ thống con đó. Cách tiếp cận hệ thống theo hai thành phần chính là tiền đề cho cách tiếp cận hướng đối tượng trong các giai đoạn sau. Tuy nhiên, việc tiếp cận chủ yếu là hướng xoay quanh dữ liệu để thu thập và tổ chức dữ liệu nhằm khai thác mặt đáp ứng nhu cầu thông tin. Hướng tiếp cận gây khó khăn trong những hệ thống lớn và thường xuyên thay đổi cũng như là trong việc thiết kế nhằm tái sử dụng một thành phần đã có.

Giới thiệu về phương pháp phát triển hướng đối tượng

Vào thập niên 90, phương pháp tiếp cận phân tích thiết kế đối tượng là sự tổng hợp của phương pháp Descartes và phương pháp hệ thống. Trong khi các mô hình được đưa ra trong những thập niên trước thường đưa ra dữ liệu và xử lý theo hai hướng độc lập nhau. Khái niệm đối tượng là sự tổng hợp giữa khái niệm xử lý và khái niệm dữ liệu chung trong một cách tiếp cận, và một hệ thống là một tập hợp các đối tượng liên kết nội. Có nghĩa rằng việc xây dựng hệ thống chính là việc xác định các đối tượng đó bằng cách cố gắng ánh xạ các đối tượng của thế giới thực thành đối tượng hệ thống, thiết kế và xây dựng nó, và hệ thống hình thành chính là qua sự kết hợp của các đối tượng này. Phương pháp hướng đối tượng được xem là phương pháp phân tích thiết kế thể hệ thứ ba, các phương pháp tiêu biểu là OOD, HOOD, BON, OSA, ... và sau này là OOSA, OOA, OMT, CRC, OOM, OOAD, OOSE, RUP/UML

Đặc trưng cơ bản

- *Tính bao bọc (encapsulation)*: quan niệm mối quan hệ giữa đối tượng nhận và đối tượng cung cấp thông qua khái niệm hộp đen. Nghĩa là đối tượng nhận chỉ truy xuất đối tượng cung cấp qua giao diện được định nghĩa bởi đối tượng cung cấp, đối tượng nhận không được truy cập đến các đặc trưng được xem là “nội bộ” của đối tượng cung cấp.

- *Tính phân loại (classification)*: gom nhóm các đối tượng có cùng cấu trúc và hành vi vào một lớp (class).
- *Tính kết hợp (aggregation)*: kết hợp các đối tượng và các đối tượng cấu thành nó để mô tả cấu trúc cục bộ của đối tượng (ví dụ: toà nhà <-> phòng, xe <-> sườn xe, bánh xe,...) , hoặc sự liên kết phụ thuộc lẫn nhau giữa các đối tượng.
- *Tính thừa kế (heritage)*: phân loại tổng quát hoá và chuyên biệt hoá các đối tượng, và cho phép chia sẻ các đặc trưng của một đối tượng.

Phân loại

Phương pháp lập trình hướng đối tượng được chia thành 2 hướng như sau:

- Hướng lập trình: từ lập trình đơn thể chuyển sang lập trình hướng đối tượng với lý thuyết cơ bản dựa trên việc trừu tượng hóa kiểu dữ liệu.
- Hướng hệ quản trị CSDL: phát triển thành CSDL hướng đối tượng

Có 2 cách tiếp cận riêng biệt:

- Phương pháp kỹ thuật: hướng công nghệ phần mềm như OOD, HOOD, BON, BOOCH, MECANO, OODA,...
- Phương pháp toàn cục: hướng về HTTT như OOA, OOSA, OOAD, OMT, OOM,...

Ưu điểm

- Cấu trúc hoá được các cấu trúc phức tạp và sử dụng được cấu trúc đệ quy: các phương pháp đối tượng đều sử dụng các mô hình bao gồm nhiều khái niệm để biểu diễn nhiều ngữ nghĩa khác nhau của hệ thống. Ví dụ: trong mô hình lớp của OMT có khái niệm mỗi kết hợp thành phần cho phép mô tả một đối tượng là một thành phần của đối tượng khác, trong khi nếu dùng mô hình ER truyền thống không có khái niệm này do đó không thể biểu diễn được quan hệ thành phần.
- Xác định được đối tượng của hệ thống qua định danh đối tượng¹
- Tính thừa kế được đưa ra tạo tiền đề cho việc tái sử dụng

Mô hình

Mô hình (model) là một dạng thức trừu tượng về một hệ thống, được hình thành để hiểu hệ thống trước khi xây dựng hoặc thay đổi hệ thống đó. Theo Efraim Turban, mô hình là một dạng trình bày đơn giản hoá của thế giới thực. Bởi vì, hệ thống thực tế thì rất phức tạp và rộng lớn và khi tiếp cận hệ thống, có những chi tiết những mức độ phức tạp không cần thiết phải được mô tả và giải quyết. Mô hình cung cấp một phương tiện (các khái niệm) để quan niệm hoá vấn đề và giúp chúng ta có thể trao đổi các ý tưởng trong một hình thức cụ thể trực quan, không mơ hồ.

Các đặc điểm của một mô hình:

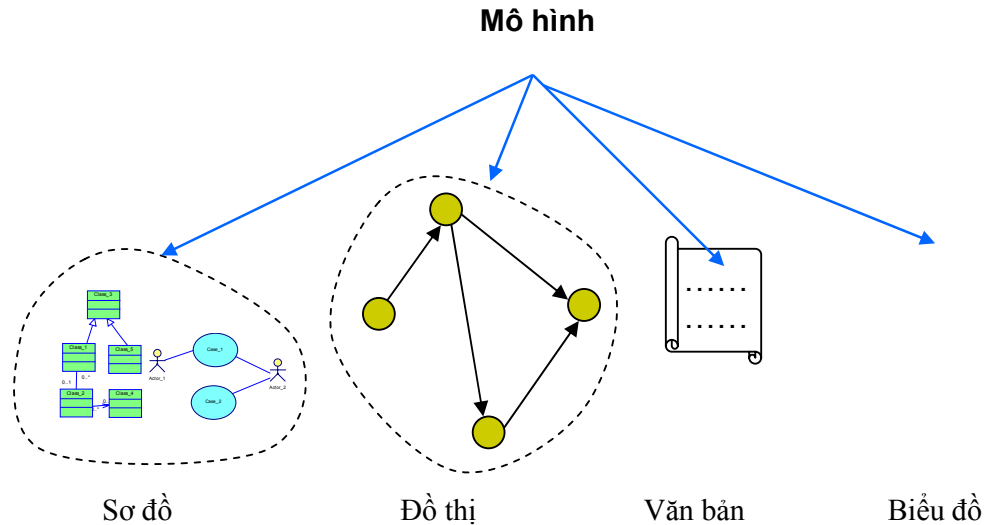
- Diễn đạt một mức độ trừu tượng hóa (ví dụ: mức quan niệm, mức tổ chức, mức vật lý,...)
- Tuân theo một quan điểm (quan điểm của người mô hình hoá)
- Có một hình thức biểu diễn (văn bản, đồ họa: sơ đồ, biểu đồ, đồ thị,...)

Hầu hết các kỹ thuật mô hình hóa sử dụng trong phân tích thiết kế là các ngôn ngữ đồ họa (đa số là sơ đồ - diagram), các ngôn ngữ này bao gồm một tập hợp các ký hiệu. Các ký hiệu này

¹ OID: Object Identifier

được dùng đi kèm theo các quy tắc của phương pháp luận giúp cho việc trao đổi các quan hệ thông tin phức tạp được rõ ràng hơn việc mô tả bằng văn bản.

Ví dụ :



Mô hình tĩnh và mô hình động

Mô hình tĩnh (static model): được xem như là hình ảnh về thông số hệ thống tại một thời điểm xác định. Các mô hình tĩnh được dùng để trình bày cấu trúc hoặc những khía cạnh tĩnh của hệ thống.

Mô hình động (dynamic model): được xem như là một tập hợp các hành vi, thủ tục kết hợp với nhau để mô tả hành vi của hệ thống. Các mô hình động được dùng để biểu diễn sự tương tác của các đối tượng để thực hiện công việc hệ thống.

Mục đích của mô hình hoá

Đứng trước sự gia tăng mức độ phức tạp của một hệ thống, việc trực quan hoá và mô hình hóa ngày càng trở nên chính yếu trong cách tiếp cận về một hệ thống, đặc biệt là cách tiếp cận hướng đối tượng. Việc sử dụng các ký hiệu để trình bày hoặc mô hình hóa bài toán có các mục đích sau:

- Làm sáng tỏ vấn đề: chúng ta có thể đưa ra được các lỗi hoặc các thiếu sót của hệ thống từ việc tiếp cận trực quan đồ họa hơn là các dạng trình bày khác như văn bản, đoạn mã,... Hơn nữa, việc mô hình hoá giúp chúng ta dễ dàng hiểu được hệ thống.
- Mô phỏng được hình ảnh tương tự của hệ thống: hình thức trình bày của mô hình có thể đưa ra được một hình ảnh giả lập như hoạt động thực sự của hệ thống thực tế, điều này giúp cho người tiếp cận cảm thấy thuận tiện khi làm việc với mô hình (là hình ảnh thu nhỏ của hệ thống thực tế)
- Gia tăng khả năng duy trì hệ thống: các ký hiệu trực quan có thể cải tiến khả năng duy trì hệ thống. Thay đổi các vị trí được xác định trực quan và việc xác nhận trực quan trên mô hình các thay đổi đó sẽ giảm đi các lỗi. Do đó, chúng ta có thể tạo ra các thay đổi nhanh hơn và các lỗi được kiểm soát hoặc xảy ra ít hơn.
- Làm đơn giản hóa vấn đề: mô hình hoá có thể biểu diễn hệ thống ở nhiều mức, từ mức tổng quát đến mức chi tiết, mức càng tổng quát thì ký hiệu sử dụng càng ít (do đó càng đơn giản hoá việc hiểu) và hệ thống được biểu diễn càng tổng quát.

Phương pháp luận phát triển hệ thống

Phương pháp luận phát triển hệ thống bao gồm hai thành phần :

- Quy trình (process) : bao gồm các giai đoạn (phase) và tiến trình trong đó định nghĩa thứ tự các giai đoạn và các luật hình thành nên một quá trình phát triển hệ thống từ các công việc khởi tạo đến các công việc kết thúc của một dự án hệ thống.
- Các khái niệm (notation), phương pháp : các mô hình (bao gồm các phương pháp mô hình hoá của mô hình) cho phép mô hình hoá các kết quả của quá trình phát triển hệ thống.

Các giai đoạn cơ bản trong một quy trình :

Để tự động hóa hoạt động xử lý, hệ thống phải trải qua một quá trình gồm nhiều bước được gọi là quá trình phát triển hệ thống. Cũng giống như nhiều tiến trình khác, phát triển hệ thống tự động cũng theo chu trình được gọi là vòng đời (Life cycle). Khái niệm vòng đời là một khái niệm rộng nó bắt đầu từ sự khởi đầu xây dựng cho đến kết thúc việc khai thác hệ thống. Nếu chúng ta chỉ chú trọng đến giai đoạn xây dựng và triển khai thì gọi là phát triển hệ thống. Vòng đời phát triển hệ thống - SDLC (Systems Development Life Cycle) là một phương pháp luận chung để phát triển nhiều loại hình hệ thống khác nhau. Tuy nhiên, các giai đoạn trong quá trình này cũng thay đổi khác nhau khoảng từ 3 cho đến 20 tùy theo qui mô và loại hình hệ thống chúng ta đang tiếp cận.

Phần sau đây sẽ giới thiệu các giai đoạn cơ bản làm nền tảng chung cho hầu hết quá trình phát triển hệ thống:

Giai đoạn khởi tạo

Hoạt động chính của giai đoạn này là khảo sát tổng quan hệ thống, vạch ra các vấn đề tồn tại trong hệ thống và các cơ hội của hệ thống, cũng như trình bày lý do tại sao hệ thống nên hoặc không nên được đầu tư phát triển tự động hóa. Một công việc quan trọng tại thời điểm này là xác định phạm vi của hệ thống đề xuất, trường dự án và nhóm phân tích viên ban đầu cũng lập một kế hoạch các hoạt động của nhóm trong các giai đoạn tiếp theo của dự án phát triển hệ thống. Kế hoạch này xác định thời gian và nguồn lực cần thiết. Đánh giá khả thi của dự án và nhất là phải xác định được chi phí cần phải đầu tư và lợi ích mang lại từ hệ thống. Kết quả của giai đoạn này là xác định được dự án hoặc được chấp nhận để phát triển, hoặc bị từ chối, hoặc phải định hướng lại.

Giai đoạn phân tích

Giai đoạn phân tích bao gồm các bước sau:

- Thu thập yêu cầu hệ thống: các phân tích viên làm việc với người sử dụng để xác định tất cả những gì mà người dùng mong muốn từ hệ thống đề xuất.
- Nguyên cứu các yêu cầu và cấu trúc hoá (mô hình hoá) để dễ dàng nhận biết và loại bỏ những yếu tố dư thừa.
- Phát sinh các phương án thiết kế chọn lựa phù hợp với yêu cầu và so sánh các phương án này để xác định giải pháp nào là đáp ứng tốt nhất các yêu cầu trong một mức độ cho phép về chi phí, nhân lực, và kỹ thuật của tổ chức. Kết quả của giai đoạn này là bản mô tả về phương án được chọn.

Trong phân tích hướng đối tượng giai đoạn này quan tâm đến mức độ trừu tượng hoá đầu tiên bằng cách xác định các lớp và các đối tượng đóng vai trò quan trọng nhằm diễn đạt các yêu cầu cũng như mục tiêu hệ thống. Để hiểu rõ các yêu cầu hệ thống chúng ta cần xác định ai là người dùng và là tác nhân hệ thống. Trong phương pháp phát triển hướng đối tượng cũng như phương pháp truyền thống, các mô tả kịch bản hoạt động được sử dụng để trợ giúp các phân tích viên hiểu được yêu cầu. Tuy nhiên, các kịch bản này có thể được mô tả không đầy đủ hoặc không theo một hình thức. Do đó, khái niệm use case được dùng trong giai đoạn này

nhằm biểu diễn chức năng hệ thống và sự tương tác người dùng hệ thống. Các kịch bản hoạt động lúc này sử dụng các mô hình động (dynamic diagram) nhằm mô tả nội dung của use case để làm rõ sự tương tác giữa các đối tượng, vai trò cũng như sự cộng tác của các đối tượng trong hoạt động của use case hệ thống. Trong giai đoạn phân tích, chỉ có các lớp tồn tại trong phạm vi hệ thống (ở thế giới thực) mới được mô hình hoá và như vậy thì kết quả mô hình hoá trong giai đoạn này sẽ phản ánh phạm vi của hệ thống. Các lớp về kỹ thuật, giao diện định nghĩa phần mềm cũng không quan tâm ở giai đoạn này.

Giai đoạn thiết kế

Trong giai đoạn này kết quả của giai đoạn phân tích sẽ được chi tiết hoá để trở thành một giải pháp kỹ thuật để thực hiện. Các đối tượng và các lớp mới được xác định để bổ sung vào việc cài đặt yêu cầu và tạo ra một hạ tầng cơ sở kỹ thuật về kiến trúc. Ví dụ: các lớp mới này có thể là lớp giao diện (màn hình nhập liệu, màn hình hỏi đáp, màn hình duyệt,...). Các lớp thuộc phạm vi vấn đề có từ giai đoạn phân tích sẽ được "nhúng" vào hạ tầng cơ sở kỹ thuật này, tạo ra khả năng thay đổi trong cả hai phương diện: Phạm vi vấn đề và hạ tầng cơ sở. Giai đoạn thiết kế sẽ đưa ra kết quả là bản đặc tả chi tiết cho giai đoạn xây dựng hệ thống.

Về mức độ thiết kế thì có thể chia kết quả của giai đoạn này thành hai mức:

Thiết kế luận lý

Đặc tả hệ thống ở mức độ trừu tượng hóa dựa trên kết quả của giải pháp được chọn lựa từ giai đoạn phân tích. Các khái niệm và mô hình được dùng trong giai đoạn này độc lập với phần cứng, phần mềm sẽ sử dụng và sự chọn lựa cài đặt. Theo quan điểm lý thuyết, ở bước này hệ thống có thể cài đặt trên bất kỳ nền tảng phần cứng và hệ điều hành nào, điều này cho thấy giai đoạn này chỉ tập trung để biểu diễn khía cạnh hành vi và tính năng của đối tượng hệ thống.

Thiết kế vật lý

Chuyển đổi kết quả thiết kế luận lý sang các đặc tả trên phần cứng, phần mềm và kỹ thuật đã chọn để cài đặt hệ thống. Cụ thể là đặc tả trên hệ máy tính, hệ quản trị cơ sở dữ liệu, ngôn ngữ lập trình đã chọn,... Kết quả của bước này là các đặc tả hệ thống vật lý sẵn sàng chuyển cho các lập trình viên hoặc những người xây dựng hệ thống khác để lập trình xây dựng hệ thống.

Giai đoạn xây dựng

Trong giai đoạn xây dựng (giai đoạn lập trình), các lớp của giai đoạn thiết kế sẽ được biến thành những dòng mã lệnh (code) cụ thể trong một ngôn ngữ lập trình hướng đối tượng (không nên dùng một ngôn ngữ lập trình hướng chức năng!). Phụ thuộc vào khả năng của ngôn ngữ được sử dụng, đây có thể là một công việc khó khăn hoặc dễ dàng. Khi tạo ra các mô hình phân tích và thiết kế trong UML, tốt nhất nên cố gắng né tránh việc ngay lập tức biến đổi các mô hình này thành các dòng mã lệnh. Trong những giai đoạn trước, mô hình được sử dụng để dễ hiểu, dễ giao tiếp và tạo nên cấu trúc của hệ thống; vì vậy, vội vàng đưa ra những kết luận về việc viết mã lệnh có thể sẽ thành một trở ngại cho việc tạo ra các mô hình chính xác và đơn giản. Giai đoạn xây dựng là một giai đoạn riêng biệt, nơi các mô hình được chuyển thành các mã lệnh.

Giai đoạn thử nghiệm

Một hệ thống phần mềm thường được thử nghiệm qua nhiều giai đoạn và với nhiều nhóm thử nghiệm khác nhau. Các nhóm sử dụng nhiều loại biểu đồ UML khác nhau làm nền tảng cho công việc của mình: Thử nghiệm đơn vị sử dụng biểu đồ lớp (class diagram) và đặc tả lớp, thử nghiệm tích hợp thường sử dụng biểu đồ thành phần (component diagram) và biểu đồ cộng tác (collaboration diagram), và giai đoạn thử nghiệm hệ thống sử dụng biểu đồ Use case

(use case diagram) để đảm bảo hệ thống có phương thức hoạt động đúng như đã được định nghĩa từ ban đầu trong các biểu đồ này.

Giai đoạn cài đặt và bảo trì

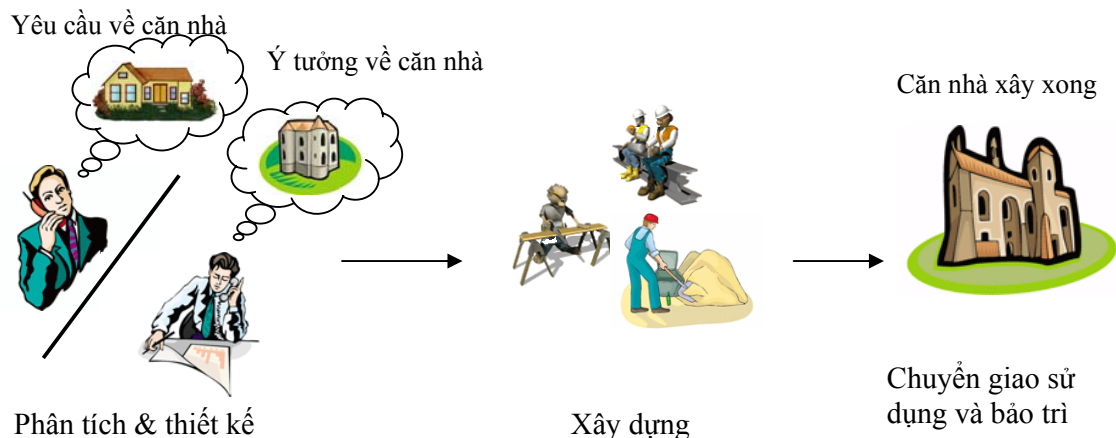
Điều chỉnh hệ thống phù hợp với nhu cầu sử dụng, các thay đổi phát sinh bao gồm:

- Chức năng sử dụng chưa phù hợp tốt nhất với người sử dụng hoặc khó sử dụng
- Các điều kiện và yêu cầu của người dùng hệ thống thay đổi, đòi hỏi phải chỉnh sửa sao cho hệ thống vẫn hữu dụng
- Các lỗi hệ thống phát sinh do quá trình kiểm tra còn sót lại
- Nâng cấp phiên bản mới của hệ thống

Bảo trì hệ thống không nên xem như là một giai đoạn tách rời mà nên xem như là một sự lặp lại chu trình của những giai đoạn trước đòi hỏi phải được nghiên cứu đánh giá và cài đặt. Tuy nhiên, nếu một hệ thống không còn hoạt động như mong muốn do có sự thay đổi quá lớn về hoạt động, hoặc nhu cầu mới đặt ra vượt quá sự giải quyết của hệ thống hiện tại, hoặc chi phí để bảo trì là quá lớn. Lúc này yêu cầu về hệ thống mới được xác lập để thay thế hệ thống hiện tại và một quy trình lại bắt đầu.

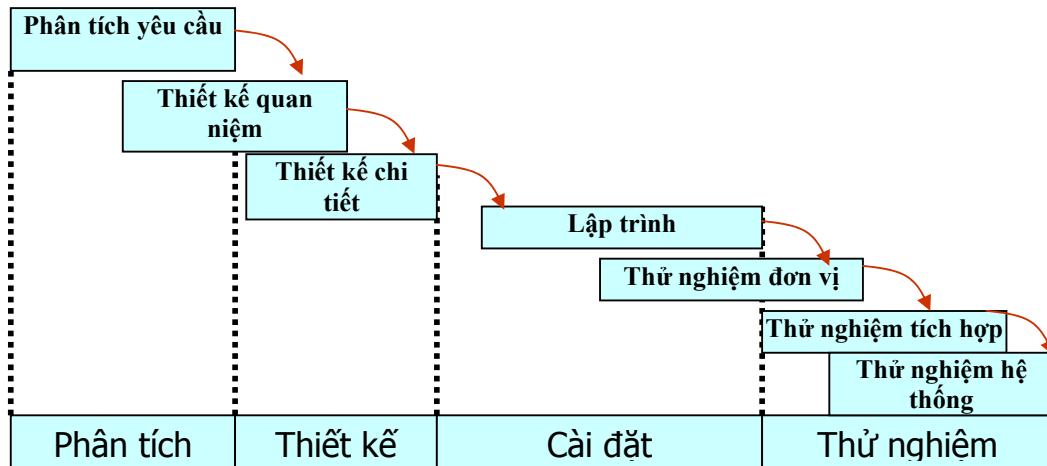
Ví dụ về quy trình phát triển

Chúng ta có thể hình dung rằng chúng ta muốn xây dựng một căn nhà. Công việc đầu tiên là chúng ta chắc chắn là chúng ta dự tính xem chúng ta sẽ bỏ số tiền bao nhiêu để xây dựng căn nhà này, dựa trên số tiền này chúng ta tìm kiếm và phác họa (có thể chỉ trong ý tưởng) căn nhà này phải như thế nào? Loại căn nhà theo kiểu gì, có mấy phòng, chiều rộng và chiều dài bao nhiêu, rồi nào đến nền nhà, màu sắc, tiện nghi?... Rồi sau đó, chúng ta sẽ chọn một đơn vị xây dựng (trong số nhiều đơn vị mà thỏa yêu cầu nhất). Tất cả các yêu trên sẽ phải trao đổi với đơn vị xây dựng này nhằm thống nhất về giá cả cũng như các điều khoản về yêu cầu xây dựng. Giai đoạn này được xem như là giai đoạn phân tích. Tiếp đó, đơn vị xây dựng sẽ thực hiện công việc thiết kế chi tiết của căn nhà, và từng đơn vị trong căn nhà (phòng, tường, trần, mái, phòng khách, phòng ăn, phòng ngủ,...). Giai đoạn này được xem là giai đoạn thiết kế. Sau đó các bản thiết kế chi tiết của căn nhà sẽ được bộ phận thi công dựa vào đó để tiến hành việc xây dựng. Giai đoạn này được xem là giai đoạn xây dựng. Căn nhà sau khi hoàn tất sẽ được chuyển giao để sử dụng, tất nhiên trong quá trình sử dụng nếu có các hư hỏng thì đơn vị xây dựng sẽ phải tiến hành bảo trì và sửa chữa.



Mô số qui trình phát triển

Qui trình thác nước (waterfall – Boehm 1970)

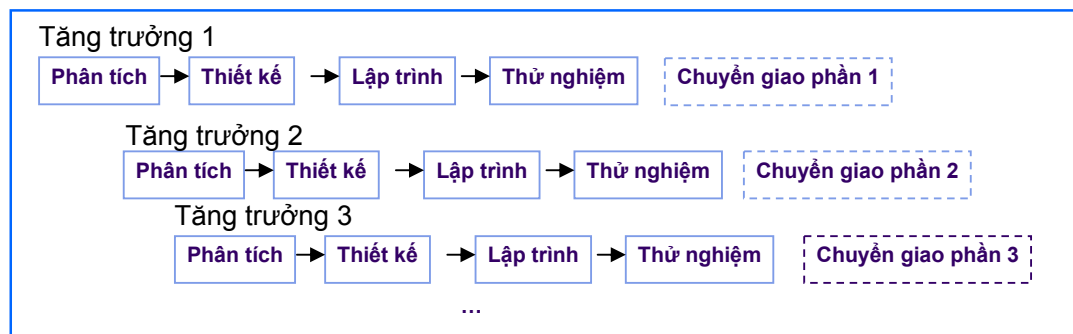


Đây là một qui trình đầu tiên được đề xuất và đã đưa ra được các giai đoạn căn bản nhất và đầy đủ cho một quá trình phát triển hệ thống, các giai đoạn bao gồm : phân tích, thiết kế, cài đặt và thử nghiệm hệ thống. Từ khi được đề xuất qui trình này nhanh chóng được phổ cập sử dụng rộng rãi trong giới công nghiệp và cho đến bây giờ đã có nhiều cải tiến hoàn thiện.

Nhược điểm :

- Qui trình là các giai đoạn tuần tự nối tiếp nhau, có nghĩa là giai đoạn phân tích phải được hoàn thành rồi đến giai đoạn thiết kế,... không cho phép sự quay lui và do đó, khi áp dụng qui trình này sẽ khó khăn khi ở giai đoạn trước có sự thay đổi (do sai sót, do nhu cầu người dùng thay đổi hoặc do có sự tiến hoá hệ thống,...).

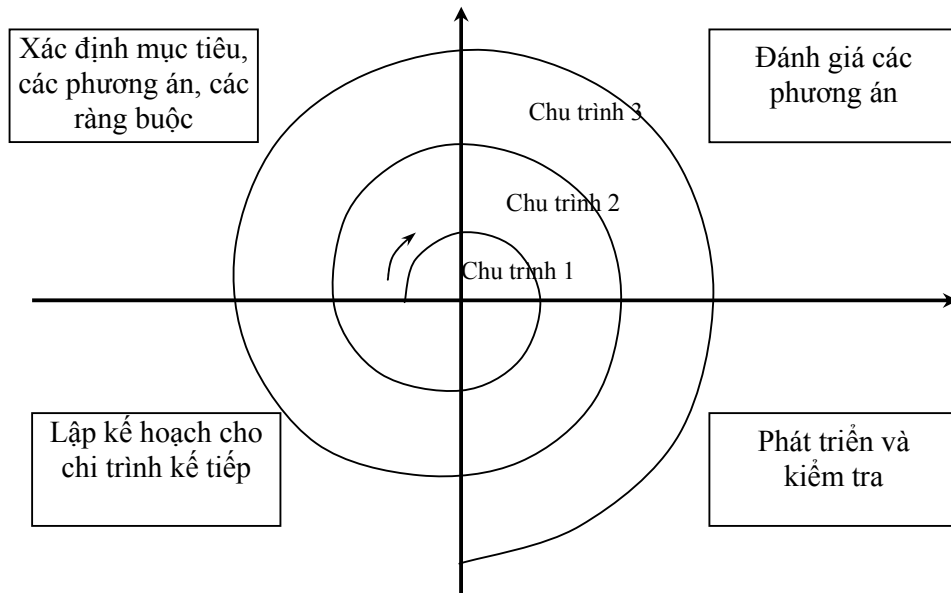
Qui trình tăng trưởng (D.R. Graham 1988)



- Quan điểm chính của qui trình này là phát triển từng phần (phần hệ con) của hệ thống dùng qui trình thác nước.
- Lập : phân chia hệ thống thành những phần có thể phát triển một cách độc lập. Mỗi thành phần trong quá trình phát triển sẽ được áp dụng qui trình thác và được xem như một tăng trưởng của hệ thống. Khi thành phần cuối cùng hoàn tất thì quá trình phát triển toàn bộ hệ thống kết thúc.

Nhược điểm : qui trình này không thể áp dụng cho những hệ thống có sự phân chia không rõ ràng hoặc không thể phân chia thành những thành phần tác biệt.

Qui trình xoắn ốc (Boehm 88)

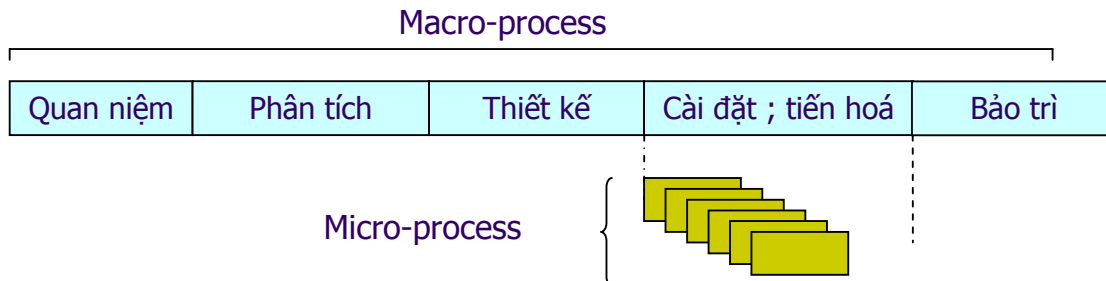


Là một quá trình gồm nhiều vòng lặp dựa trên bốn giai đoạn :

- Giai đoạn 1 :
 - o Đối với vòng lặp đầu tiên : phân tích yêu cầu
 - o Từ vòng lặp thứ hai trở đi : thiết lập mục tiêu cho vòng lặp, xác định các phương án để đạt mục tiêu đó ; các ràng buộc xuất phát từ các kết quả của các vòng lặp trước.
- Giai đoạn 2 :
 - o Đánh giá các phương án dựa trên các sản phẩm đạt được và tiến trình thực thi phương án.
 - o Xác định và giải quyết các rủi ro.
- Giai đoạn 3 :
 - o Phát triển và kiểm tra sản phẩm.
- Giai đoạn 4 :
 - o Lập kế hoạch cho vòng lặp tiếp theo.

Qui trình xoắn ốc cũng có thể áp dụng qui trình khác, ví dụ giai đoạn 3 có thể được thực hiện áp dụng qui trình thác nước.

Qui trình Booch (1996)



Gồm hai tiến trình :

- Macro process : đóng vai trò như là bộ khung của micro process và bao phủ toàn bộ phạm vi dự án. Công việc chính của macro process là liên quan đến quản lý kỹ thuật của hệ thống trong việc chú trọng đến yêu cầu của người dùng và thời gian hoàn thành sản phẩm mà ít quan tâm đến chi tiết thiết kế hệ thống. Macro process gồm :