

Công nghệ phần mềm

Bùi Đức Tiến, TS., GVCC.

Tel. 091 351 4311

Email. TienBuiDuc@gmail.com

MỤC ĐÍCH



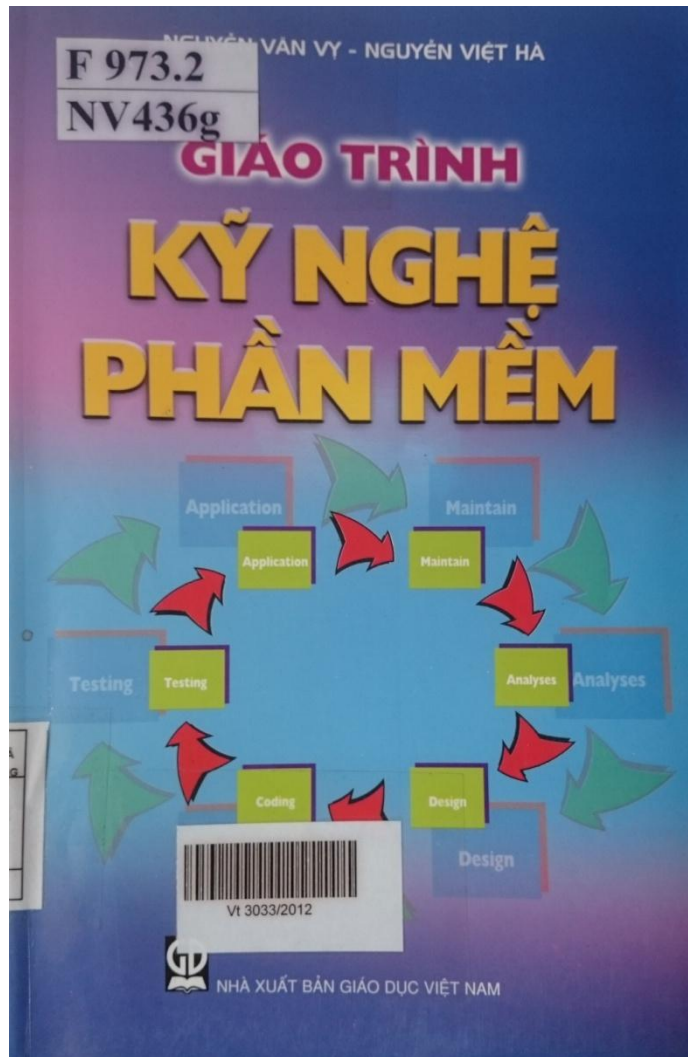
- ❧ Giúp sinh viên có được kiến thức cơ bản nhất trong lĩnh vực công nghệ phần mềm
- ❧ có hiểu biết sâu sắc về qui trình phát triển phần mềm
- ❧ Nắm vững những yêu cầu công việc cần phải làm ở mỗi giai đoạn của qui trình, những phương pháp, công cụ của từng giai đoạn
- ❧ có khả năng vận dụng trong các dự án phần mềm .

YÊU CẦU

- ∞ Biết làm việc nhóm
- ∞ Biết áp dụng quy trình vào giải quyết Bài tập lớn của nhóm
- ∞ Sv cần:
 - ∞ Nghiên cứu trước giáo trình, tài liệu học tập;
 - ∞ Tham gia đầy đủ các giờ lên lớp;
 - ∞ Làm bài tập



TÀI LIỆU THAM KHẢO



🌀 **Giáo trình chính:**

Nguyễn Văn Vy, Nguyễn Việt Hà (2010), *Giáo trình kỹ nghệ phần mềm*. NXB Giáo dục.

🌀 Tài liệu tham khảo:

🌀 TS. Lê Văn Phụng. Kỹ nghệ phần mềm, NXB TT&TT, 2014

🌀 TS. Lê Văn Phụng, CN. Lê Hương Giang, Kỹ nghệ phần mềm nâng cao, NXB TT&TT, 2015

🌀 Ian Sommerville (2011), *Software Engineering (9th edition)*. Addison-Wesley Publishing

🌀 Kho tài liệu trên mạng:

https://drive.google.com/drive/folders/1pwhALHM-ByOF_zsvZCm6cxpANTIxAYSj?usp=sharing

Tổng quan

- Phân tích hệ thống và ý tưởng phần mềm
- Thiết kế và cài đặt phần mềm
- Đảm bảo, kiểm chứng và duy trì tính toàn vẹn phần mềm
- Tự động hóa các hoạt động phát triển phần mềm
- Vấn đề quản lý dự án phần mềm

ĐỀ CƯƠNG

- Chương 1. phần mềm và kỹ nghệ phần mềm
- Chương 2. Xác định và đặc tả yêu cầu phần mềm.
- Chương 3. Thiết kế phần mềm
- Chương 4. Lập trình
- Chương 5. Xác minh và thẩm định
- Phụ lục. Hướng dẫn làm bài tập lớn

Chương 1.

phần mềm và kỹ nghệ phần mềm

1.1. khái niệm về phần mềm

- 1.1.1. khái niệm. Các phần mềm thường được mô tả với 3 bộ phận cấu thành:
 - Tập các lệnh
 - Cấu trúc dữ liệu (lưu trữ trên các bộ nhớ)
 - Các tài liệu

Chương 1.

phần mềm và kỹ nghệ phần mềm

1.1. khái niệm về phần mềm

- 1.1.2. các đặc trưng của phần mềm

- Phần mềm được kỹ nghệ, không được chế tạo theo nghĩa cổ điển
- Phần mềm không “hỏng đi” nhưng “thoái hóa” theo thời gian
- Phần mềm vẫn được xđ theo đơn đặt hàng
- Sự phức tạp và tính thay đổi luôn là bản chất của phần mềm
- Ngày nay các phần mềm được phát triển theo nhóm

Chương 1.

phần mềm và kỹ nghệ phần mềm

1.1. khái niệm về phần mềm

- 1.1.3. các thành phần phần mềm
 - Thành phần máy thực hiện được
 - » Các câu lệnh được xây dựng trên 1 ngôn ngữ lập trình
 - » Tính sử dụng lại của phần mềm
 - Thành phần máy không thực hiện được
 - » Các giải thích trong các chương trình phần mềm
 - » Các tài liệu phân tích, thiết kế, các hướng dẫn sử dụng

Chương 1.

phần mềm và kỹ nghệ phần mềm

1.1. khái niệm về phần mềm

- 1.1.4. phân loại các phần mềm

- Phân loại theo mức độ hoàn thiện
 - » Phần mềm đơn lẻ
 - » Phần mềm mang tính hệ thống
- Theo vai trò – chức năng phần mềm thực hiện
 - » Phần mềm hệ thống
 - » Phần mềm công cụ
 - » Phần mềm ứng dụng

- Theo lĩnh vực được ứng dụng
 - » Phần mềm hệ thống
 - » Phần mềm thời gian thực
 - » Phần mềm nghiệp vụ
 - » Phần mềm khoa học công nghệ
 - » Phần mềm nhúng
 - » Phần mềm máy tính cá nhân
 - » Phần mềm trên nền WEB
 - » Phần mềm trí tuệ nhân tạo

Chương 1.

phần mềm và kỹ nghệ phần mềm

1.2. sự phát triển của phần mềm và thách thức

- 1.2.1. quá trình tiến hóa của phần mềm

- 1950-1960

- » Sx mang tính đơn chiếc

- 1960- giữa 1970

- » Hệ thống phần mềm đa chương trình, đa NSD

- » Tương tác người-máy

- » Hệ thống thời gian thực

- » Tiến bộ lưu trữ

- » Thư viện phần mềm, quy mô phần mềm lớn

- Giữa 1970-1990

- Mạng cục bộ và mạng toàn cầu

- Máy tính cá nhân ra đời

- Chi phí phần cứng giảm, phần mềm tăng

- Phần mềm hướng cấu trúc hoàn thiện

- Công cụ trợ giúp phát triển phần mềm

- Sau 1990

- Công nghệ hướng đối tượng

- Các hệ thông minh

- Internet, các hệ thống WEB Base,

- Sử dụng lại

Chương 1.

phần mềm và kỹ nghệ phần mềm

- 1.2. sự phát triển của phần mềm và thách thức
 - 1.2.2. khủng hoảng phần mềm và sự ra đời của kỹ nghệ phần mềm. v/đ:
 - Tăng quy mô của phần mềm → Tăng chi phí, Tăng thời gian phát triển
 - Phụ thuộc kinh nghiệm của người phát triển phần mềm.
 - Chất lượng phần mềm không ổn định
 - Thiếu kỹ sư phần mềm
 - Gánh nặng bảo trì

Chương 1.

phần mềm và kỹ nghệ phần mềm

- 1.2. sự phát triển của phần mềm và thách thức
 - 1.2.3. thách thức đối với phát triển phần mềm hiện nay
 - Phần cứng phát triển nhanh, Tinh vi và năng lực cao
 - Quy mô và độ phức tạp phần mềm tăng → tốn tài nguyên, khó phát triển nhanh, khó bảo trì
 - ➔ cần phải có kỹ nghệ phần mềm

Chương 1.

phần mềm và kỹ nghệ phần mềm

- 1.3. kỹ nghệ phần mềm

- 1.3.1. khái niệm

- Định nghĩa. Fritz Bauer: “kỹ nghệ phần mềm là việc thiết lập và ứng dụng đúng đắn các nguyên lý công nghệ để thu được phần mềm một cách kinh tế, vừa tin cậy, vừa làm việc hiệu quả trên các máy tính thực”
 - Pamas (1987): “việc xây dựng phần mềm nhiều phiên bản bởi nhiều người”
 - Pressman: “là bộ môn tích hợp cả quy trình, các phương pháp, các công cụ để phát triển phần mềm máy tính”
 - ➔ kỹ nghệ phần mềm là kỹ nghệ phát triển một hệ thống gồm 3 yếu tố chủ yếu:
 - Phương pháp
 - Công cụ
 - Thủ tục

Chương 1.

phần mềm và kỹ nghệ phần mềm

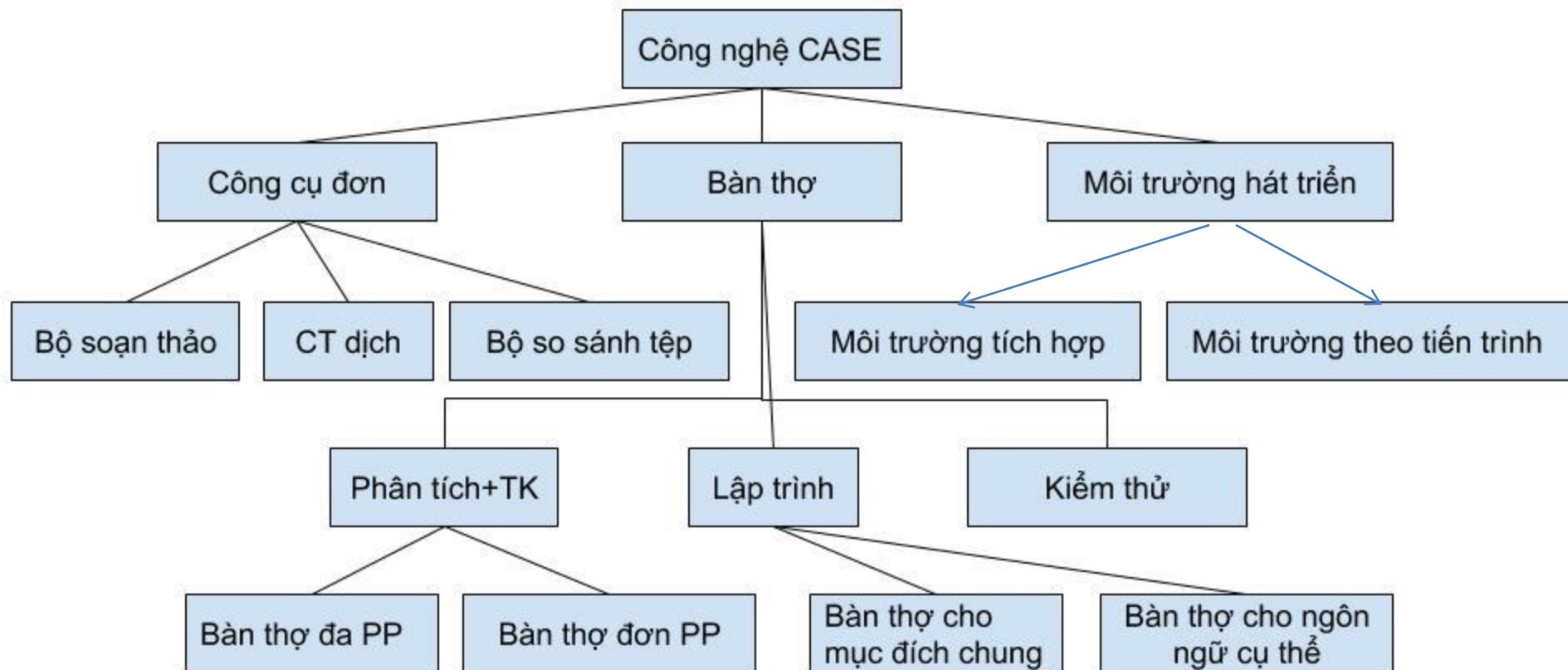
- 1.3. kỹ nghệ phần mềm
 - 1.3.2. các phương pháp trong kỹ nghệ phần mềm
 - Phương pháp luận
 - Các hoạt động theo 1 định hướng công nghệ và trình tự thực hiện: định hướng cấu trúc, định hướng dữ liệu, định hướng đối tượng...
 - Phương pháp cụ thể.
 - Là cách thức giải quyết các vấn đề cụ thể phát sinh khi thực hiện các nv khác nhau trong các gđ khác nhau

Chương 1.

phần mềm và kỹ nghệ phần mềm

- 1.3. 3. các công cụ trong kỹ nghệ phần mềm
 - Công cụ đơn: cho phép thực hiện 1/ 1 số chức năng trọn vẹn
 - Bàn thợ (Workbenches): thông tin do ta tạo ra có thể được dùng cho các công cụ khác hay cho các giai đoạn tiếp theo của quá trình phát triển.
 - CASE (Computer Added Software Engineering): tên gọi chung cho các phần mềm trợ giúp tạo ra các phần mềm khác
 - UML (Unified Modeling Language): ngôn ngữ mô hình hóa thống nhất dùng cho tất cả các gđ phát triển phần mềm hướng đối tượng. Các công cụ được xây dựng trên UML: Rational Rose, Power Designer...

Phân loại công cụ phát triển phần mềm



1.3.4. các thủ tục trong CNFM

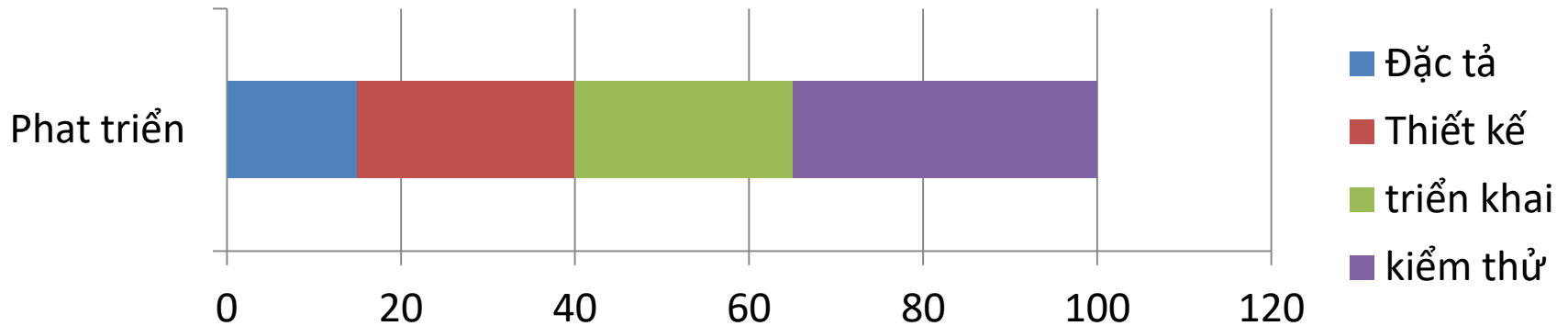
- Các thủ tục là chất keo dán các pp và công cụ lại để chúng được sử dụng hợp lí và đúng hạn.
- Các thủ tục thường gồm:
 - Xđ trình tự các PP áp dụng cho mỗi dự án
 - Định ra các sản phẩm cần bàn giao, các chuẩn, các độ đo để đánh giá, kiểm soát chất lượng...
 - XĐ mốc bàn giao sản phẩm

1.3.5. tiến trình phần mềm

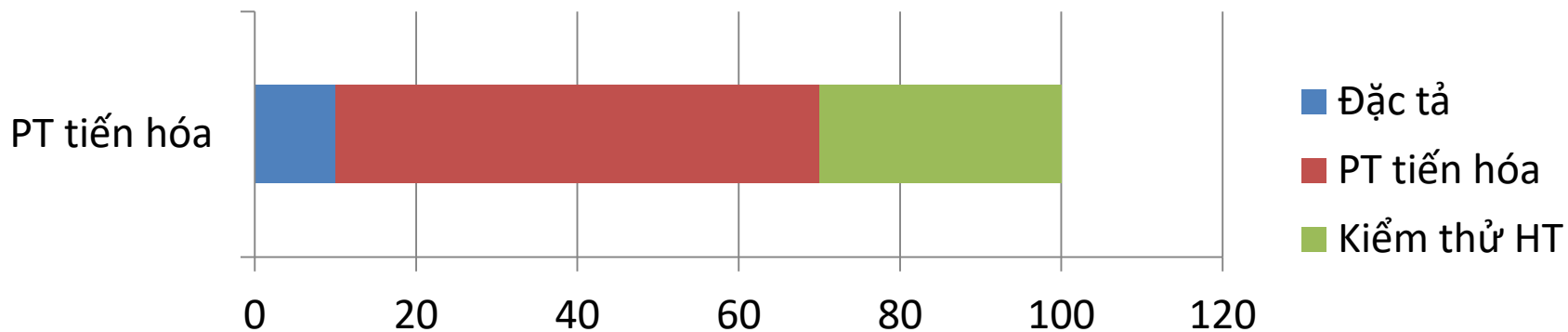
Tiến trình phần mềm là 1 loạt các hoạt động dự kiến trước theo 1 lộ trình. Thường gồm các bước:

- Xđ yêu cầu
 - Phát triển phần mềm
 - Thẩm định phần mềm
 - Tiến hóa phần mềm
-
- The diagram uses purple curly braces to group the steps. The first three steps (Xđ yêu cầu, Phát triển phần mềm, and Thẩm định phần mềm) are grouped under the label 'Trong xưởng'. The last step (Tiến hóa phần mềm) is grouped under the label 'đưa vào ứng dụng'.
- Trong xưởng
- đưa vào ứng dụng

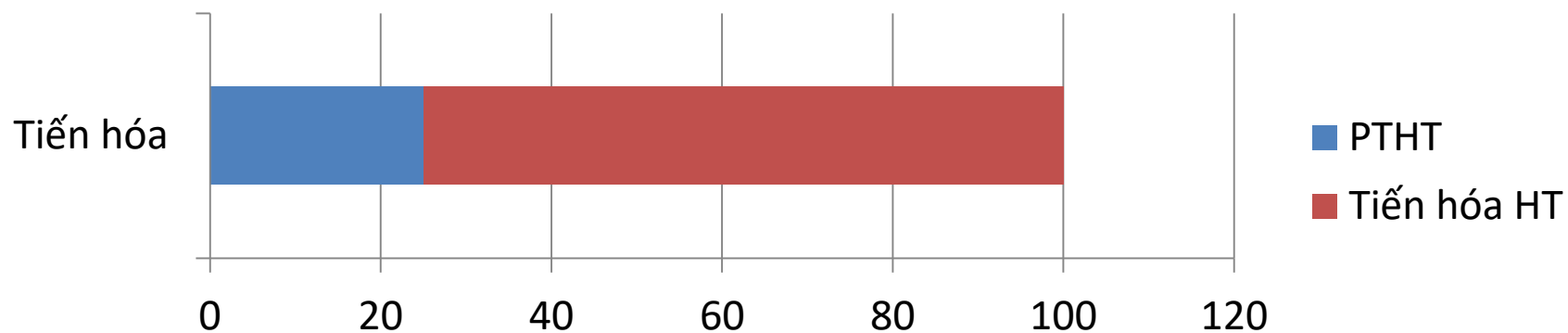
1.3.6. chi phí xây dựng phần mềm



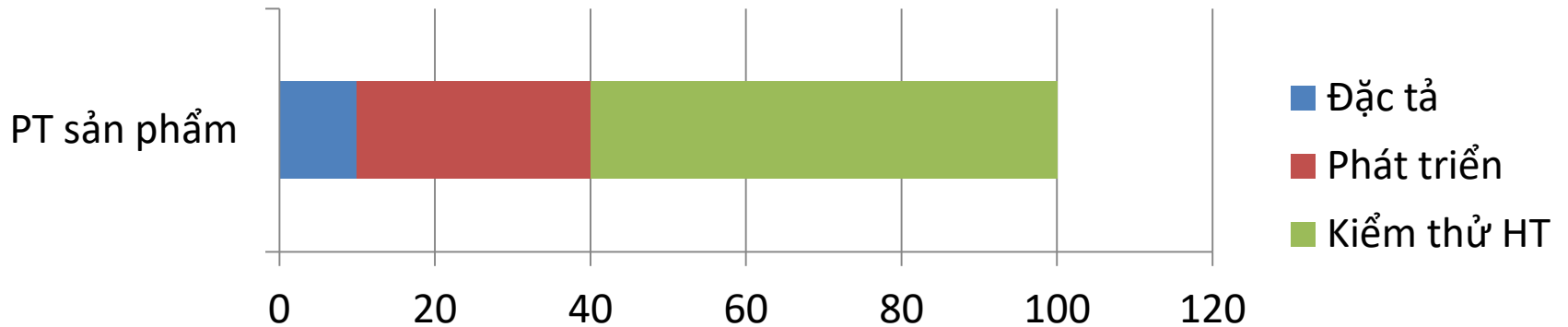
1.3.6. chi phí phát triển tiến hóa



1.3.6. chi phí tiến hóa



1.3.6. chi phí phát triển sản phẩm chung



1.3.7. các thuộc tính của phần mềm được kĩ nghệ tốt

- Có thể bảo trì được
- Đáng tin cậy
- Có hiệu quả
- Khả dụng

1.3.8.A. Các công đoạn trong phát triển phần mềm

- **Giai đoạn đặc tả:** xác định các tính năng và điều kiện hoạt động của hệ thống. (thu thập yêu cầu và phân tích)
- **Giai đoạn phát triển:** Thiết kế phần mềm (software design), viết code (code generation)
- **Giai đoạn kiểm tra:** kiểm tra phần mềm (software testing), kiểm tra tính hợp lý của phần mềm.
- **Giai đoạn bảo trì:** Sửa lỗi (correction), thay đổi môi trường thực thi (adaptation), tăng cường (enhancement)

1.3.8.A. Các công đoạn trong phát triển phần mềm- Đặc tả

- Đây là bước hình thành bài toán hoặc đề tài.
- Ở bước này quản trị dự án hoặc phân tích viên hệ thống phải biết được:
 - vai trò của phần mềm cần phát triển trong hệ thống,
 - ước lượng công việc, lập lịch biểu và phân công công việc.
 - phải biết người đặt hàng muốn gì.
 - Các yêu cầu phải được thu thập đầy đủ và được phân tích theo chiều rộng và chiều sâu.
 - Công cụ sử dụng chủ yếu ở giai đoạn này là các lược đồ, sơ đồ phản ánh rõ các thành phần của hệ thống và mối liên quan giữa chúng với nhau.
- **Chú ý: Phần đặc tả chỉ quan tâm chủ yếu đến giá trị Vào, Ra chứ không quan tâm đến cấu trúc và nội dung các thao tác cần thực hiện.**

1.3.8.A. Các công đoạn trong phát triển phần mềm- Phát triển

- Dựa vào các nội dung đã xác định được, dùng ngôn ngữ đặc tả hình thức (dựa trên các kiến trúc toán học) hoặc phi hình thức (tựa ngôn ngữ tự nhiên) hoặc kết hợp cả hai để mô tả những yếu tố sau đây của chương trình:
 - Giá trị nhập, giá trị xuất.
 - Các phép biến đổi
 - Các yêu cầu cần đạt được ở mỗi điểm của chương trình.
- Sau bước thiết kế là bước triển khai các đặc tả chương trình thành một sản phẩm phần mềm dựa trên một ngôn ngữ lập trình cụ thể. Trong giai đoạn này các lập trình viên sẽ tiến hành cài đặt các thao tác cần thiết để thực hiện đúng các yêu cầu đã được đặc tả.

1.3.8.A. Các công đoạn trong phát triển phần mềm- Kiểm tra

- Sau giai đoạn phát triển, ta cần phải chứng minh tính đúng đắn của chương trình sau khi đã tiến hành cài đặt.
- thông thường, ở bước này ta coi các chương trình như những hộp đen. Vấn đề đặt ra là xây dựng một cách có chủ đích các tập dữ liệu nhập khác nhau để giao cho chương trình thực hiện rồi dựa vào kết quả thu được để đánh giá chương trình. Công việc như trên được gọi là kiểm thử chương trình. Công việc kiểm thử nhằm vào các mục tiêu sau:
 - Kiểm tra để phát hiện lỗi của chương trình.
 - Kiểm tra tính ổn định, hiệu quả cũng như khả năng tối đa của chương trình.
- Tùy theo mục đích mà người ta thiết kế các tập dữ liệu thử sao cho có thể phủ hết các trường hợp cần quan tâm.

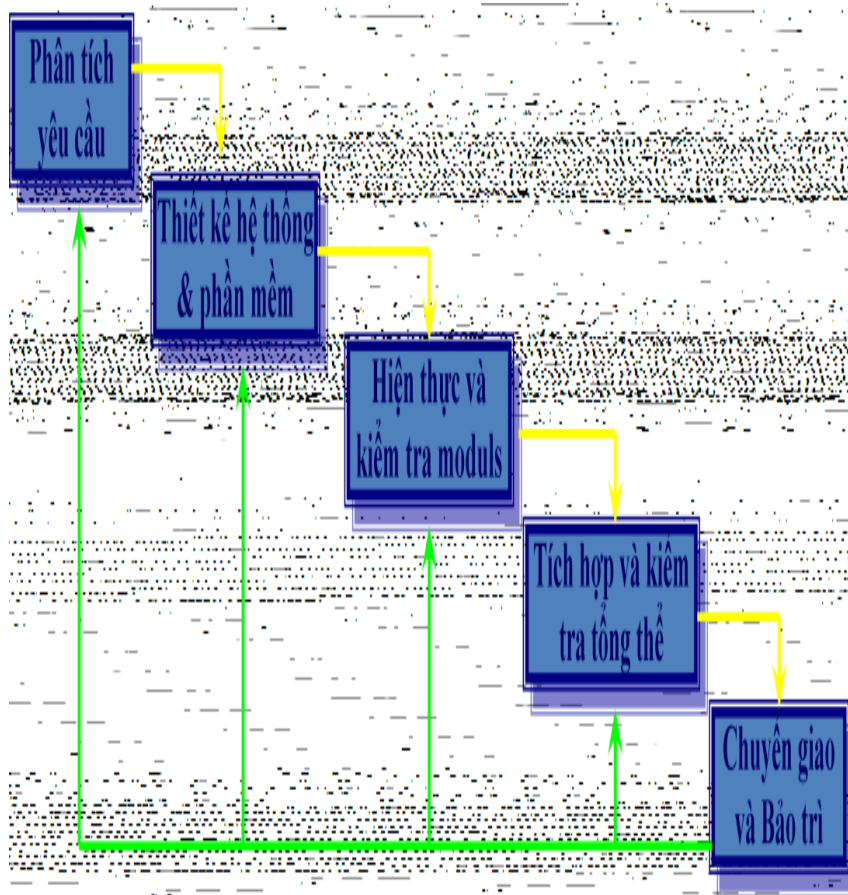
1.3.8.A. Các công đoạn trong phát triển phần mềm- Bảo trì

- quản lý việc triển khai và sử dụng phần mềm cũng là một vấn đề cần được quan tâm trong qui trình phát triển phần mềm.
- Trong quá trình xây dựng phần mềm, toàn bộ các kết quả phân tích, thiết kế, cài đặt và hồ sơ liên quan cần phải được lưu trữ và quản lý cẩn thận nhằm đảm bảo cho công việc được tiến hành một cách hiệu quả nhất và phục vụ cho công việc bảo trì phần mềm về sau.
- công việc quản lý không chỉ dừng lại trong quá trình xây dựng phần mềm mà trái lại còn phải được tiến hành liên tục trong suốt quá trình sống của nó

1.3.8. Một số loại mô hình tiến trình phần mềm tiêu biểu

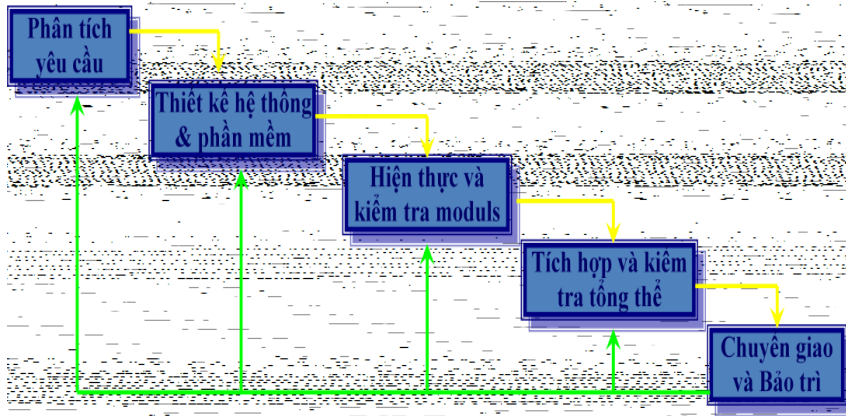
- Mô hình thác nước- waterfall
- Mô hình hình mẫu- Prototyping - Evolutionary Development
- Mô hình xoắn ốc- Boehm's Spiral Model
- Phát triển phần mềm theo hướng sử dụng lại RAD- Rapid Application Development

1.3.8.1. Mô hình thác nước/tuần tự – Sequency model



- Kỹ nghệ hệ thống: xây dựng yêu cầu
- Phân tích:
- Thiết kế: chuyển hóa các yêu cầu phần mềm thành các mô tả thiết kế
 - Thiết kế kiến trúc
 - Thiết kế dữ liệu
 - Thiết kế các thủ tục
 - Thiết kế giao diện
- Mã hóa
- Kiểm thử
- Bảo trì

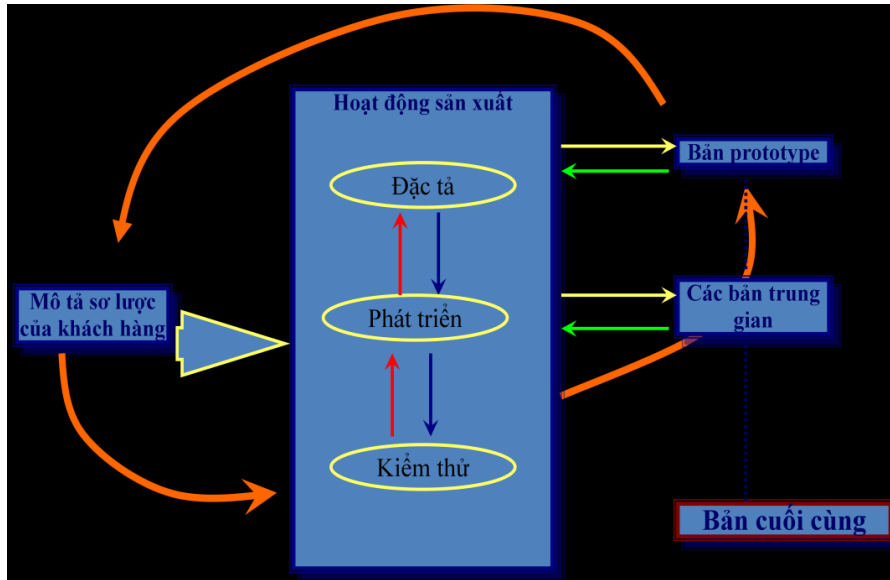
1.3.8.1. Mô hình thác nước/tuần tự – Sequency model



Hạn chế: khó thực hiện các thay đổi một khi đã thực hiện xong một giai đoạn nào đó

- Mô hình này chia quá trình phát triển phần mềm thành những giai đoạn tuần tự nối tiếp nhau.
- Mỗi giai đoạn sẽ có một mục đích nhất định.
- Kết quả của giai đoạn trước sẽ là thông tin đầu vào cho giai đoạn tiếp theo sau

1.3.8.2. Mô hình hình mẫu

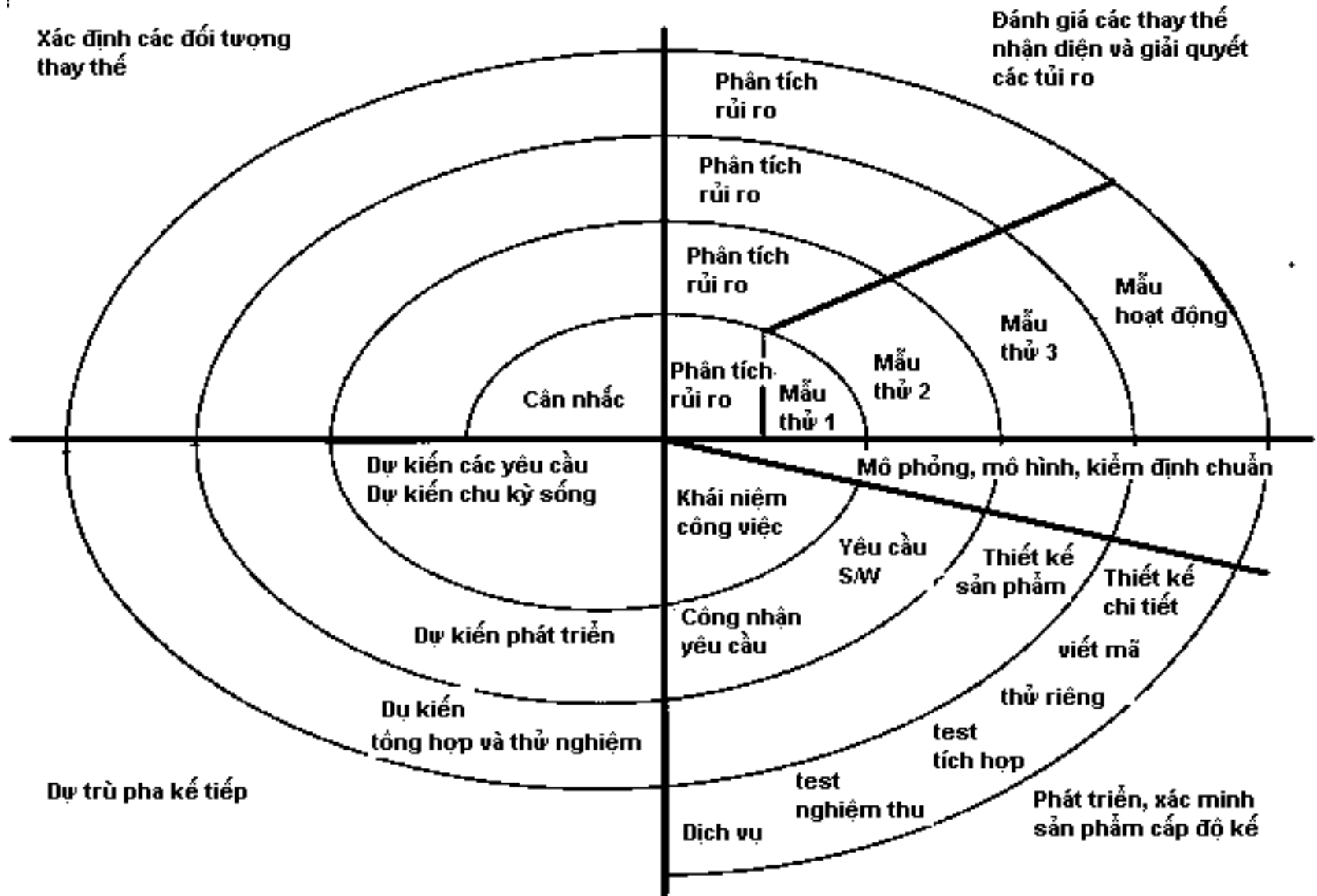


- Tương tự như mô hình thác nước với bổ sung vào các giai đoạn thực hiện phần mềm mẫu ngay khi xác định yêu cầu nhằm mục tiêu phát hiện nhanh các sai sót về yêu cầu
- Các giai đoạn có thể tiến hành lặp đi lặp lại

1.3.8.2. Mô hình hình mẫu

- Ưu: là một hướng tiếp cận tốt khi các yêu cầu chưa rõ ràng và khó đánh giá được tính hiệu quả của các thuật toán.
- Ứng dụng:
 - Dùng cho các hệ thống nhỏ.
 - Cần sự cấp bách về thời gian triển khai ngắn.
 - Trong trường hợp những hệ thống mà việc đặc tả các yêu cầu là rất khó và không rõ ràng ngay từ đầu.
- Hạn chế:
 - Các bản mẫu có thể bị “throw-away” gây lãng phí cho dự án.
 - Các tiến trình không được phân định rõ ràng
 - HT thường có cấu trúc lỏng lẻo
 - Cần có những kỹ năng đặc biệt trong quản lý và phát triển
 - Khách hàng hối thúc nhà phát triển hoàn thành sản phẩm một khi thấy được các bản mẫu (prototype) đầu tiên

1.3.8.4. Mô hình xoắn ốc

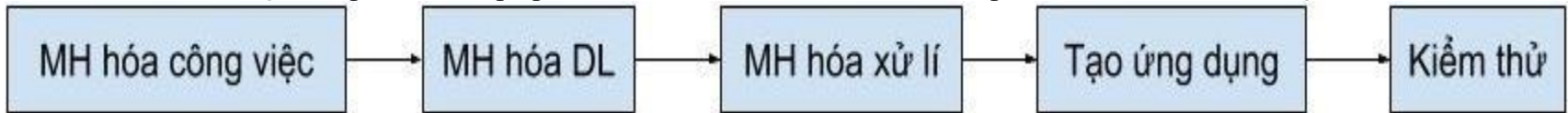


1.3.8.4. Mô hình xoắn ốc

- Mô hình này là sự kết hợp của mô hình bản mẫu thiết kế và mô hình tuần tự tuyến tính được lặp lại nhiều lần.
- Ở lần lặp tiếp theo hệ thống sẽ được tìm hiểu và xây dựng hoàn thiện hơn ở lần lặp trước đó.
- Ngoài ra ở cuối mỗi lần lặp sẽ có thêm công đoạn phân tích mức độ rủi ro để quyết định xem có nên đi tiếp theo hướng này nữa hay không.
- Mô hình này phù hợp với các hệ thống phần mềm lớn do có khả năng kiểm soát rủi ro ở từng bước tiến hóa.
- Tuy nhiên vẫn chưa được sử dụng rộng rãi như mô hình thác nước hoặc bản mẫu do đòi hỏi năng lực quản lý, năng lực phân tích rủi ro cao.

1.3.8.5. Mô hình RAD

(Rapid Application Developer Model,)



Business modeling:

- Thông tin nào điều khiển xử lý nghiệp vụ ?
- Thông tin gì được sinh ra?
- Ai sinh ra nó ?
- Thông tin đi đến đâu ?
- Ai xử lý chúng ?

Application Generation :

- Dùng các công cụ để xây dựng phần mềm

Data modeling:

- XD DL cần cho nghiệp vụ
- Định nghĩa các thuộc tính của từng đối tượng và xác lập quan hệ giữa các đối tượng

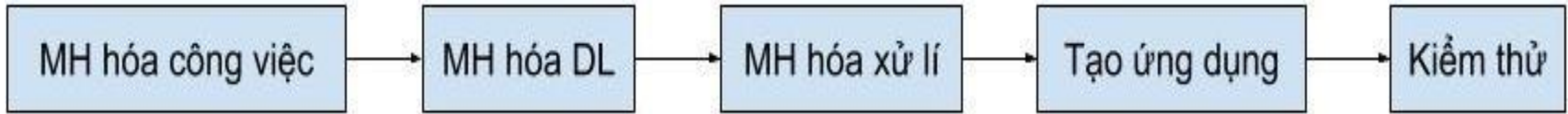
Process modeling :

- XD luồng thông tin thực hiện chức năng nghiệp vụ.
- Tạo mô tả xử lý để cập nhật (thêm, sửa, xóa, khôi phục) từng đối tượng dữ liệu.

Testing and Turnover: :

- Kiểm thử và chạy tải

1.3.8.5. Mô hình RAD



- Là quy trình phát triển phần mềm với mỗi chu trình phát triển rất ngắn (60-90 ngày)
- Xây dựng dựa trên hướng thành phần với khả năng tái sử dụng (reuse).
- Gồm một số nhóm (teams), mỗi nhóm làm 1 RAD theo các pha: Mô hình nghiệp vụ, Mô hình dữ liệu, Mô hình xử lý, Tạo ứng dụng, Kiểm thử và đánh giá

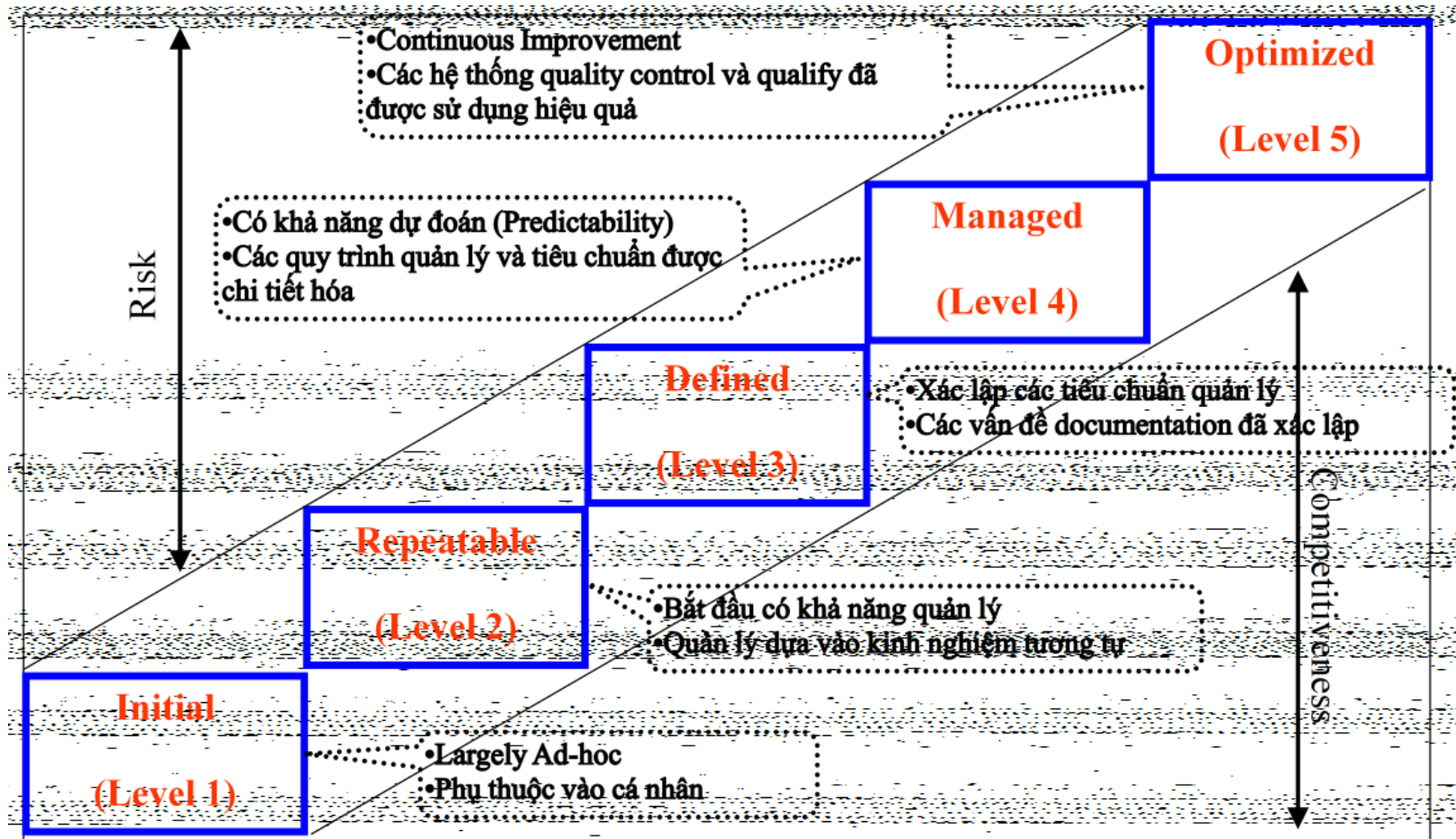
Các hạn chế của mô hình RAD:

- Cần nguồn nhân lực dồi dào để tạo các nhóm cho các chức năng chính
- Yêu cầu hai bên giao kèo trong thời gian ngắn phải có phần mềm hoàn chỉnh, thiếu trách nhiệm của một bên dễ làm dự án đổ vỡ
- - RAD không tốt cho ứng dụng không thể môđun hóa hoặc đòi hỏi tính năng cao
- - Mạo hiểm kỹ thuật cao thì không nên dùng RAD

1.4. Các tiêu chuẩn dùng trong ngành Công nghiệp phần mềm

- The Capability Maturity Model (CMM) của Software Engineering Institute (SEI) - Đại học Carnegie Mellon.
 - Chú trọng đến tính hệ thống và khả năng quản trị của các công ty phần mềm hơn là một quy trình (process) cụ thể.
- The process Improvement Paradigm (PIP) của Software Engineering Laboratory (SEL) – NASA's Goddard Space Flight Center
 - Tương tự như CMM, chú trọng đến tính hệ thống và những hướng dẫn để tăng cường tính năng của các quá trình quản lý
- Các chuẩn của Department of Defense
- International Standards Organisation - ISO 9001
- The European ESPRIT project

CMM



CHƯƠNG 2. XÁC ĐỊNH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM

- 2.1. Tổng quan về yêu cầu phần mềm
 - 2.1.1. [khái niệm yêu cầu phần mềm](#)
 - 2.1.2. [phân loại các yêu cầu phần mềm](#)
 - 2.1.2.1. yêu cầu chức năng
 - 2.1.2.2. yêu cầu phi chức năng
 - 2.1.2.3. yêu cầu miền lĩnh vực
 - 2.1.3. tiến trình kỹ nghệ yêu cầu
 - 2.1.3.1. nghiên cứu khả thi
 - 2.1.3.2. phát hiện và phân tích yêu cầu
 - 2.1.3.3. đặc tả yêu cầu
 - 2.1.3.4. thẩm định yêu cầu
 - 2.1.4. tài liệu yêu cầu phần mềm (TLYCPM)
 - 2.1.4.1. nội dung và yêu cầu đối với TLYCPM
 - 2.1.4.2. cấu trúc TLYCPM

CHƯƠNG 2. XÁC ĐỊNH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM

- 2.2. xử lý yêu cầu phần mềm
 - 2.2.1. khảo sát hệ thống và phân tích khả thi
 - 2.2.2. phát hiện và phân tích yêu cầu
 - 2.2.2.1. những khó khăn trong việc nhận ra yêu cầu
 - 2.2.2.2. tiến trình phát hiện và phân tích yêu cầu
 - 2.2.3. các kỹ thuật phân tích yêu cầu
 - 2.2.3.1. tiếp cận yêu cầu định hướng cách nhìn
 - 2.2.3.2. kỹ thuật xử lý yêu cầu định hướng cách nhìn
 - 2.2.3.3. kỹ thuật phân tích yêu cầu dựa trên mô hình
 - 2.2.3.4. kỹ thuật phân tích hình thức hóa
 - 2.2.4. tiến hóa và quản lý yêu cầu
 - 2.2.4.1. sự cần thiết tiến hóa yêu cầu
 - 2.2.4.2. quản lý sự thay đổi yêu cầu

CHƯƠNG 2. XÁC ĐỊNH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM

- 2.3. đặc tả yêu cầu (ĐTYC)
 - 2.3.1. khái niệm về ĐTYC
 - 2.3.2. các ngôn ngữ ĐTYC
 - 2.3.2.1. ngôn ngữ tự nhiên có cấu trúc
 - 2.3.2.2. ngôn ngữ mô tả chương trình (Program Description Language PDL)
 - 2.3.3. ĐTYC dựa trên mô hình
 - 2.3.3.1. đặc tả dữ liệu hướng cấu trúc
 - 2.3.3.2. đặc tả xử lý hướng cấu trúc- mô hình luồng dữ liệu
 - 2.3.3.3. đặc tả hành vi hướng đối tượng (HĐT)- biểu đồ phân tích tương tác
 - 2.3.4. đặc tả yêu cầu hình thức hóa
 - 2.3.4.1. các PP đặc tả hình thức hóa
 - 2.3.4.2. đặc tả đại số
 - 2.3.4.3. máy trạng thái hữu hạn
 - 2.3.5. so sánh các loại hình đặc tả

CHƯƠNG 2. XÁC ĐỊNH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM

- 2.4. thẩm định yêu cầu (TĐYC)
 - 2.4.1. khái niệm về TĐYC
 - 2.4.2. các kĩ thuật TĐYC
 - 2.4.2.1. rà soát yêu cầu
 - 2.4.2.2. làm bản mẫu
 - 2.4.2.3. tạo sinh các ca kiểm thử
 - 2.4.2.4. phân tích tính nhất quán được tự động

CHƯƠNG 2. XÁC ĐỊNH VÀ ĐẶC TẢ YÊU CẦU PHẦN MỀM

- 2.5. đặc tả hệ thống và làm bản mẫu
 - 2.5.1. tiến trình làm bản mẫu phần mềm
 - 2.5.2. tạo bản mẫu trong tiến trình phần mềm
 - 2.5.3. các kĩ thuật tạo bản mẫu
 - 2.5.3.1. ngôn ngữ đặc tả thi hành được
 - 2.5.3.2. các ngôn ngữ bậc rất cao
 - 2.5.3.3. các ngôn ngữ thể hệ thứ 4
 - 2.5.3.4. lắp ghép từ các thành phần dùng lại được

2.1.1. khái niệm yêu cầu phần mềm

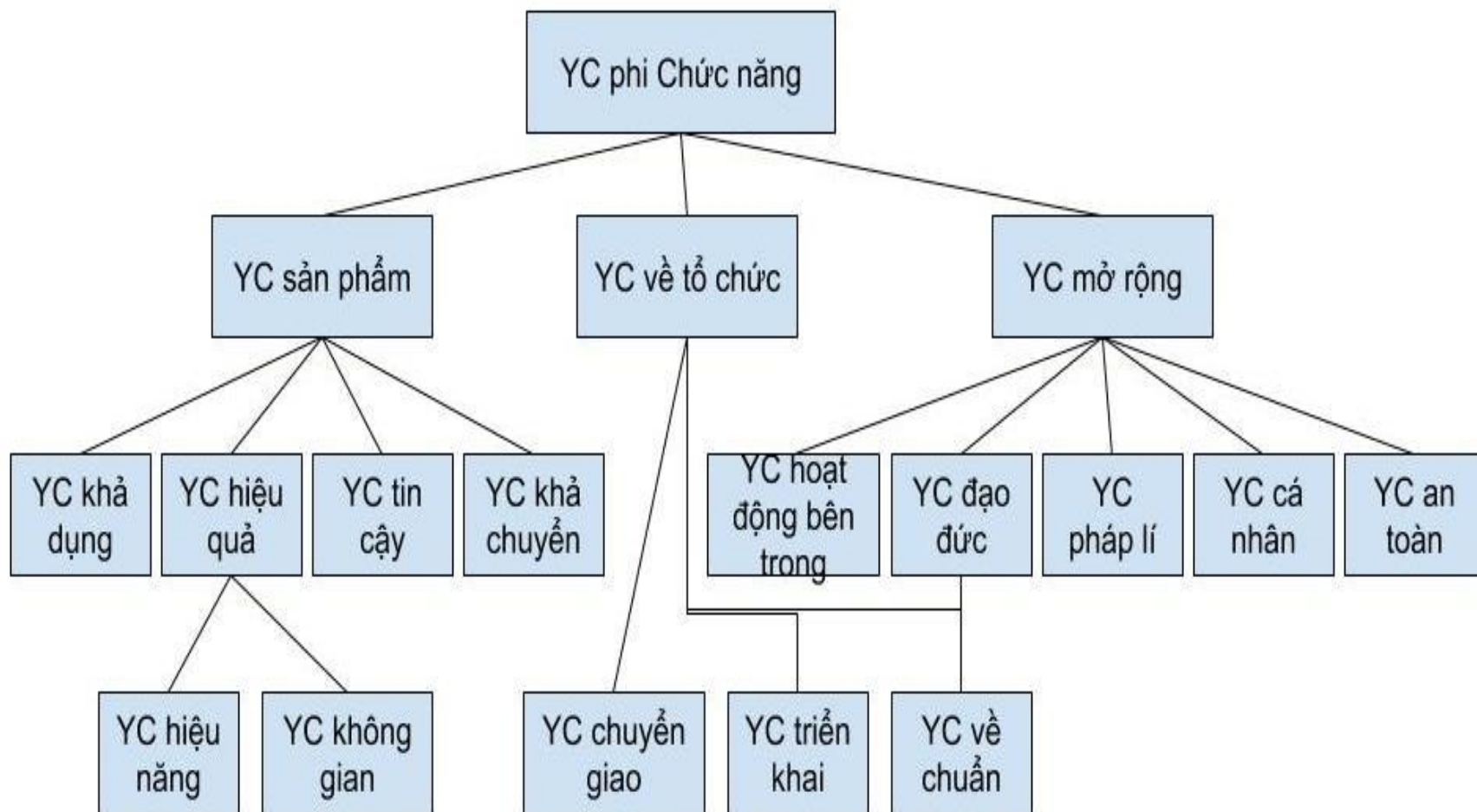
- Phân biệt “nhu cầu” <> “yêu cầu”
- Yêu cầu cho 1 hệ thống phần mềm mô tả những việc mà hệ thống làm (**yêu cầu chức năng**, dịch vụ) và những ràng buộc mà nó phải tuân thủ khi thực hiện (**yêu cầu phi chức năng**).
- **3 mức yêu cầu:**
 - **Yêu cầu người sử dụng** (hướng NSD): là những phát biểu bằng ngôn ngữ tự nhiên cùng các biểu đồ để mô tả các dịch vụ mà hệ thống cung cấp và các ràng buộc khi nó hoạt động
 - **Yêu cầu hệ thống :**
 - nêu ra các dịch vụ của hệ thống và chi tiết các ràng buộc của nó.
 - Mức trung gian giữa NSD+kỹ sư phần mềm
 - **Đặc tả phần mềm:**
 - là sự mô tả khái quát các chức năng phần mềm hỗ trợ hoạt động nghiệp vụ.
 - Là cơ sở để thiết kế và triển khai phần mềm.
 - Đặc tả phần mềm Bổ sung thêm chi tiết sẽ thành đặc tả yêu cầu hệ thống
 - Hướng kỹ sư phần mềm



2.1.2. phân loại các yêu cầu phần mềm

- 2.1.2.1. yêu cầu chức năng (Functional Requirement)
 - Là những phát biểu về chức năng/dịch vụ mà hệ thống cung cấp.
 - Yêu cầu chức năng có thể gồm cả những chức năng mà hệ thống không nên thực hiện
- 2.1.2.2. yêu cầu phi chức năng (non- Functional Requirement)
 - Là những ràng buộc lên các chức năng/dịch vụ mà hệ thống cung cấp
 - Gồm các ràng buộc về tg, ngân sách, bản quyền...

Các dạng yêu cầu phi chức năng



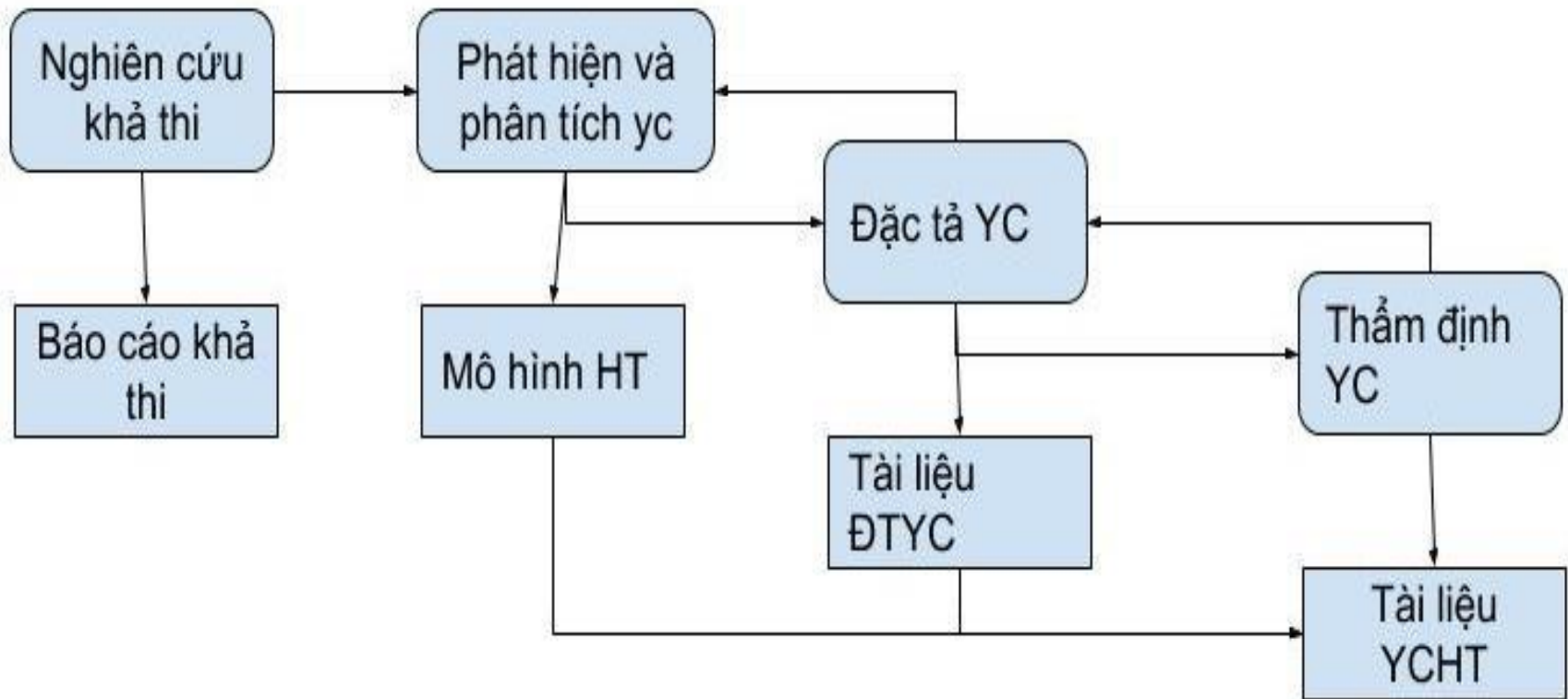
2.1.2. phân loại các yêu cầu phần mềm

- 2.1.2.3. yêu cầu miền lĩnh vực (Domain Requirement)
 - Là các yêu cầu xuất phát từ miền ứng dụng của hệ thống.
 - Thường là các quy tắc nghiệp vụ liên quan đến các vấn đề như đạo đức, bí mật, an toàn



2.1.3. Tiến trình kỹ nghệ yêu cầu

TIẾN TRÌNH KỸ NGHỆ YÊU CẦU



2.1.4. Tài liệu yêu cầu phần mềm

- Nội dung và yêu cầu đối với TLYCPM
 - Là tài liệu chính thức cho người phát triển phần mềm
 - Chỉ mô tả các hoạt động của hệ thống từ bên ngoài
 - Chỉ ra được các ràng buộc của hệ thống trong quá trình vận hành
 - Dễ thay đổi
 - Phục vụ như TL tham khảo cho người bảo trì hệ thống
 - Dự đoán trước được vòng đời của hệ thống
 - Mô tả được các đáp ứng đối với những sự cố, thay đổi ngoài dự tính
 - Các yc phải **đầy đủ+ nhất quán**

2.1.4. Tài liệu yêu cầu phần mềm

- Cấu trúc TLYCPM

1. Giới thiệu

1. Mục tiêu của TLYC
2. Phạm vi của sản phẩm
3. Các định nghĩa, viết tắt
4. TL tham khảo
5. Tổng quan về TL

2. Mô tả chung

1. Các đặc điểm của sản phẩm
2. Các chức năng của sản phẩm
3. Các đặc trưng của NSD
4. Các ràng buộc chung
5. Các giả định và sự phụ thuộc

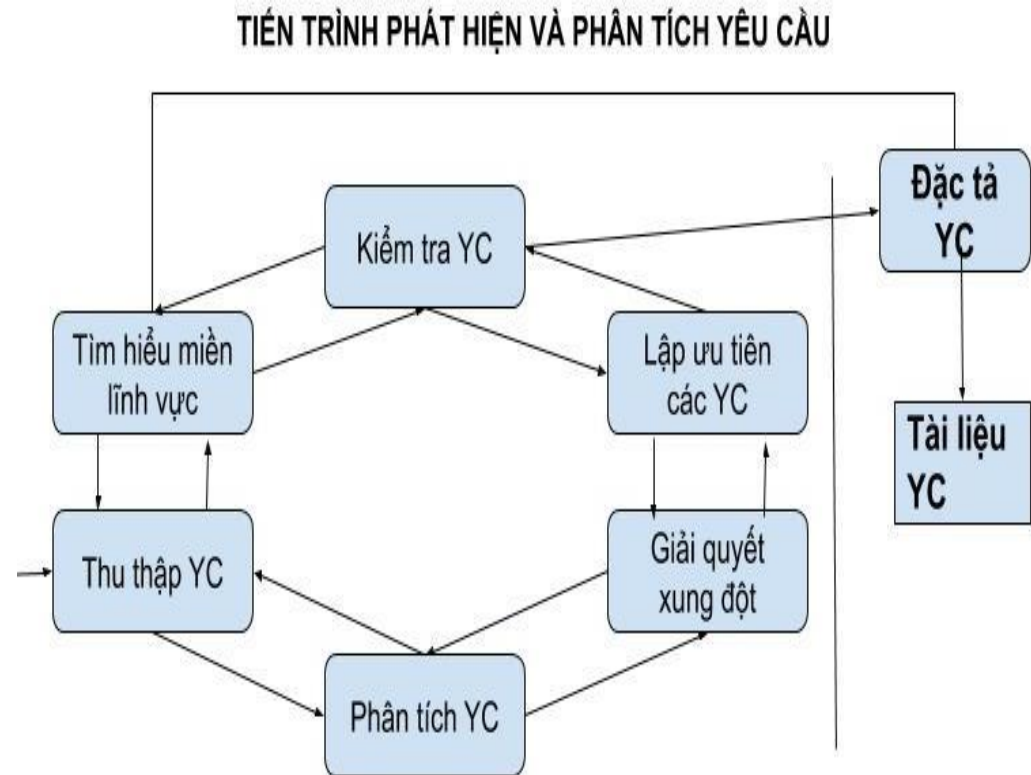
3. Các yêu cầu cụ thể

4. Phụ lục

5. Các chỉ dẫn

2.2. xác định yêu cầu phần mềm

- Khảo sát hệ thống và phân tích khả thi
 - Khả thi về kinh tế
 - Khả thi về kĩ thuật
 - Khả thi về pháp lí
 - Khả thi về hoạt động
 - Khả thi về thời gian
- Phát hiện và phân tích yêu cầu
 - Khó khăn
 - Tiến trình phát hiện và phân tích yêu cầu



2.2. xác định yêu cầu phần mềm

Các kĩ thuật phân tích yêu cầu

- Tiếp cận yêu cầu định hướng cách nhìn
 - Từ nguồn hay đích của dữ liệu
 - Từ khung làm việc
 - Từ sự tiếp nhận dịch vụ
- Kĩ thuật xác định yêu cầu định hướng cách nhìn:
 - Xđ khung nhìn
 - Cấu trúc khung nhìn
 - Tài liệu khung nhìn
 - ánh xạ HT<-> Khung nhìn

- Kĩ thuật phân tích yêu cầu dựa trên mô hình
 - MH phân tích YC-MH nghiệp vụ:
 - 2 cách tiếp cận:
 - Tiếp cận định hướng chức năng : dùng luồng dữ liệu
 - Tiếp cận định hướng đối tượng
 - Mô hình ngữ cảnh
 - MH cấu trúc chức năng
 - Mô tả chi tiết các chức năng
 - các đối tượng dữ liệu cần lưu trữ
 - Các mối liên kết DL với CN
 - Từ điển giải thích
 - Các thể hiện của MH phân tích

2.2. xác định yêu cầu phần mềm

- Kỹ thuật phân tích hình thức hóa

Tiến hóa và quản lý yêu cầu

- Sự cần thiết tiến hóa YC
 - Thực tiến khách quan
 - Phân loại YC:
 - YC lâu dài
 - YC dễ thay đổi
- Quản lý sự thay đổi YC
 - Định danh các YC
 - Định ra tiến trình quản lý thay đổi
 - Phân tích vấn đề đặt ra và đặc tả sự thay đổi
 - Phân tích thay đổi và dự toán chi phí
 - Triển khai sự thay đổi
 - Có chính sách lần vết

2.3. đặc tả yêu cầu

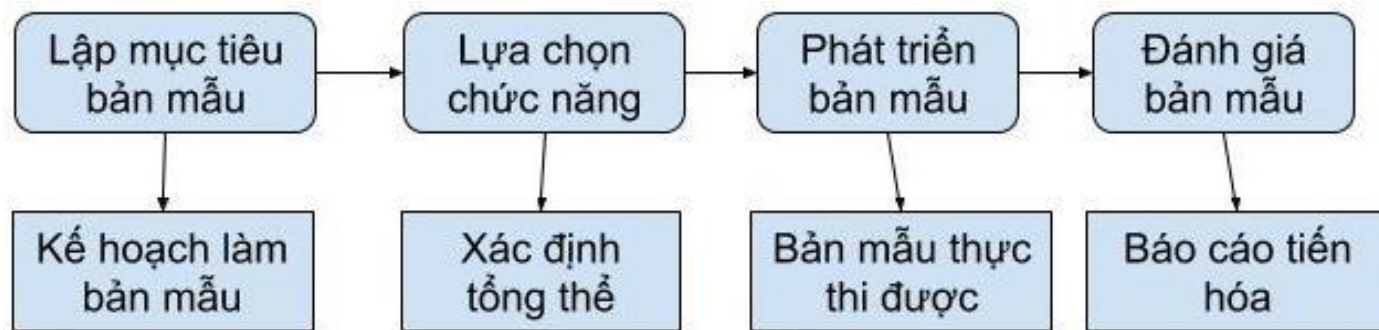
- Khái niệm về đặc tả yêu cầu
- Các ngôn ngữ đặc tả yêu cầu
 - Ngôn ngữ tự nhiên có cấu trúc
 - Ngôn ngữ mô tả chương trình – PLD
 - Ngôn ngữ đặc tả yêu cầu chuyên dụng
- Đặc tả yêu cầu dựa trên mô hình
 - Đặc tả dữ liệu hướng cấu trúc- MH thực thể QH
 - Đặc tả xử lý hướng cấu trúc- MH luồng dữ liệu
 - Đặc tả hành vi đối tượng- biểu đồ phân tích tương tác
- Đặc tả yêu cầu hình thức hóa (xem)
 - Đặc tả đại số
 - Máy trạng thái hữu hạn

2.4. thẩm định yêu cầu

- Khái niệm về thẩm định yêu cầu
- Các kĩ thuật thẩm định yêu cầu
 - Rà soát yêu cầu
 - Làm bản mẫu
 - Tạo sinh các ca kiểm thử
 - Phân tích tính nhất quán được tự động

2.5. đặc tả hệ thống và làm bản mẫu

- Tiến trình làm bản mẫu phần mềm



- Tạo bản mẫu trong tiến trình phần mềm:
 - Bản mẫu là mở rộng quá trình phân tích YC
 - Là một phần của hệ thống
 - Là chính hệ thống

2.5. đặc tả hệ thống và làm bản mẫu

- Các kĩ thuật tạo bản mẫu:
 - Sử dụng ngôn ngữ đặc tả thi hành đc
 - Sử dụng ngôn ngữ bậc cao
 - Các ngôn ngữ thế hệ thứ 4
 - Lắp ghép từ các thành phần dùng lại được

Chương 3. Thiết kế phần mềm

- 3.1. Tổng quan về thiết kế phần mềm
 - 3.1.1. khái niệm và vai trò của thiết kế
 - 3.1.2. triển khai thiết kế
 - 3.1.3. các chiến lược và pp thiết kế
 - 3.1.4. chất lượng thiết kế và các giải pháp đảm bảo chất lượng
- 3.2. thiết kế kiến trúc
 - 3.2.1. kiến trúc phần mềm và vai trò
 - 3.2.2. các mô hình kiến trúc
 - 3.2.3. tiến trình thiết kế kiến trúc
- 3.3. thiết kế hệ thống hướng chức năng
 - 3.3.1. thiết kế dữ liệu
 - 3.3.2. thiết kế xử lý
 - 3.3.3. các ưu, nhược của thiết kế hướng cấu trúc
- 3.5. thiết kế hệ thống thời gian thực
 - 3.5.1. khái niệm về hệ thống thời gian thực
 - 3.5.2. tiến trình thiết kế hệ thống thời gian thực
 - 3.5.3. MH hóa máy trạng thái
 - 3.5.4. bộ điều phối thời gian thực
 - 3.5.5. các hệ giám sát và điều khiển
 - 3.5.6. các hệ thu, nhận dữ liệu
- 3.6. thiết kế giao diện người-máy
 - 3.6.1. vai trò và tiến hóa giao diện N-M
 - 3.6.2. các nguyên tắc thiết kế giao diện N-M
 - 3.6.3. tiến trình thiết kế giao diện
 - 3.6.4. các hướng dẫn thiết kế giao diện
 - 3.6.5. chuẩn giao diện
- 3.7. tài liệu thiết kế

3.1.1. khái niệm và vai trò của thiết kế

- Khái niệm về thiết kế
 - *Thiết kế phần mềm là quá trình chuyển các đặc tả yêu cầu thành một biểu diễn thiết kế của hệ thống phần mềm cần xây dựng, sao cho người lập trình có thể ánh xạ nó thành chương trình vận hành được.*
- Các hoạt động chính:
 - Nghiên cứu để hiểu vấn đề
 - Chọn 1 số pp thiết kế và xđ các đặc điểm thô của nó
 - Mô tả trừu tượng cho mỗi giải pháp thiết kế
- Vai trò của thiết kế
 - là cơ sở cho việc triển khai xây dựng phần mềm
 - Tài liệu thiết kế là công cụ giao tiếp giữa các nhóm cùng xây dựng phần mềm, quản lý rủi ro, nhằm đạt hiệu quả
 - Để bảo trì hệ thống

3.1.1. khái niệm và vai trò của thiết kế

- Các khái niệm trong thiết kế

Trừu tượng:

- **mức cao nhất:** mô tả đại thể, sử dụng ngôn ngữ nghiệp vụ, ngôn ngữ người dùng.
- **mức vừa:** mô tả hướng thủ tục nhiều.
- **mức thấp:** dùng thuật ngữ chi tiết để có thể chuyển trực tiếp thành chương trình

Các dạng trừu tượng

- **TrT thủ tục:** là 1 dãy các lệnh có tên, có chức năng xđ và giới hạn. TD: “đi vào” cửa={bước tới cửa, nắm quả đấm cửa, xoay quả đấm, kéo cửa...}
- **TrT dữ liệu:** là 1 tập các dữ liệu có tên mô tả cho 1 đối tượng dữ liệu. TD: “Sec thanh toán”={số sec, tên người thanh toán, số tiền,...}
- **TrT điều khiển:** mô tả tương tác giữa các đối tượng của hệ thống mà không xđ các chi tiết bên trong chúng. TD: “đồng bộ hóa dữ liệu”

Phân rã: phân chia 1 ĐT thành nhiều ĐT nhỏ hơn

Làm mịn: <>phân rã ở chỗ nó là hoạt động mang tính chủ quan của người thiết kế.

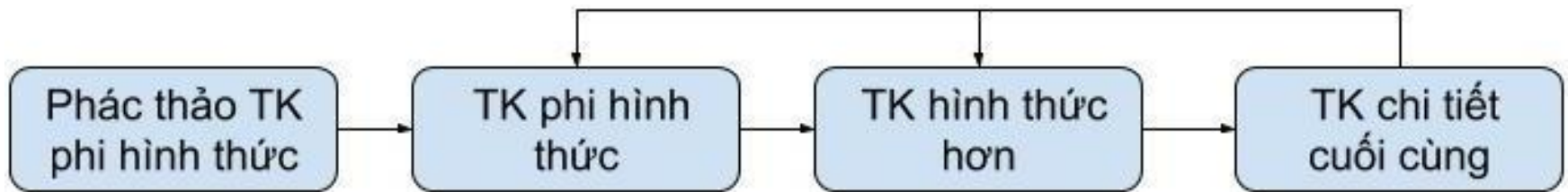
Module: phần mềm được chia thành các phần riêng biệt có tên và địa chỉ xđ

Thủ tục phần mềm

Che dấu thông tin

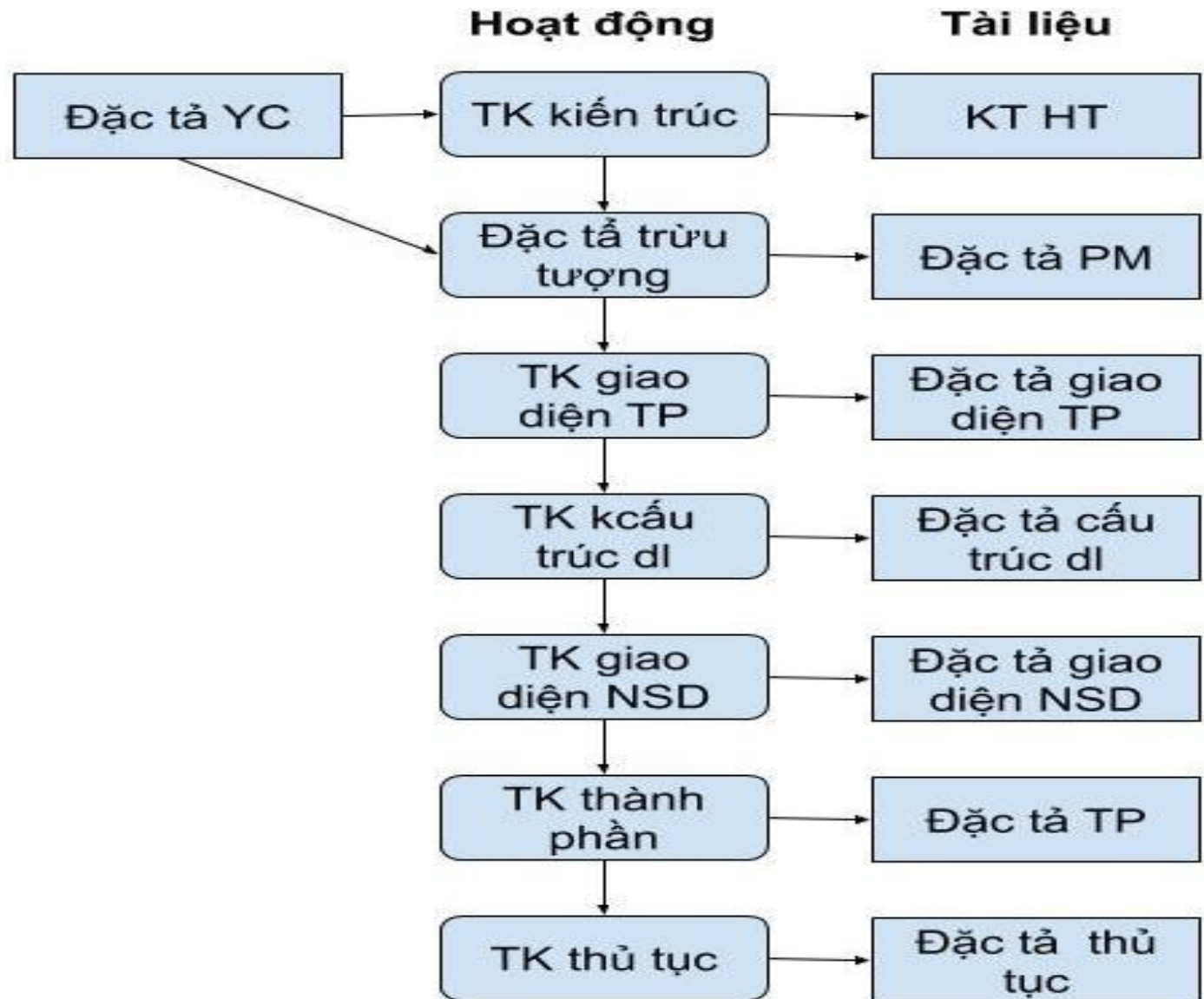
3.1.2. triển khai thiết kế

- Tiến trình thiết kế



3.1.2. triển khai thiết kế

Các
hoạt
động
và
sản
phẩm
thiết
kế



3.1.2.2. Các hoạt động và sản phẩm thiết kế

- **Thiết kế kiến trúc:** xđ các hệ con tạo nên hệ thống tổng thể và các mối quan hệ giữa chúng
- **Đặc tả trừu tượng:** đối với mỗi hệ con, cần mô tả trừu tượng các dịch vụ mà nó cung cấp cùng các ràng buộc mà nó phải tuân thủ
- **Thiết kế các giao diện thành phần:**
- **Thiết kế cấu trúc dữ liệu:** thiết kế cấu trúc dữ liệu lưu trữ và đặc tả các cấu trúc dữ liệu được dùng khi hệ thống thực hiện
- **Thiết kế hệ thống giao diện người sử dụng**
- **Thiết kế các thành phần**
- **Thiết kế thủ tục:** thiết kế đặc tả các thuật toán, quy trình dùng để thực hiện các dịch vụ của mỗi TP sao cho có thể ánh xạ trực tiếp nó vào 1 ngôn ngữ lập trình.

3.1.2.3. biểu diễn thiết kế

1 bản thiết kế phần mềm là 1 MH mô tả đối tượng của TG thực với nhiều TP và các mối quan hệ giữa chúng với nhau.

3 hình thức thường sử dụng để mô tả thiết kế :

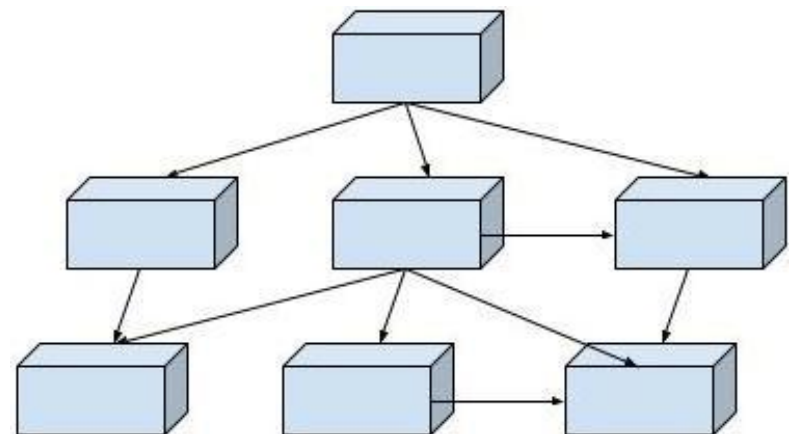
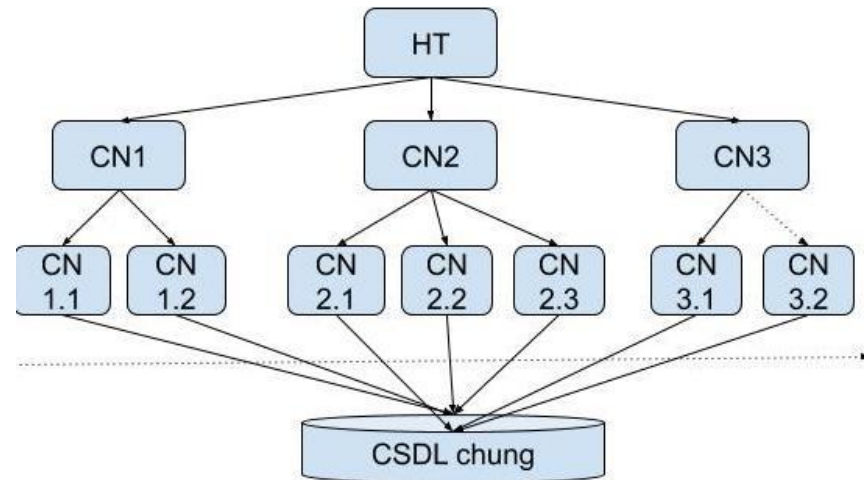
- Các biểu đồ
- Ngôn ngữ mô tả chương trình
- Dùng văn bản không hình thức hóa: thường để mô tả các thông tin không thể hình thức hóa được như các thông tin phi chức năng

3.1.2.4. các giai đoạn thiết kế

- **Thiết kế logic:** mô tả các TP của hệ thống và các mối quan hệ giữa chúng mà không gắn với bất kì 1 phương tiện vật lí nào → có thể vẽ lên 1 hệ thống phần mềm lí tưởng đáp ứng các yêu cầu đặt ra.
- **Thiết kế vật lí:** chọn các giải pháp công nghệ hiện hữu để thực hiện các cấu trúc logic cho phù hợp với điều kiện môi trường cụ thể

3.1.3. các chiến lược và PP thiết kế

- **CL và PP hướng cấu trúc:** hệ thống được phân chia thành các chức năng. Trạng thái của hệ thống thể hiện qua CSDL dùng chung và được chia sẻ cho các chức năng thao tác trên nó.
- **CL và PP hướng đối tượng:** hệ thống được nhìn nhận như 1 bộ các ĐT tương tác nhau. Mỗi ĐT bao hàm trong nó cả các dữ liệu và PP thực hiện trên chúng.
- **CL thiết kế hệ thống tương tranh:** thiết kế các quá trình hoạt động song song và có liên quan chặt chẽ với nhau.



3.1.3. các chiến lược và PP thiết kế

Cách tiếp cận chung của các PP thiết kế

- **Cách nhìn cấu trúc:** nhìn tổng thể thông qua lược đồ cấu trúc hệ thống: các TP của hệ thống và mối quan hệ giữa chúng.
- **Cách nhìn quan hệ thực thể:** đặc tả dữ liệu quan hệ
- **Cách nhìn luồng dữ liệu:** lược đồ luồng dữ liệu
- **Cách nhìn vận động:** lược đồ chuyển trạng thái để nhìn 1 hệ thống diễn ra theo tg thực

3.2. thiết kế kiến trúc

- Một hệ thống lớn luôn có thể phân rã thành các hệ thống con độc lập tương đối với nhau.
- **Thiết kế kiến trúc:** là Tiến trình XD các hệ thống con và thiết lập 1 khung làm việc cho việc điều khiển và giao tiếp giữa chúng.
- TKKT gồm các việc sau:
 - **Cấu trúc hệ thống:** cấu trúc 1 hệ thống thành 1 số hệ thống con
 - **MH hóa điều khiển:** thiết lập 1 MH chung mô tả mối quan hệ điều khiển giữa các TP trong hệ thống
 - **Phân rã Module:** phân rã 1 hệ thống con thành các module và liên kết giữa chúng.

3.3. thiết kế hệ thống hướng chức năng

- Thiết kế dữ liệu
 - Thiết kế MH dữ liệu logic
 - Thiết kế cơ sở dữ liệu vật lí
- Thiết kế xử lý
 - Thiết kế xử lý logic
 - Thiết kế kiến trúc module

3.4. thiết kế hệ thống hướng đối tượng

- Xđ kiến trúc hệ thống
- Sắp thứ tự ưu tiên các gói
- Với mỗi gói, thiết kế cho mỗi ca sử dụng thuộc gói vằng cách xđ các lớp thiết kế tham gia triển khai các lớp phân tích
- Xây dựng biểu đồ tương tác giữa các lớp
- Thiết kế chi tiết các lớp
- Phân tích và hoàn thiện biểu đồ lớp dựa trên các mẫu thiết kế

3.5. thiết kế hệ thống thời gian thực

- Xử các tác nhân kích thích mà hệ thống phải xử lý và đáp ứng
- Với mỗi kích thích và đáp ứng, xử ràng buộc TG cho cả việc tiếp nhận kích thích và việc xử lý đáp ứng
- Phân tích các kích thích và quá trình xử lý đáp ứng thành 1 số các tiến trình song song
- Với mỗi cặp Kích thích/đáp ứng, thiết kế các thuật toán cho các tính toán yêu cầu
- Thiết kế 1 hệ thống lập lịch đảm bảo các quá trình được bắt đầu đúng thời điểm
- Tích hợp hệ thống dưới sự điều khiển của 1 bộ điều phối TG thực

Chú ý

- *hệ thống phần mềm lớn thường phức tạp, người ta thường dùng **các phương pháp tiếp cận khác nhau** trong việc thiết kế các thành phần khác nhau của hệ thống.*
- **ở mức tổng thể** thì hệ thống như là một bộ các đối tượng (chứ không phải là mộ bộ các chức năng), cho nên **cách tiếp cận hướng đối tượng** là thích hợp hơn.
- **mức chi tiết** thì nên xem chúng như là **các chức năng** tương tác giữa các đối tượng.

3.6. thiết kế giao diện Người-Máy

- Thiết kế giao diện NSD phải lấy con người làm trung tâm
- Đảm bảo các nguyên tắc:
 - Tính quen thuộc với NSD
 - Tính nhất quán
 - Ít ngạc nhiên
 - Có khả năng phục hồi
 - Hướng dẫn rõ ràng
 - Đa dạng NSD

3.7. tài liệu thiết kế

Thường gồm những tài liệu sau:

- I. Phạm vi
 - A. Mục tiêu hệ thống
 - B. Phần cứng+mềm và giao diện N-M
 - C. Các chức năng chính
 - D. CSDL
 - E. Các ràng buộc, giới hạn thiết kế chính
- II. TLTK
 - A. TL phần mềm hiện có
 - B. TL hệ thống
 - C. TL người cung cấp (phần cứng, phần mềm)
 - D. tham khảo kĩ thuật
- III. Mô tả thiết kế
 - A. Mô tả dữ liệu
 - 1. tổng quan về luồng dữ liệu
 - 2. tổng quan về cấu trúc dữ liệu
 - B. cấu trúc chương trình suy dẫn
 - C. giao diện bên trong cấu trúc
- IV. Module (cho mỗi Module)
 - A. Lời thuật xử lý
 - B. mô tả giao diện
 - C. mô tả ngôn ngữ thiết kế/ mô tả khác
 - D. các Module đã dùng
 - E. tổ chức dữ liệu
 - F. Bình luận

3.7. tài liệu thiết kế (tiếp)

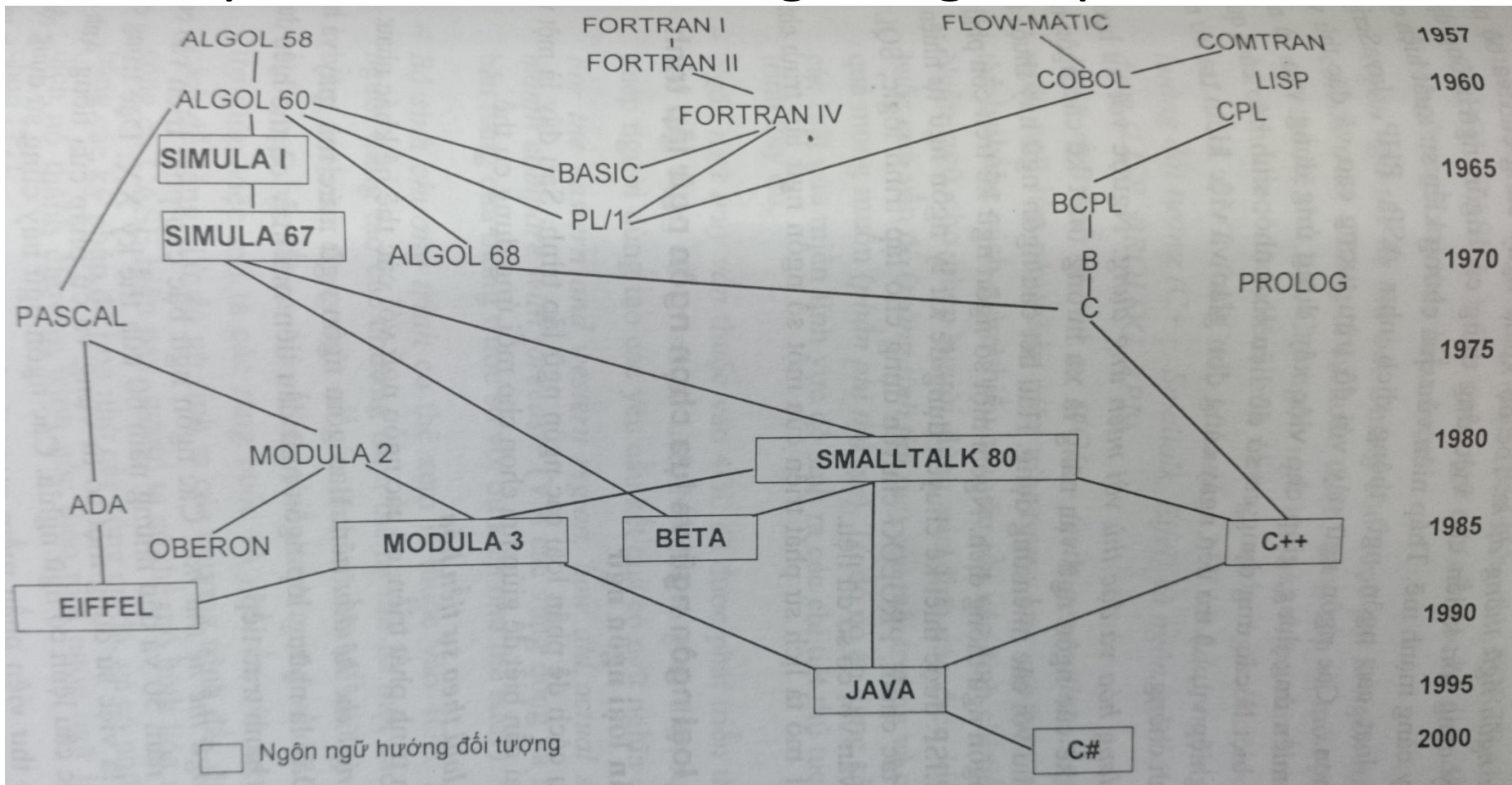
- V. cấu trúc tệp và dữ liệu toàn cục
 - A. cấu trúc tệp dữ liệu
 - 1. cấu trúc logic
 - 2. mô tả bản ghi logic
 - 3. PP truy nhập
 - B. dữ liệu toàn cục
 - C. tham khảo chéo tệp và dữ liệu
- VI. Tham khảo chéo yêu cầu
- VII. Điều khoản kiểm thử
 - A. hướng dẫn kiểm thử
 - B. chiến lược kiểm thử
 - C. Xem xét đặc biệt
- VIII. Bao gói
 - A. các điều khoản đặc biệt
 - B. xem xét chuyển giao
- IX. Lưu ý đặc biệt
- X. phụ lục

Tổng kết chương thiết kế

- Thiết kế phần mềm là quá trình chuyển hóa các đặc tả yêu cầu thành biểu diễn thiết kế. Thiết kế là tạo mô hình 1 hệ thống của thế giới thực
- 3 cách thiết kế:
 - **hướng cấu trúc:** thiết kế DL, thiết kế xử lý: lược đồ quan hệ, csdlqh, sơ đồ luồng dữ liệu
 - **hướng ĐT:** xđ các lớp, thuộc tính và mối quan hệ của chúng. Sử dụng UML
 - **hệ thống TG thực:** hệ thống đáp ứng với giới hạn TG chặt chẽ. Sử dụng PP máy trạng thái hữu hạn
- Tiến trình thiết kế: thiết kế kiến trúc → thiết kế dữ liệu, thủ tục → thiết kế giao diện (/ từ thiết kế sơ bộ → thiết kế chi tiết)

Chương 4. Lập trình

- Sự phát triển của các ngôn ngữ lập trình



4.1. lựa chọn ngôn ngữ

- **C:** có khả năng can thiệp mức thấp của hệ thống
- **Pascal/ Delphi:** có thư viện đồ họa mạnh; vẫn được sử dụng rộng rãi trong phát triển ứng dụng GIS
- **FORTRAN:** khả năng tính toán có độ chính xác cao, thư viện toán học phong phú; hỗ trợ lập trình song song, thường dùng trong tính toán khoa học
- **COBOL:** là ngôn ngữ cho ứng dụng kinh doanh, khai thác CSDL lớn. **FOXPRO** thường dùng cho phát triển các ứng dụng CSDL nhỏ
- **C++, SMALLTALK, JAVA:** lập trình hướng đối tượng
- **LIPS, PROLOG:** dùng cho các ứng dụng trí tuệ nhận tạo
- Các ngôn ngữ thông dịch: **ASP, PHP, JavaScript, PERL**

4.2. phương pháp lập trình

- **Trừu tượng hóa**
- **Lập trình thủ tục**
 - Sử dụng các thủ tục
 - Lập trình có cấu trúc
- **Lập trình hướng đối tượng (HĐT)**
 - Bao gói/ che dấu thông tin
 - Kế thừa
 - Đa hình
- **Lập trình hàm:**
- **Lập trình logic**

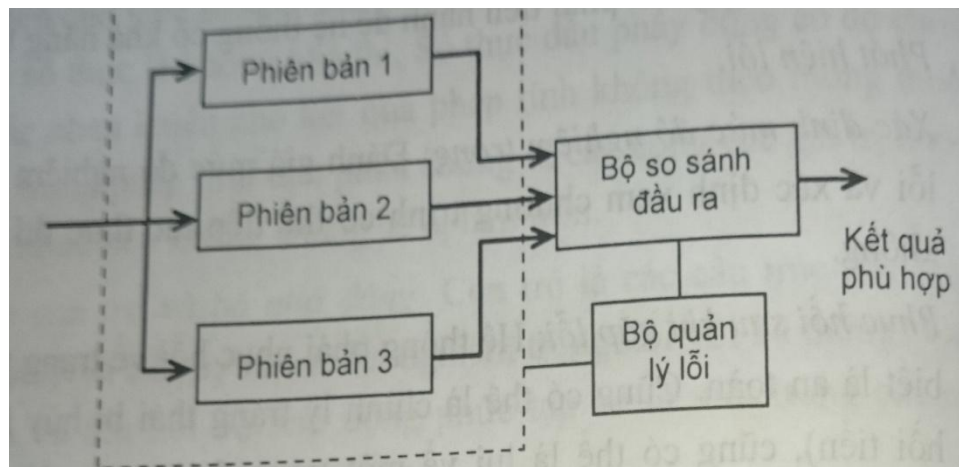
4.3. phong cách lập trình

Phong cách lập trình gồm các yếu tố:

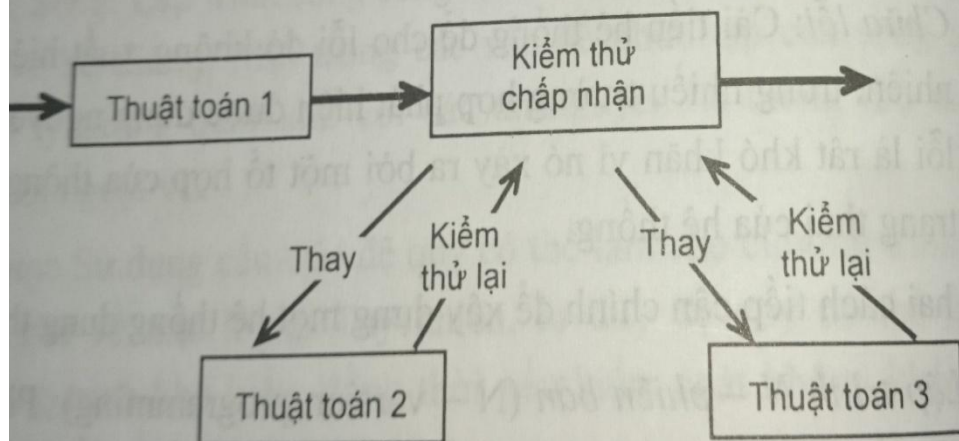
- **Tài liệu bên trong chương trình:** nhằm giải thích rõ từng đoạn chương trình.
- **Khai báo dữ liệu:** quy cách đặt tên cho các biến, hàm... cần chuẩn hóa, mang tính gợi nhớ. TD: BSinhVien- bảng SinhVien, HTinhTong- hàm tính tổng,...
- **Xây dựng câu lệnh:** quy tắc: mỗi câu lệnh nên đơn giản và trực tiếp
- **Phong cách vào/ra của các Module**
 - Kiểm tra tính hợp lệ của dữ liệu vào
 - Giữ cho định dạng dữ liệu vào đơn giản và thống nhất
 - Giảm độ ghép nối

4.5. lập trình tránh lỗi

- **Tránh lỗi:**
 - Cần phân tích và đặc tả chính xác các yêu cầu
 - Tăng cường duyệt lại, thẩm định hệ thống
 - Tổ chức nghiệm thu có hệ thống
 - Áp dụng tư duy thiết kế hiện đại: module hóa và HĐT
- **Lập trình dung thứ lỗi:** đảm bảo hệ thống vẫn hoạt động được ngay cả khi có TP sinh lỗi.
 - Phát hiện lỗi
 - Xđ mức độ nghiêm trọng
 - Phục hồi sau khi gặp lỗi
 - Chữa lỗi



Hình 4.2. Lập trình n – phiên bản

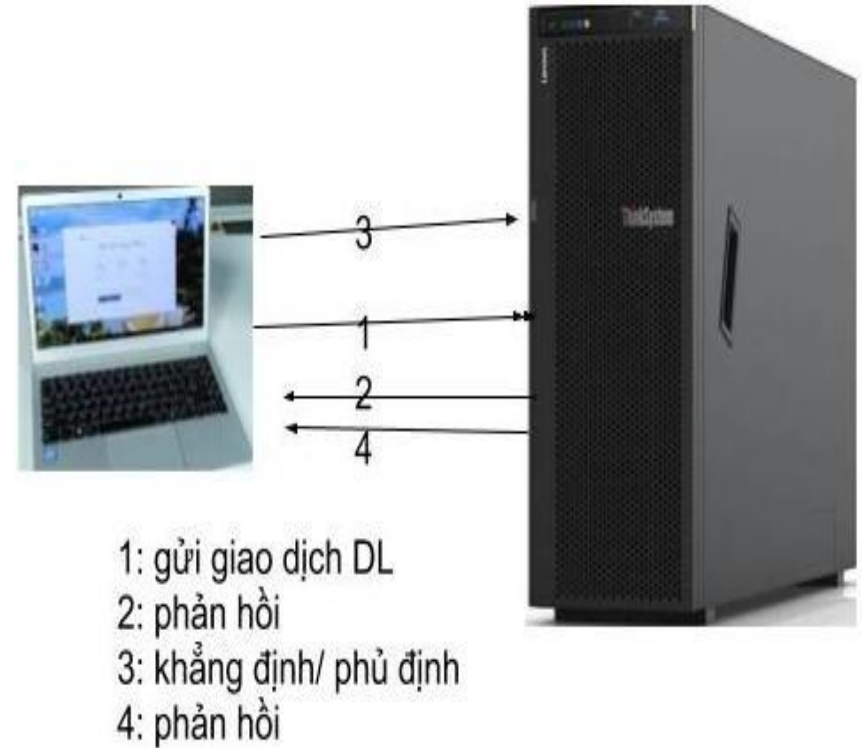


Hình 4.3. Khôi phục hồi

4.5. lập trình tránh lỗi

- **Lập trình phòng thủ:**

- chú trọng đến các lỗi có tính khách quan.
- Kiểm tra trạng thái hệ thống sau khi biến đổi. Nếu phát hiện mâu thuẫn → hủy biến đổi trạng thái + quay lại trạng thái đúng đắn trước đó (TD: giao dịch cập nhật DL 2 bước)



- **Xử lý ngoại lệ**

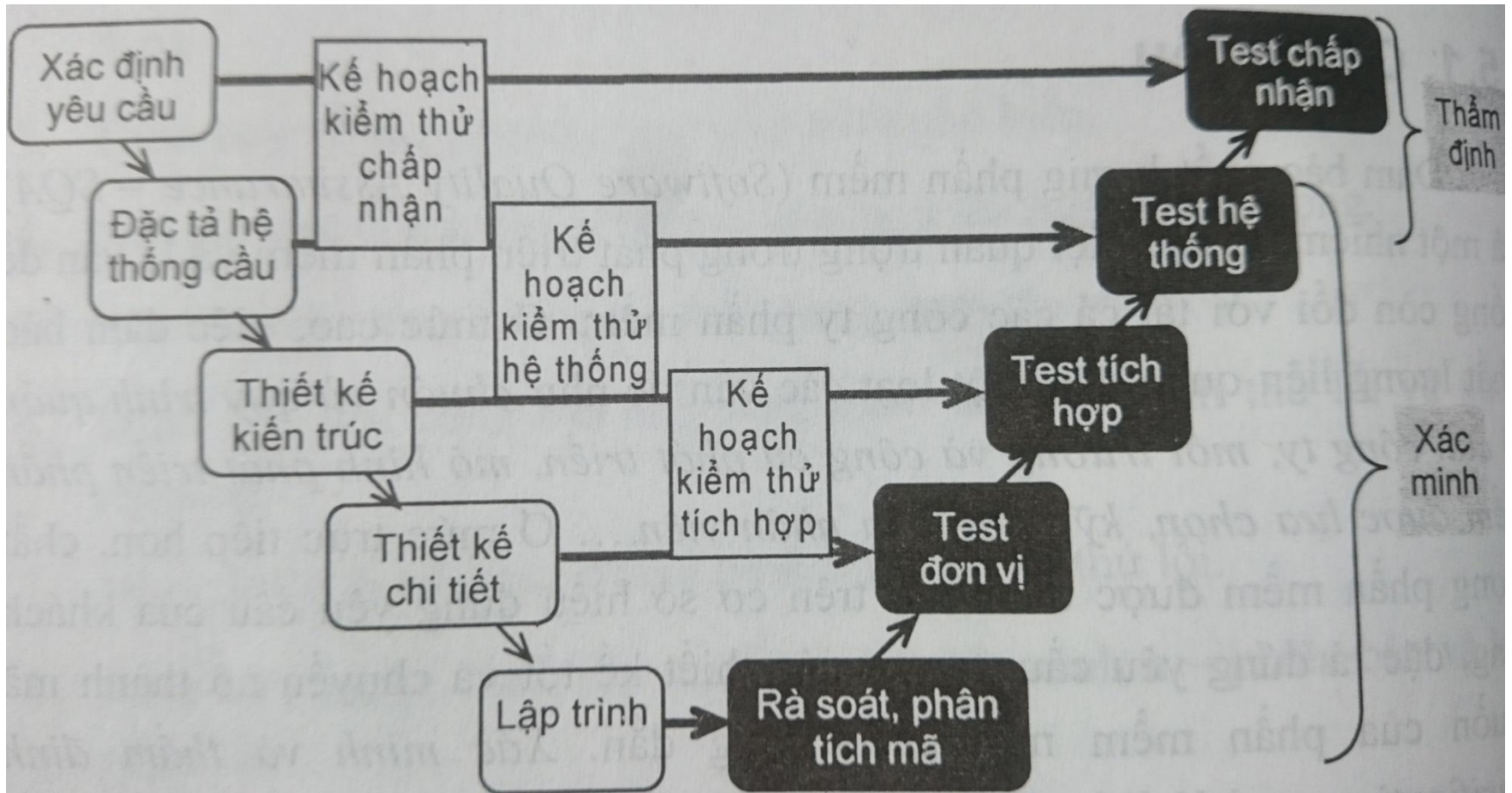
Chương 5. xác minh và thẩm định

- Đảm bảo chất lượng phần mềm (Software Quality Assurance- SQA) là nv đặc biệt quan trọng của phát triển phần mềm
 - Đảm bảo hiểu đúng yêu cầu của khách hàng
 - Đặc tả đúng yêu cầu
 - Thiết kế tốt
 - Lập trình tốt
- Xác minh và thẩm định (Verification & Validation- V&V) là các mặt của SQA:
 - **Xác minh:** là kiểm tra xem SP có đúng đặc tả k? chú trọng đến phát hiện lỗi của phần mềm
 - **Thẩm định:** kiểm tra xem sp có đáp ứng đc yêu cầu của khách hàng k? chú trọng đến sự khác biệt của SP làm ra với mong đợi của khách hàng.

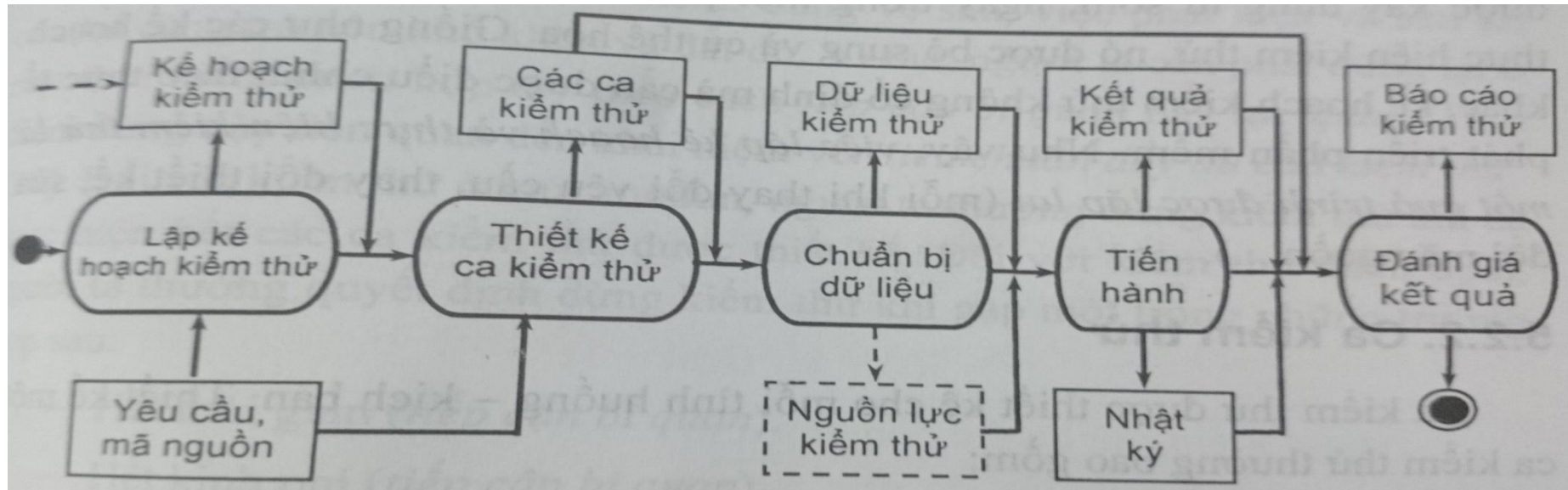
Chương 5. xác minh và thẩm định

- SQA gồm các hoạt động :
 - Rà soát kĩ thuật
 - Kiểm toán cấu hình
 - Kiểm toán chất lượng
 - Giám sát hiệu năng
 - Nghiên cứu khả thi và mô phỏng
 - Rà soát tài liệu
 - Rà soát CSDL
 - Phân tích thuật toán
 - Kiểm thử mã nguồn
 - Kiểm thử chất lượng
 - Kiểm thử cài đặt
- **GĐ phân tích YC:** thẩm định đặc tả yêu cầu có đúng với yêu cầu thực tế?
- **GĐ Thiết kế:** xem thiết kế có đúng đặc tả k?
- **GĐ mã hóa:** rà soát xem mã nguồn có đúng thiết kế chi tiết k?

Mô hình thẩm định V



5.2. kiểm thử



- Kiểm thử phần mềm là quá trình vận hành chương trình để tìm ra lỗi
- Cần thiết kế các ca Kiểm thử để có thể phát hiện các loại lỗi với chi phí TG và công sức ít nhất
- quá trình kiểm thử cần có chiến lược & PP tiếp cận
- Kiểm thử Cần được chuẩn bị, đặc tả và ghi chép lại

5.3. các phương pháp và chiến lược kiểm thử

- **Chiến lược kiểm thử:** là 1 dãy các bước mô tả, hướng dẫn quá trình kiểm thử
- **Kiểm thử hộp trắng:** là kiểm tra các đoạn mã chương trình xem nó có vận hành đúng như thiết kế không
- **Kiểm thử hộp đen:** nhập các dữ liệu đầu vào qua giao diện, thực thi nó và xem kq đầu ra có đúng không

5.4. các loại hình và mức kiểm thử

Mức	Loại hình	ĐƠN VỊ	TÍCH HỢP	HỆ THỐNG	THẨM ĐỊNH
ĐỐI TƯỢNG		Module đơn vị	Tập modules chức năng	Hệ thống phần mềm, môi trường đích	Hệ thống phần mềm, môi trường thực
CƠ SỞ SO SÁNH		Thiết kế chi tiết	Thiết kế chi tiết, kiến trúc, đặc tả chức năng	Thiết kế kiến trúc, đặc tả phần mềm	Đặc tả yêu cầu phần mềm
PHƯƠNG PHÁP		Hộp trắng (hộp đen)	Hộp đen (hộp trắng)	Hộp đen, mô hình	Hộp đen
NGƯỜI LÀM		Lập trình viên	Nhóm kiểm thử độc lập	Nhóm kiểm thử chuyên	Người dùng

5.5. các công việc kiểm thử

- Kiểm tra chức năng, luồng dữ liệu nghiệp vụ
- Kiểm tra giao diện người sử dụng
- Kiểm tra hiệu năng
- Kiểm tra tính bảo mật và điều khiển truy nhập
- Kiểm tra tính dùng được

5.6. Thí dụ tài liệu kiểm thử

- Kiểm thử giao diện

Người test.....Ngày tháng....thời gian....

STT	Yêu cầu test	Yêu cầu kết quả	Kết quả
1	Các màu sắc hiển thị trang web	Các màu sắc của các mục, liên kết, đường dẫn phải đúng như thiết kế	True
2	Kích thước của các đối tượng trên web	Kích thước các đối tượng không bị thay đổi so với ban đầu, hiển thị tốt cho mọi client browser	-Màn hình độ phân giải 768X1024 trở lên OK; -thấp hơn bị vỡ khung
..

5.6. Thí dụ tài liệu kiểm thử

- Kiểm thử chức năng

Người test.....Ngày tháng....thời gian....

STT	Yêu cầu test	Yêu cầu kết quả	Kết quả
1	Kiểm thử chức năng duyệt danh sách các sản phẩm	Các sản phẩm phải được sắp xếp theo danh mục từng loại	Pass
2	Kiểm thử chức năng quản lí giỏ hàng	Thực hiện thêm, xóa các mặt hàng trong giỏ, tổng số tiền của các mặt hàng phải đúng với công thức $\Sigma(\text{số lượng một sản phẩm} * \text{giá/ một sản phẩm đó})$	Lỗi khi thêm 1 mặt hàng nhiều lần
..

5.7. gỡ lỗi

- Phương pháp thường dùng: ***Loại trừ***
- Công cụ trợ giúp gỡ lỗi: kiểm thử tự động:
 - Bộ phân tích tĩnh
 - Bộ kiểm toán mã
 - Bộ xử lý khẳng định
 - Bộ sinh tệp kiểm thử
 - Bộ kiểm chứng kiểm thử
 - Dụng cụ kiểm thử
 - Bộ so sánh đầu ra
 - Hệ thống thực hiện kí hiệu
 - Bộ mô phỏng môi trường
 - Bộ phân tích luồng dữ liệu

Một số công cụ kiểm thử tự động:

- Cactus
- ESC/ Java
- FlawFinder
- Quick Test Professional của HP
- Rational Rose của IBM
- MVC framework của MS đã tích hợp vào trong bộ Visual Studio
- JUnit – mã nguồn mở...

Công cụ kiểm thử an ninh:

- ITS4
- MOPS
- RATS...

Hướng dẫn làm bài tập lớn

- Khảo sát yêu cầu, xây dựng đặc tả yêu cầu
 - Khảo sát sơ bộ → TL khảo sát sơ bộ + kế hoạch phân công công việc.
 - Khảo sát chi tiết yêu cầu → tài liệu khảo sát + phân tích yêu cầu → TL đặc tả yêu cầu
- Phân tích thiết kế
 - Phân tích + kiểm tra đặc tả YC → TL bổ sung + TL phân tích (các sơ đồ)
 - TK: → TL thiết kế chức năng/ thiết kế HĐT, thiết kế CSDL, thiết kế giao diện
- Lập trình
 - Các module phần mềm : các đoạn mã
- Kiểm thử phần mềm
 - Kiểm thử các module → TL kiểm thử
- Thẩm định phần mềm
 - Thẩm định phù hợp đặc tả YC → TL Đánh giá
 - Thẩm định của người sử dụng → TL tổng hợp các ý kiến đánh giá của NSD.

Khảo sát và xây dựng đặc tả yêu cầu

- Khảo sát yêu cầu, xây dựng đặc tả yêu cầu
 - Khảo sát sơ bộ → TL khảo sát sơ bộ + kế hoạch phân công công việc.
 - Khảo sát chi tiết yêu cầu → tài liệu khảo sát + phân tích yêu cầu → TL đặc tả yêu cầu
- Khảo sát sơ bộ: Có 3 tác nhân ngoài: Khách hàng, Nhân viên, Lãnh đạo →

Bảng phân công công việc (nhóm trưởng)

Mã CV	Công việc	Người làm	Thời hạn	Kết quả
YC1	Khảo sát chi tiết yêu cầu Khách Hàng: TL đặc tả YC Khách Hàng	A	1/4/2020	Chưa nộp
YC2	Khảo sát chi tiết yêu cầu Nhân Viên: TL đặc tả YC Nhân Viên	B	1/4/2020	Chưa nộp
YC3	Khảo sát chi tiết yêu cầu Lãnh Đạo: TL đặc tả YC Lãnh Đạo	C	1/4/2020	Chưa nộp
YC4	Kiểm tra, Tổng hợp yêu cầu, TL đặc tả yêu cầu	D	3/4	Chưa
TK1	Xác minh, phân tích thiết kế YC1	B	15/4	Chưa
TK2	Xác minh, phân tích thiết kế YC2	C	15/4	Chưa
TK3	Xác minh, phân tích thiết kế YC3	D	15/4	Chưa
TK4	Kiểm tra, TL Tổng hợp thiết kế	A	17/4	Chưa

Bảng phân công công việc (nhóm trưởng)

Mã CV	Công việc	Người làm	Thời hạn	Kết quả
C1.1	Xây dựng lớp Khách Hàng	D	30/4	Chưa nộp
C1.2	xd cơ sở dữ liệu Khách hàng và cài đặt SQL Server	D	30/4	Chưa nộp
C1.3	Làm các Form cho Khách Hàng	D	30/4	Chưa nộp
..
T1.1	Kiểm thử C1.1	A	2/5	Chưa
T1.2	Kiểm thử C1.2	A	2/5	Chưa
T1.3	Kiểm thử C1.3	A	2/5	Chưa