

Công nghệ phần mềm

Chương 01: Tổng quan về phát triển phần mềm

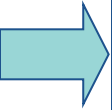


Giảng viên: Lê Thị Hoàng Anh

Email: anhlt@huce.edu.vn



Nội dung



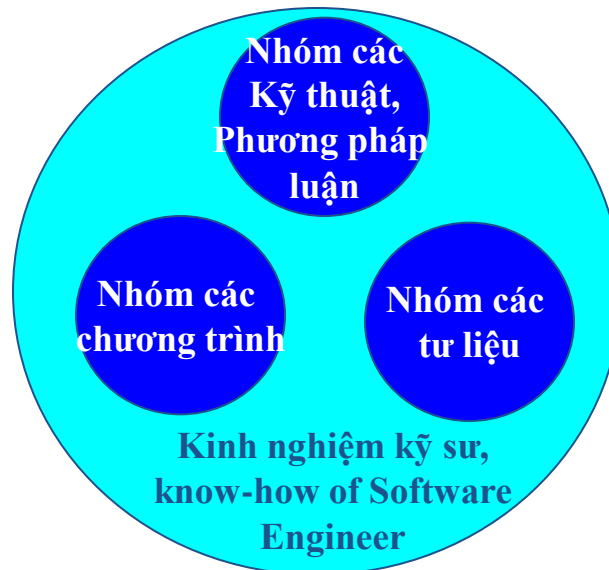
1. Phần mềm

- Định nghĩa phần mềm
- Lịch sử phát triển phần mềm
- Các đặc trưng phần mềm
- Phân loại phần mềm
- Các thành phần của phần mềm
- Việc ứng dụng phần mềm
- Các thách thức đối với phần mềm máy tính



Phần mềm

- Phần mềm software:
 - Nghĩa hẹp: Là các chương trình máy tính và những tài liệu liên quan (tài liệu đặc tả yêu cầu, tài liệu phân tích thiết kế, tài liệu lập trình, tài liệu kiểm thử, tài liệu hướng dẫn sử dụng...)
 - Nghĩa rộng:





Lịch sử phát triển phần mềm

- Sự tiến hóa của phần mềm gắn liền với sự tiến hóa của phần cứng và có thể chia làm bốn giai đoạn:
 - 1950-1960:
 - Phần cứng thay đổi liên tục, phần lớn phần mềm được chuyên dụng cho ứng dụng đặc biệt.
 - Phần mềm được coi là nghệ thuật “theo bản năng”
 - Phát triển phần mềm chưa được quản lý
 - Môi trường phần mềm có tính cá nhân dẫn đến thiết kế & tiến trình không tường minh, thường không có tài liệu



Lịch sử phát triển phần mềm (tiếp)

- 1960-1970:
 - Hệ thống đa người dùng dẫn đến khái niệm mới về tương tác người máy.
 - Hệ thống thời gian thực.
 - Tiến bộ lưu trữ trực tuyến dẫn đến thế hệ đầu tiên của hệ QTCSDL.
 - Số lượng các hệ thống dựa trên máy tính phát triển dẫn đến thư viện phần mềm mở rộng.



Lịch sử phát triển phần mềm (tiếp)

- 1970-1990:
 - Hệ thống phân tán
 - Mạng diện rộng và cục bộ, số giải thông cao, tăng nhu cầu thâm nhập dữ liệu dẫn đến yêu cầu lớn phát triển phần mềm
 - Sử dụng phổ cập các bộ vi xử lý (ô tô, rô bốt, lò vi sóng ...) máy tính cá nhân và các máy trạm
 - Chi phí cho phần mềm tăng nhanh và có khuynh hướng vượt chi phí mua phần cứng.



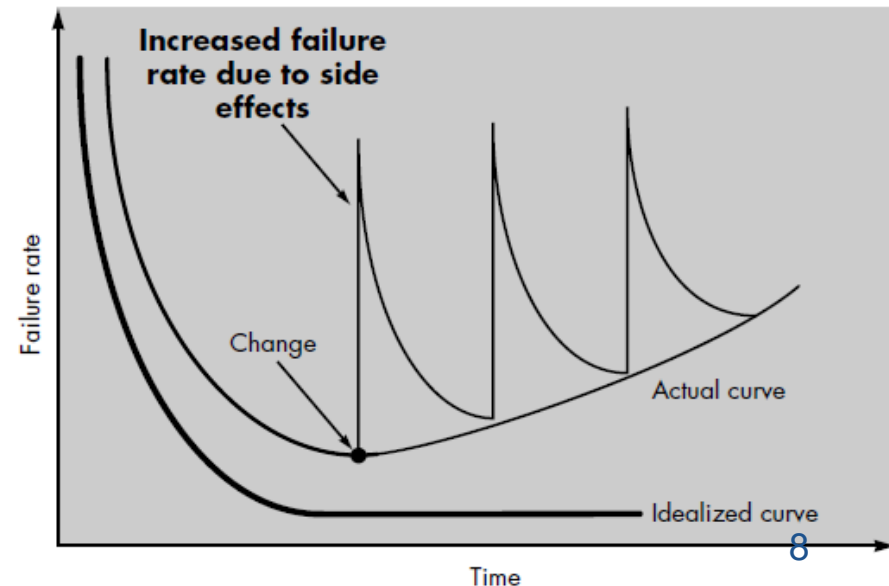
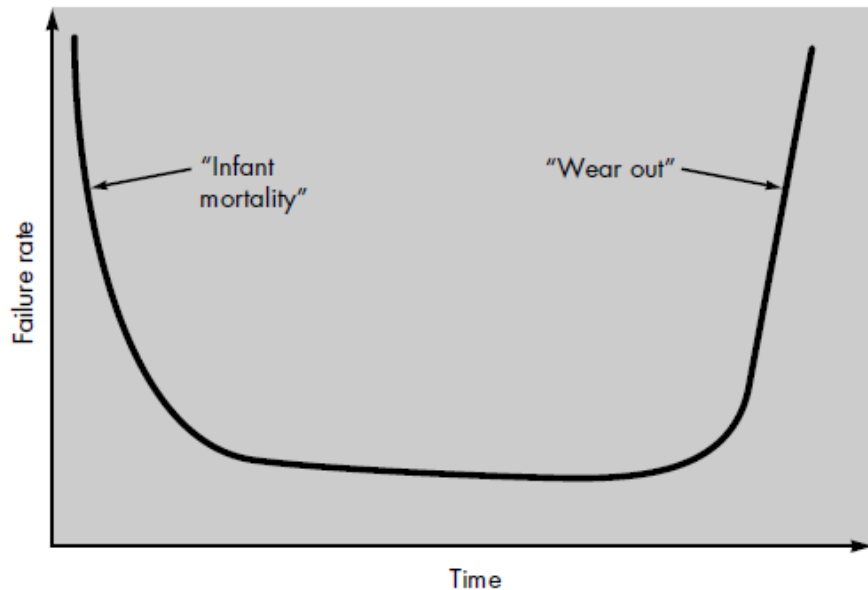
Lịch sử phát triển phần mềm (tiếp)

- Sau 1990:
 - Kỹ nghệ hướng đối tượng
 - Hệ chuyên gia và phần mềm trí tuệ nhân tạo
 - Phần mềm mạng nơ ron nhân tạo
 - Kỹ thuật phần mềm hướng thành phần (COM, DCOM của Microsoft, CORBA của OMG, JavaBeans, Enterprise JavaBeans của SUN)
 - Kỹ nghệ hướng dịch vụ SOA



Các đặc trưng của phần mềm

- Phần mềm được phát triển hay kỹ nghệ hóa chứ không phải sản xuất, chế tạo.
- Phần mềm không bị hao mòn như phần cứng mà có xu hướng tốt lên sau mỗi lần phát hiện lỗi và sửa lỗi





Các đặc trưng của phần mềm (tiếp)

- Từ đặc trưng thứ hai chúng ta có thể suy rộng ra:
 - Chức năng phần mềm thường biến hóa, thay đổi theo thời gian và nơi sử dụng.
 - Hiệu ứng lan sóng trong thay đổi phần mềm.
 - Cần khả năng tư duy nhị phân trong xây dựng phát triển phần mềm.



Các đặc trưng của phần mềm (tiếp)

- Phần mềm phần lớn đều được xây dựng theo đơn đặt hàng chứ ít khi được lắp ráp từ những thành phần có sẵn.
- Phần mềm vốn chứa lỗi tiềm tàng, theo quy mô càng lớn thì khả năng chứa lỗi càng cao.
- Lỗi phần mềm dễ được phát hiện bởi người ngoài.
- Phần mềm có thể sao chép rất đơn giản.



Phần mềm

- Phân loại phần mềm
 - Sản phẩm phần mềm có nhiều cách phân loại:
 - Theo phương thức hoạt động: Phần mềm hệ thống, phần mềm ứng dụng, phần mềm dịch mã (trình dịch), Nền tảng ứng dụng...
 - Theo người dùng: Phần mềm đại trà và phần mềm chuyên biệt.
 - Theo khả năng hay quyền hạn can thiệp vào mã nguồn:
 - Phần mềm mã nguồn đóng (closed source software)
 - Phần mềm mã nguồn mở (open source software)

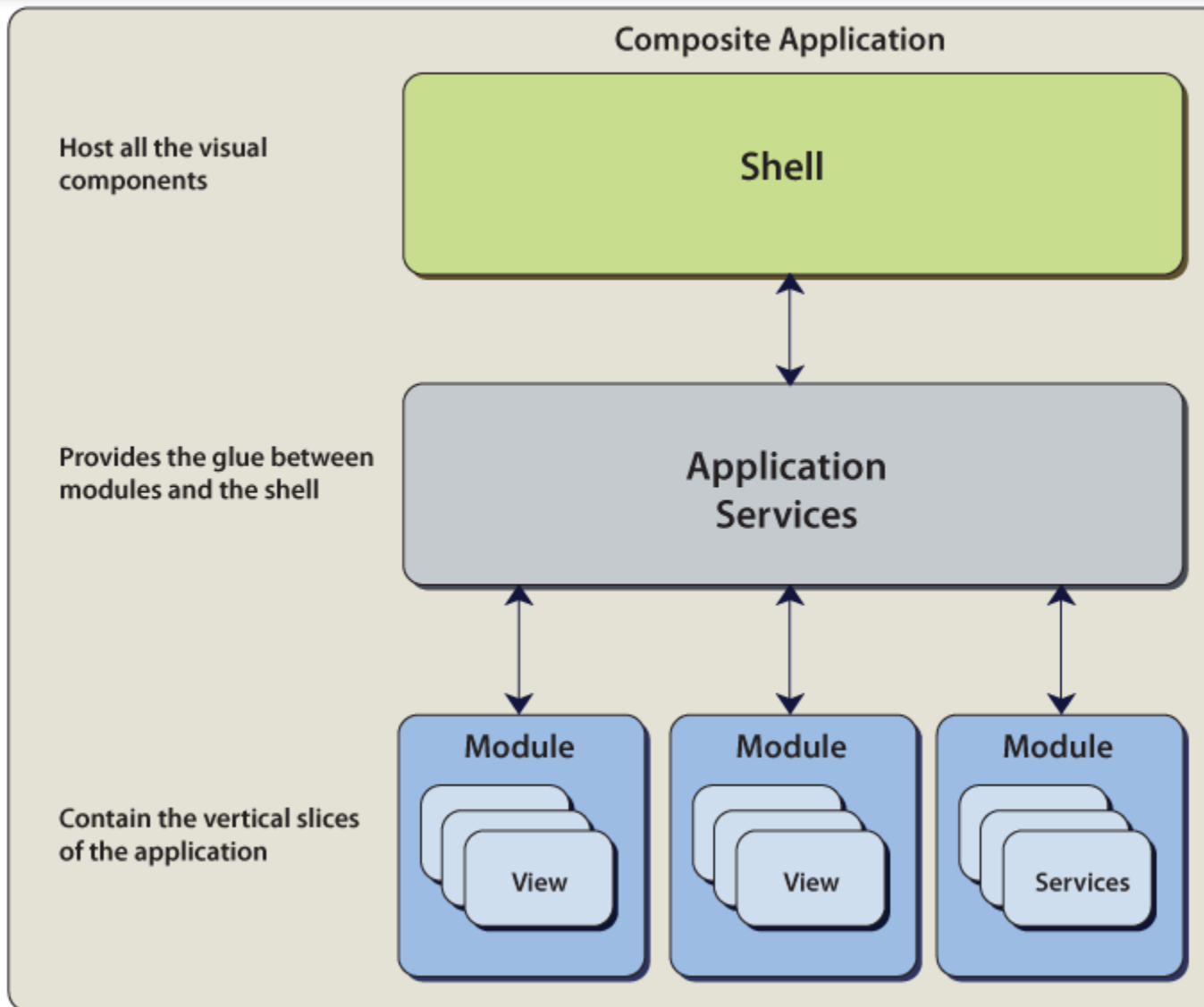


Các thành phần của phần mềm

- Phần mềm bao gồm:
 - Các lệnh (chương trình) khi được thực hiện trên máy vi tính thì đưa ra hoạt động và kết quả mong muốn.
 - Các cấu trúc dữ liệu hoặc cơ sở dữ liệu làm cho chương trình thao tác thông tin thích hợp.
 - Các tài liệu mô tả thao tác và cách dùng chương trình...

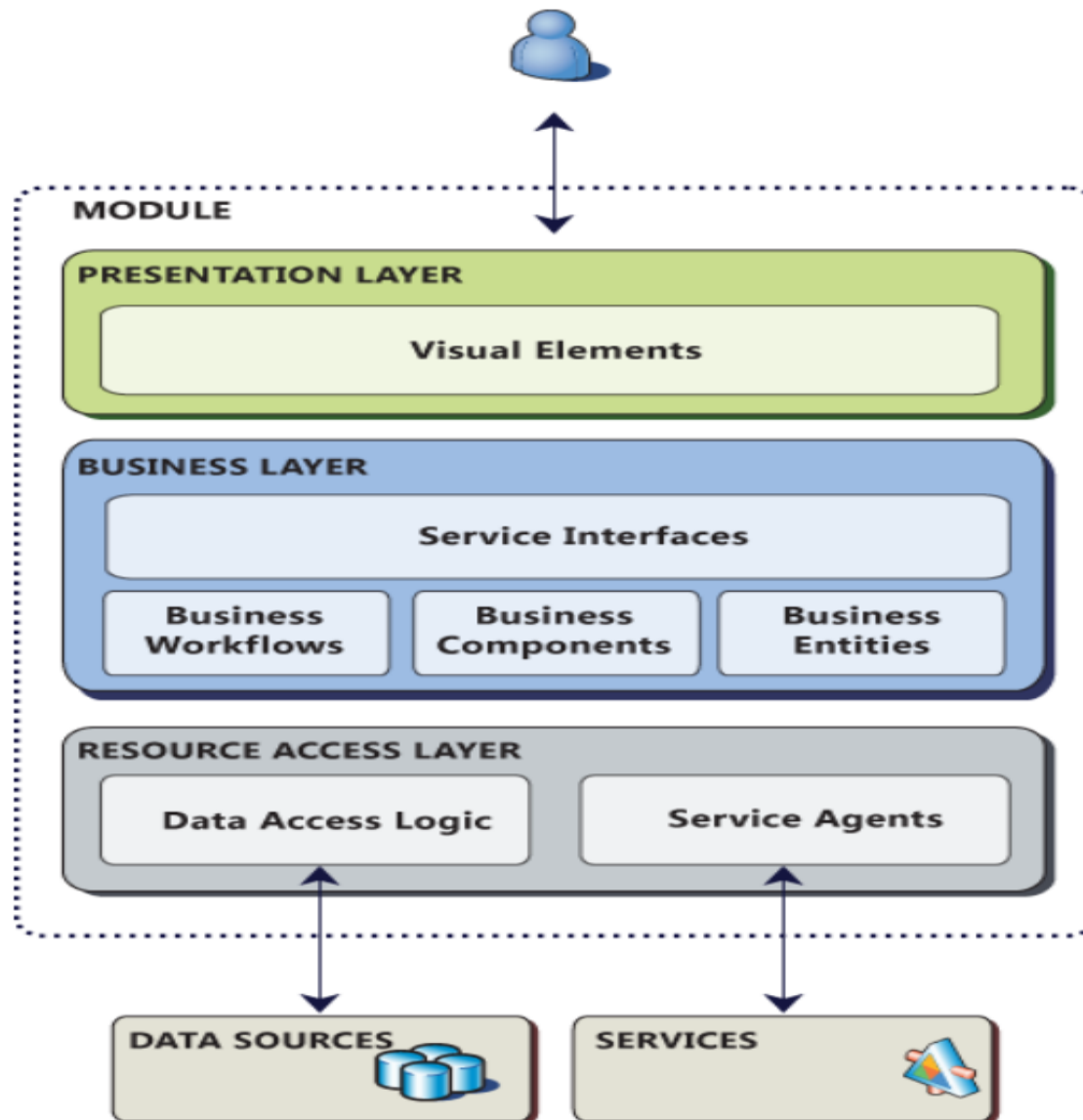


Thành phần ứng dụng với nhiều module



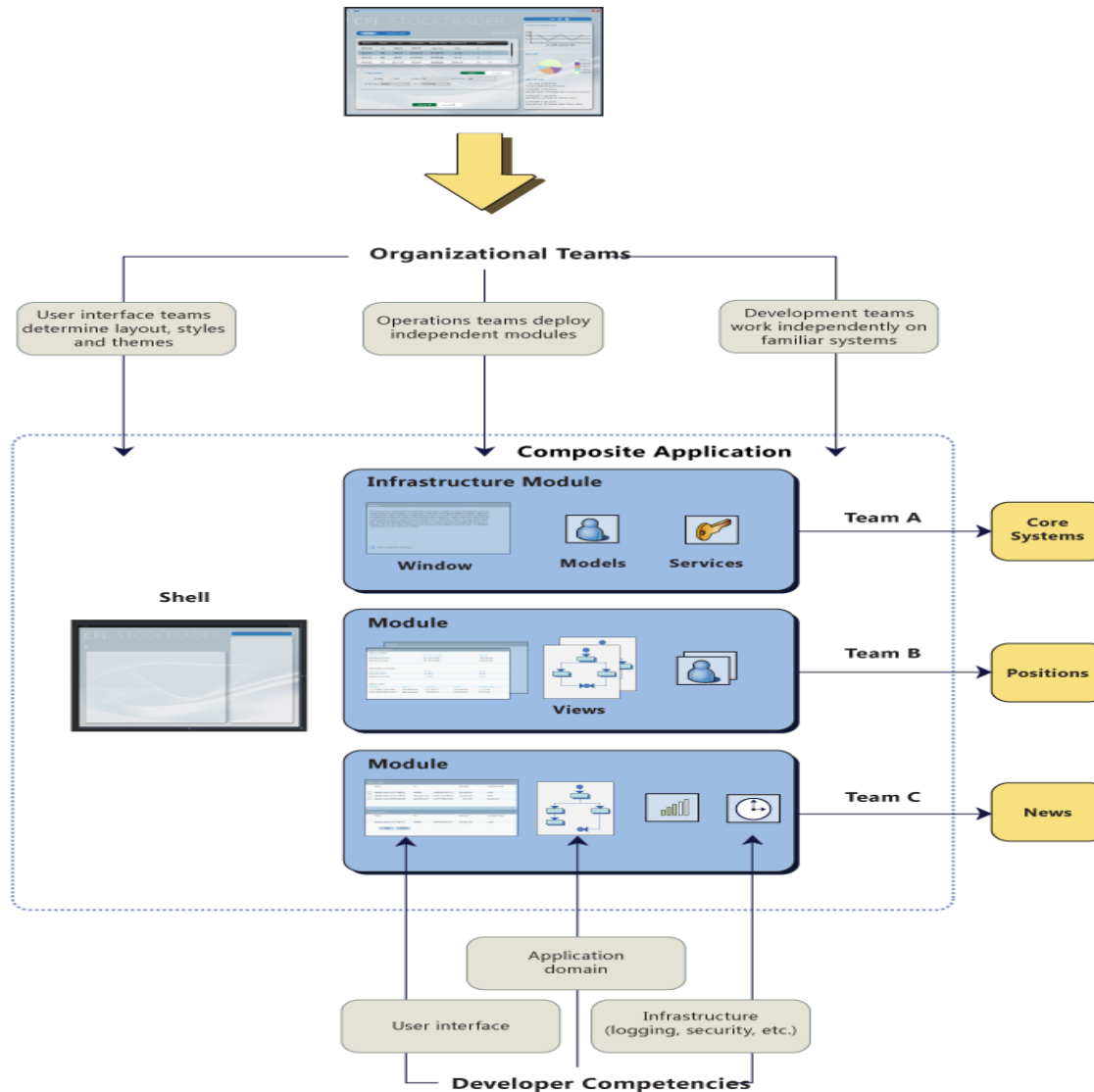
Module composition

Các tầng thành phần trong thiết kế module





Đội phát triển sử dụng module

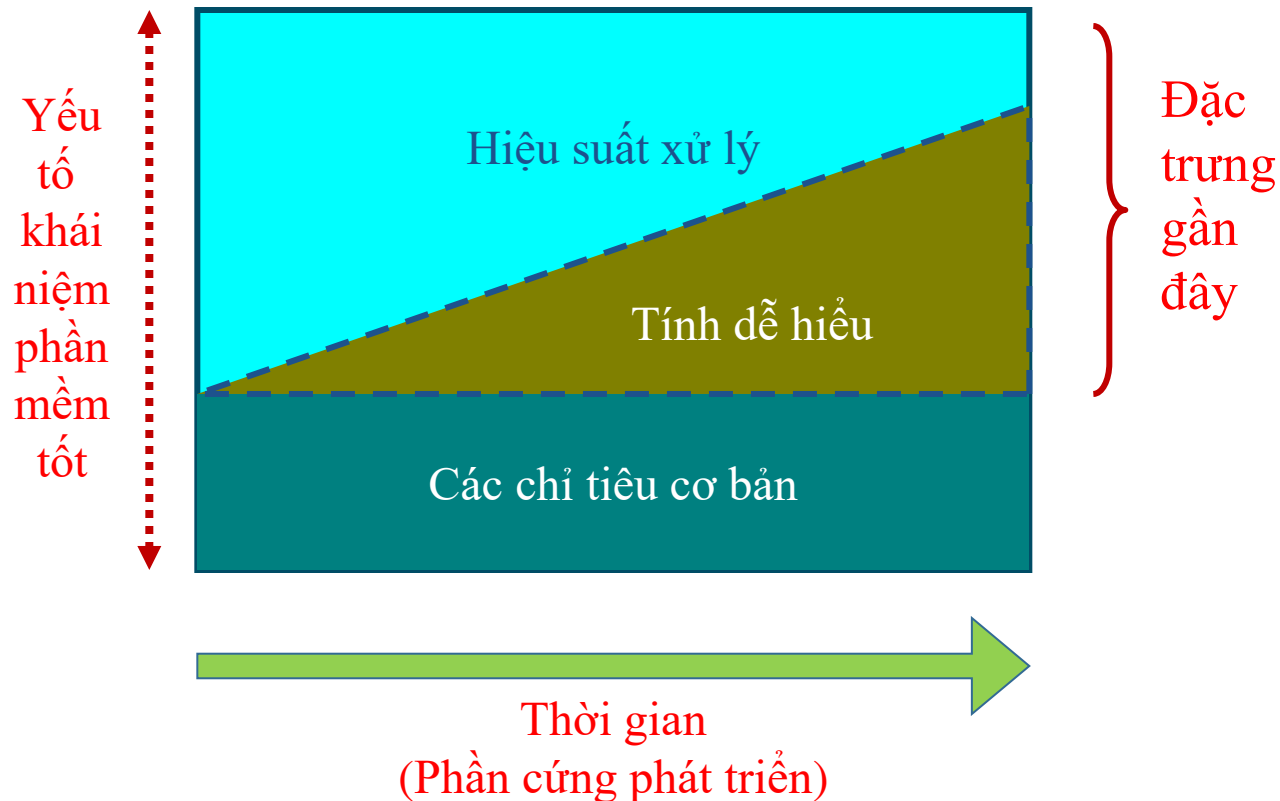


Modules and associated teams
of a composite application



Tiêu chí của một phần mềm tốt

- Để đánh giá chất lượng phần mềm người ta đưa ra các tiêu chí có tính định tính





Các tiêu chí cơ bản

- Phản ánh đúng yêu cầu người dùng (tính hiệu quả effectiveness).
- Chứa ít lỗi tiềm tàng.
- Giá thành không vượt quá giá ước lượng ban đầu.
- Dễ vận hành, sử dụng.
- Tính an toàn và độ tin cậy cao.



Hiệu suất xử lý cao

- Hiệu suất thời gian tốt (efficiency)
 - Độ phức tạp tính toán thấp (time complexity)
 - Thời gian quay vòng ngắn (Turn Around Time TAT)
 - Thời gian hồi đáp nhanh (Response Time)
- Sử dụng tài nguyên hữu hiệu: CPU, RAM, HDD, Internet resources...



Tính dễ hiểu

- Kiến trúc và cấu trúc thiết kế dễ hiểu
- Dễ kiểm tra, kiểm thử, kiểm chứng
- Dễ bảo trì
- Có tài liệu (mô tả yêu cầu, điều kiện kiểm thử, vận hành, bảo trì, FAQ...) với chất lượng cao.

Tính dễ hiểu chỉ tiêu ngày càng quan trọng



Việc ứng dụng phần mềm

- Ứng dụng CNTT vào trong nhiều lĩnh vực đời sống:
 - Giải trí
 - Giáo dục
 - Y tế
 - Công tác quản lý hành chính và kinh doanh
 - Tìm kiếm việc làm
 - Ngành quốc phòng
 - Lĩnh vực tài chính...

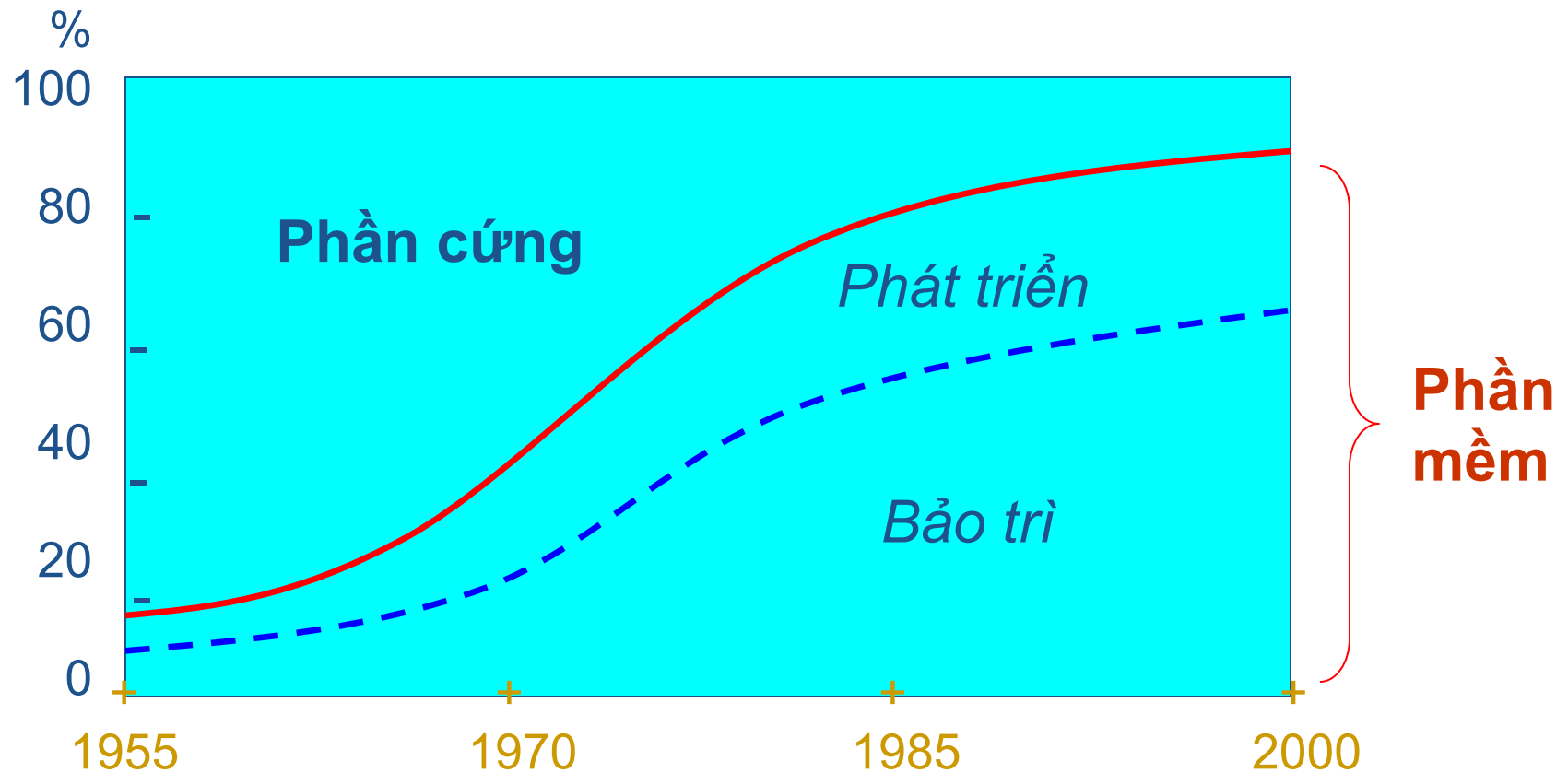


Phát triển phần mềm và các thách thức

- Một số yếu tố khi phát triển phần mềm:
 - Phần mềm càng lớn sẽ kéo theo phức tạp hóa và tăng chi phí phát triển.
 - Đổi vai trò giá thành phần cứng và phần mềm
 - Công sức cho bảo trì càng tăng thì chi phí cho backlog càng lớn.
 - Nhân lực chưa đáp ứng được nhu cầu phần mềm.
 - Những phiên hà của phần mềm gây ra những vấn đề xã hội.



So sánh chi phí cho phần cứng và phần mềm





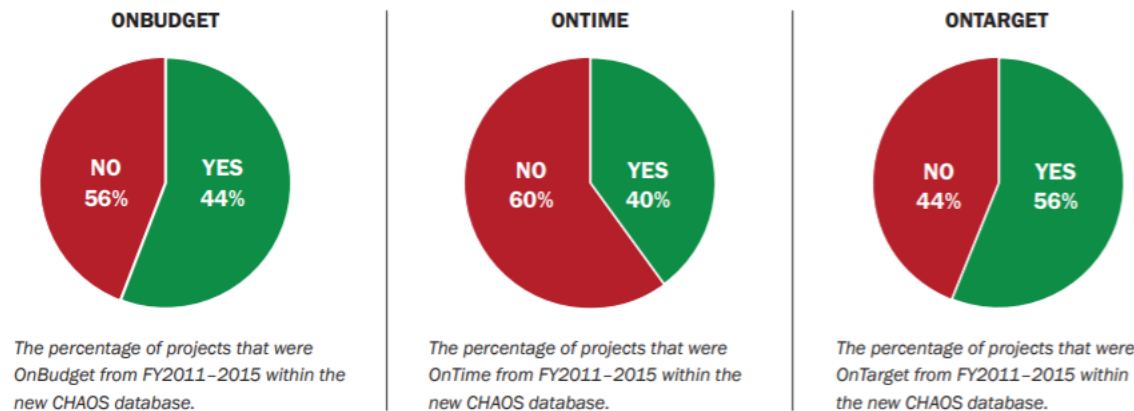
Khó khăn trong phát triển phần mềm

- Không có phương pháp mô tả rõ ràng định nghĩa yêu cầu người dùng(khách hàng) nên sau khi bàn giao sản phẩm dễ có trục trặc. Với những phần mềm quy mô lớn tư liệu đặc tả không thay đổi kịp thời với nhu cầu thay đổi của người dùng.
- Chuẩn về phần mềm tốt không thể đo được một cách định lượng nên không thể đánh giá được một hệ thống đúng đắn hay không do vậy độ tin cậy phần mềm sẽ giảm



Khó khăn trong phát triển phần mềm

- Mất quá nhiều thời gian và chi phí để xây dựng. Theo thống kê báo cáo của CHAOS năm 2015 cho tất cả các dự án:



TRADITIONAL RESOLUTION FOR ALL PROJECTS

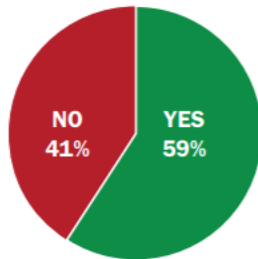
	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

The Traditional resolution of all software projects from FY2011–2015 within the new CHAOS database.



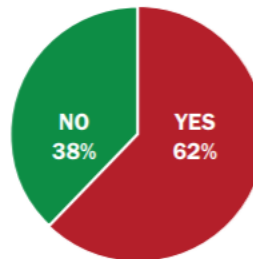
Báo cáo thống kê năm 2015

VALUABLE



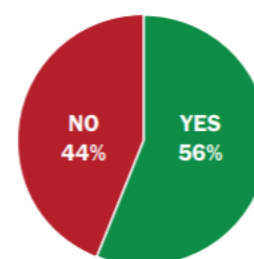
The percentage of projects considered valuable from FY2011-2015 within the new CHAOS database.

ONGOAL



The percentage of projects that were OnGoal from FY2011-2015 within the new CHAOS database.

SATISFACTORY



The percentage of projects considered satisfactory from FY2011-2015 within the new CHAOS database.

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional



Khó khăn trong phát triển phần mềm

- Phần mềm phải khả dụng ở mọi nơi: Điện thoại di động, tích hợp vào tivi ...
- Phần mềm phải khả dụng trong hệ thống cần đảm bảo an toàn như máy năng lượng hạt nhân, máy bay, xe hơi...
- Khó để phát triển phần mềm cho các hệ thống phức tạp như hệ thống phân tán, hệ thống nhúng, hệ thống thời gian thực.
- Với những hệ thống có các yêu cầu bị thay đổi thì phần mềm nhanh chóng bị lạc hậu.



Một số kinh nghiệm khi phát triển phần mềm

- Cần phải có phương pháp luận thiết kế nhất quán và các chuẩn tư liệu về quy trình kỹ nghệ phần mềm để nâng cao chất lượng phần mềm.
- Cần có chiến lược kiểm thử hợp lý để vừa có chi phí hợp lý vừa tránh nguy cơ vượt thời gian khi phát hiện lỗi ở kiểm thử giai đoạn cuối.



Một số kinh nghiệm khi phát triển phần mềm

- Nếu coi trọng việc lập trình hơn khâu thiết kế sẽ làm giảm chất lượng phần mềm.
- Không coi thường việc tái sử dụng phần mềm để tăng năng suất lao động.
- Việc bảo trì kéo dài sẽ làm giảm chất lượng tư liệu gây ảnh hưởng các khâu khác. Tuy chi phí cho bảo trì có xu hướng tăng nhưng nếu đầu tư nhân lực cho bảo trì lớn sẽ làm giảm hiệu suất lao động.



Một số kinh nghiệm khi phát triển phần mềm

- Quản lý dự án chặt chẽ kéo theo quản lý lịch trình rõ ràng đồng thời có tiêu chuẩn để ước lượng nhân lực rõ ràng sẽ tránh được nguy cơ kéo dài vượt thời hạn và kinh phí của dự án.



Nội dung

→ 2. Kỹ nghệ phần mềm

- Định nghĩa
- Mô hình tuần tự tuyến tính
- Mô hình bản mẫu
- Mô hình RAD (Rapid Application Development)
- Mô hình gia tăng
- Mô hình xoắn ốc
- Phát triển Agile và mô hình 4GT



Kỹ nghệ phần mềm

- Định nghĩa:
 - Kỹ nghệ phần mềm là việc thiết lập và sử dụng các nguyên lý công nghệ đúng đắn để xây dựng và thu được phần mềm một cách vừa kinh tế vừa tin cậy và hiệu quả chạy trên các máy thực.
 - Bao gồm các yếu tố chủ chốt:
 - Lý thuyết
 - Phương pháp
 - Công cụ
 - Thủ tục



Quy trình phát triển phần mềm

- Phát triển phần mềm:
 - Gồm một tập các bước phải bao hàm đủ lý thuyết công nghệ áp dụng vào phương pháp, công cụ và thủ tục.
 - Các bước này gồm một tập hợp các hoạt động được phân chia thành các pha chính: phân tích, thiết kế, lập trình, kiểm thử, bảo trì.
 - Quy trình chỉ ra phải thực hiện những công việc gì?
 - Phương pháp chỉ ra cách thực hiện các công việc đó ra sao để nhằm mục đích xây dựng và phát triển phần mềm.



Không có quy trình



- Không biết khi nào hoàn thành do không có phân tích và thiết kế
- Không có cách đánh giá các yêu cầu và tiêu chuẩn chất lượng đã thỏa mãn hay chưa?



Mô hình phát triển phần mềm

- Mô hình phát triển phần mềm:
 - Là một thể hiện trừu tượng của quy trình phát triển phần mềm.
 - Nó biểu diễn các đặc tả về quy trình và cách thức thực hiện các công việc trong quy trình một cách cụ thể bằng cách sử dụng các công cụ.
- Một số loại mô hình:
 - Mô hình tuần tự, mô hình xử lý tăng dần, mô hình quy trình tiến hóa.



Lựa chọn mô hình phát triển

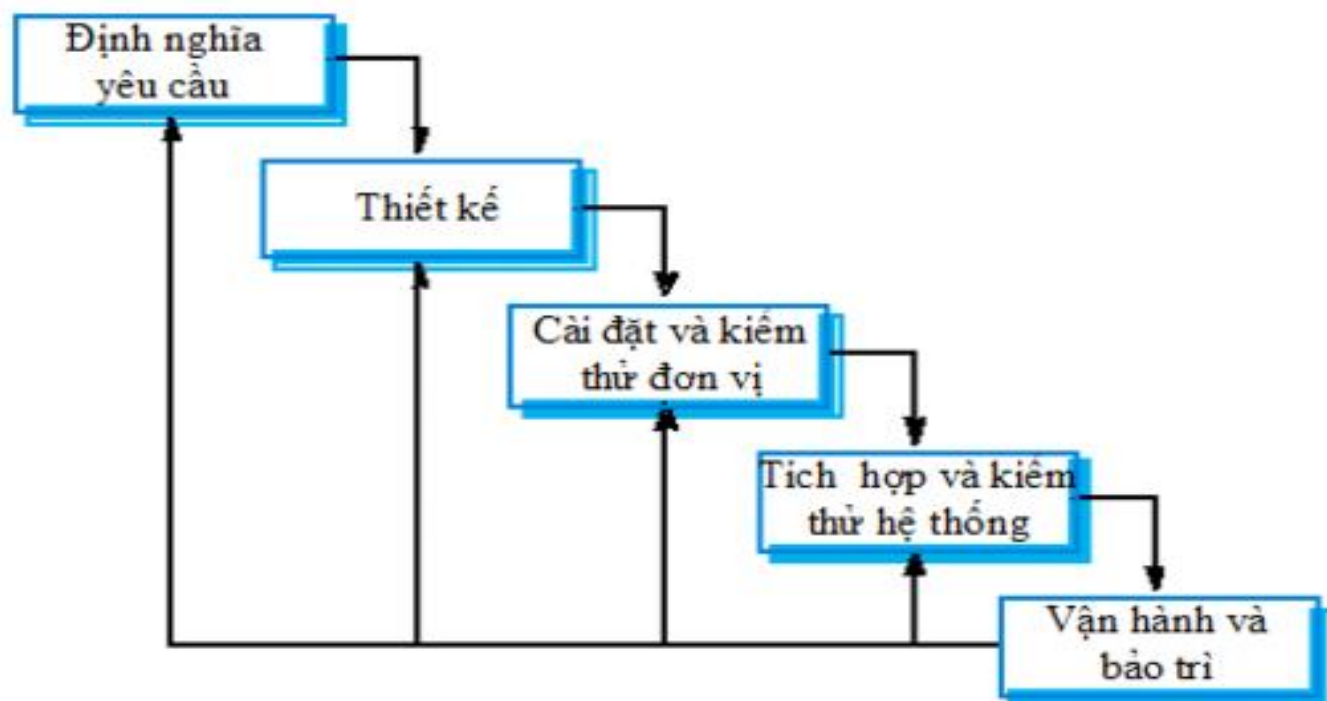
- Việc lựa chọn cần dựa vào các yếu tố:
 - Bản chất của dự án và ứng dụng:
 - Nhận thức rủi ro
 - Sự hiểu biết và kỹ năng của các kỹ sư phần mềm
 - Kiến thức miền ứng dụng của người phát triển
 - Phương pháp và công cụ được dùng
 - Cách thức kiểm soát và các kết quả chuyển giao được yêu cầu



Mô hình tuần tự tuyến tính

(Classic life cycle / waterfall model)

- Hoạt động như sau:





Mô hình tuần tự tuyến tính

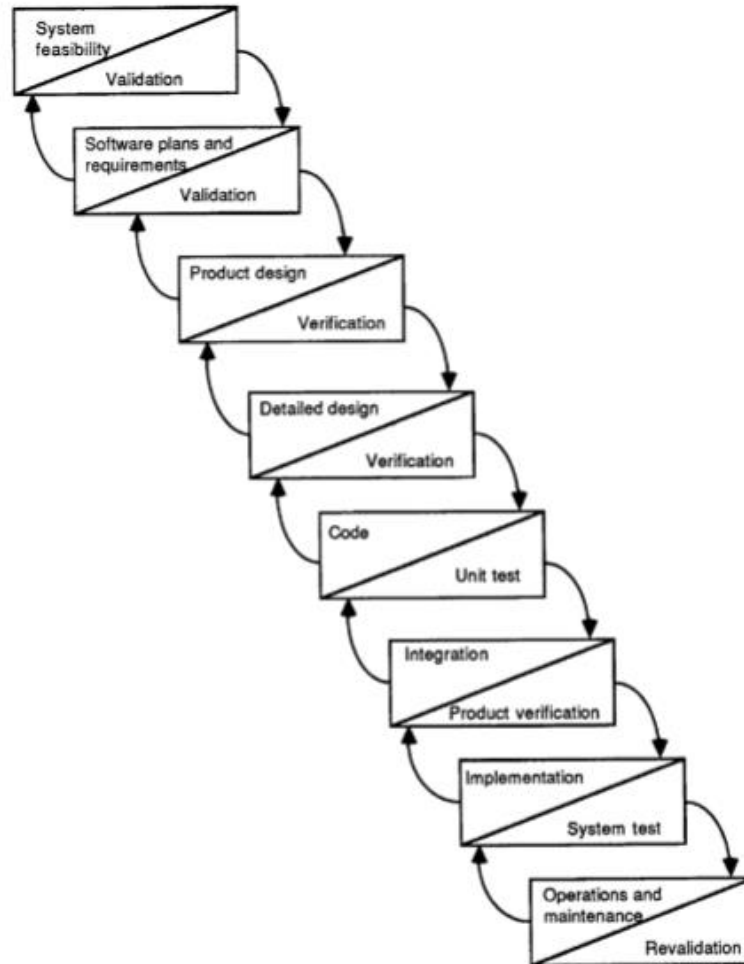


Figure 1. The waterfall model of the software life cycle.



Mô hình tuần tự tuyến tính

- Theo mô hình này các pha được thực hiện trình tự: bắt đầu ở mức hệ thống tiến dần xuống phân tích, thiết kế, mã hóa, kiểm thử và bảo trì.
- Phát triển hệ thống: Công việc phát triển hệ thống là công việc thiết lập ra các yêu cầu cho hệ thống, ánh xạ một số tập con các yêu cầu này sang phần mềm.



Mô hình tuần tự tuyến tính

- Phân tích yêu cầu phần mềm : Từ yêu cầu xác định ở bước trên ta cần
 - Phân loại và sắp xếp nhóm các yêu cầu
 - Thẩm định yêu cầu xem xét khả năng thực thi
 - Xác định rủi ro
- Thiết kế hệ thống và thiết kế phần mềm: Thiết kế hệ thống là thiết kế cấu trúc ngoài của phần mềm, thiết kế cơ bản. Thiết kế phần mềm là thiết kế chi tiết, thiết kế cấu trúc bên trong của phần mềm



Mô hình tuần tự tuyến tính

- Thiết kế là tập trung vào 4 thuộc tính phân biệt của chương trình: Cấu trúc dữ liệu, kiến trúc phần mềm, chi tiết nguyên mẫu thủ tục, đặc trưng giao diện.
- Mã hóa là công việc lập trình viết code.
- Kiểm thử: Phần kiểm thử không chỉ bắt đầu sau khi chúng ta đã viết mã, mà ở đây chúng ta có nhiều loại kiểm thử.
- Vận hành bảo trì: Đáp ứng những thay đổi, nâng cấp phần mềm đã phát triển do sự thay đổi của môi trường, nhu cầu



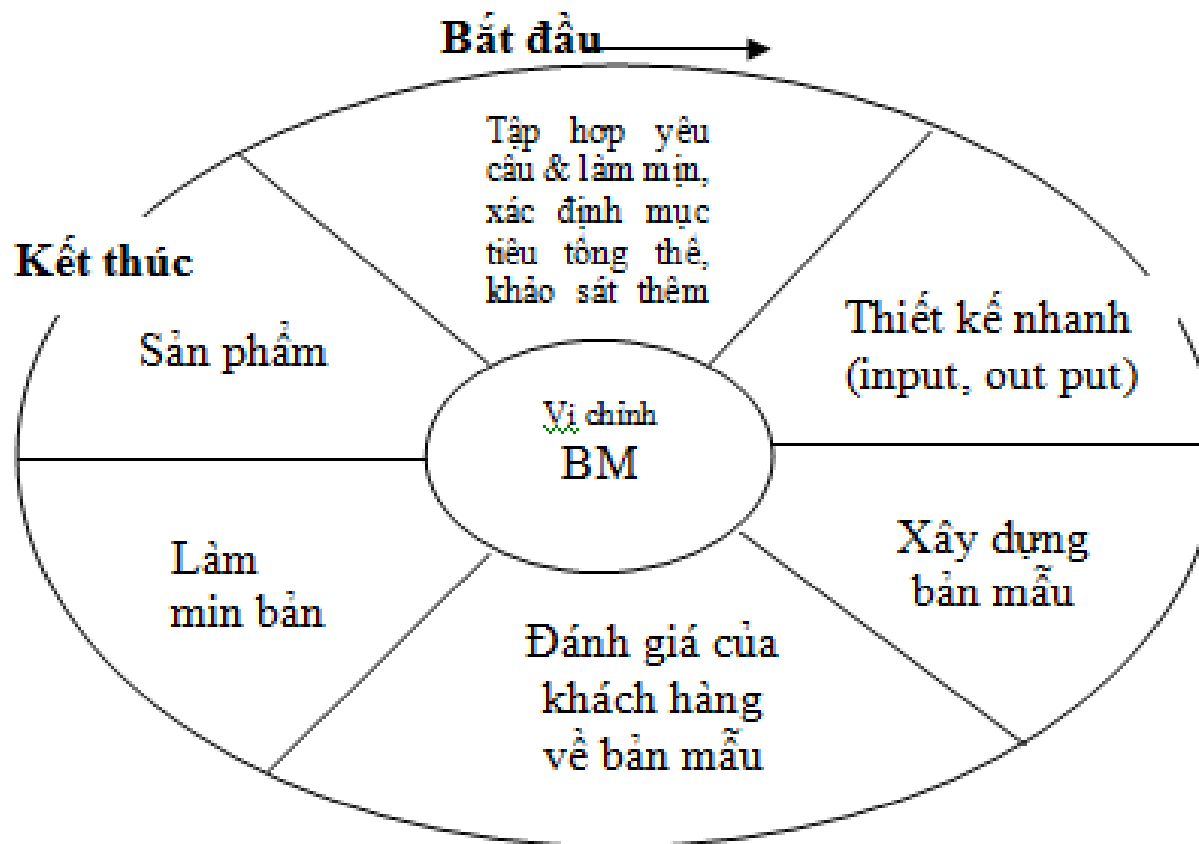
Mô hình tuần tự tuyến tính

- Điểm yếu:
 - Thực tế các dự án ít khi tuân theo dòng tuần tự của mô hình, mà thường có lặp lại (như mô hình của Boehm)
 - Khách hàng ít khi tuyên bố rõ ràng khi nào xong hết các yêu cầu
 - Khách hàng phải có lòng kiên nhẫn chờ đợi thời gian nhất định mới có sản phẩm. Nếu phát hiện ra lỗi nặng thì là một thảm họa!



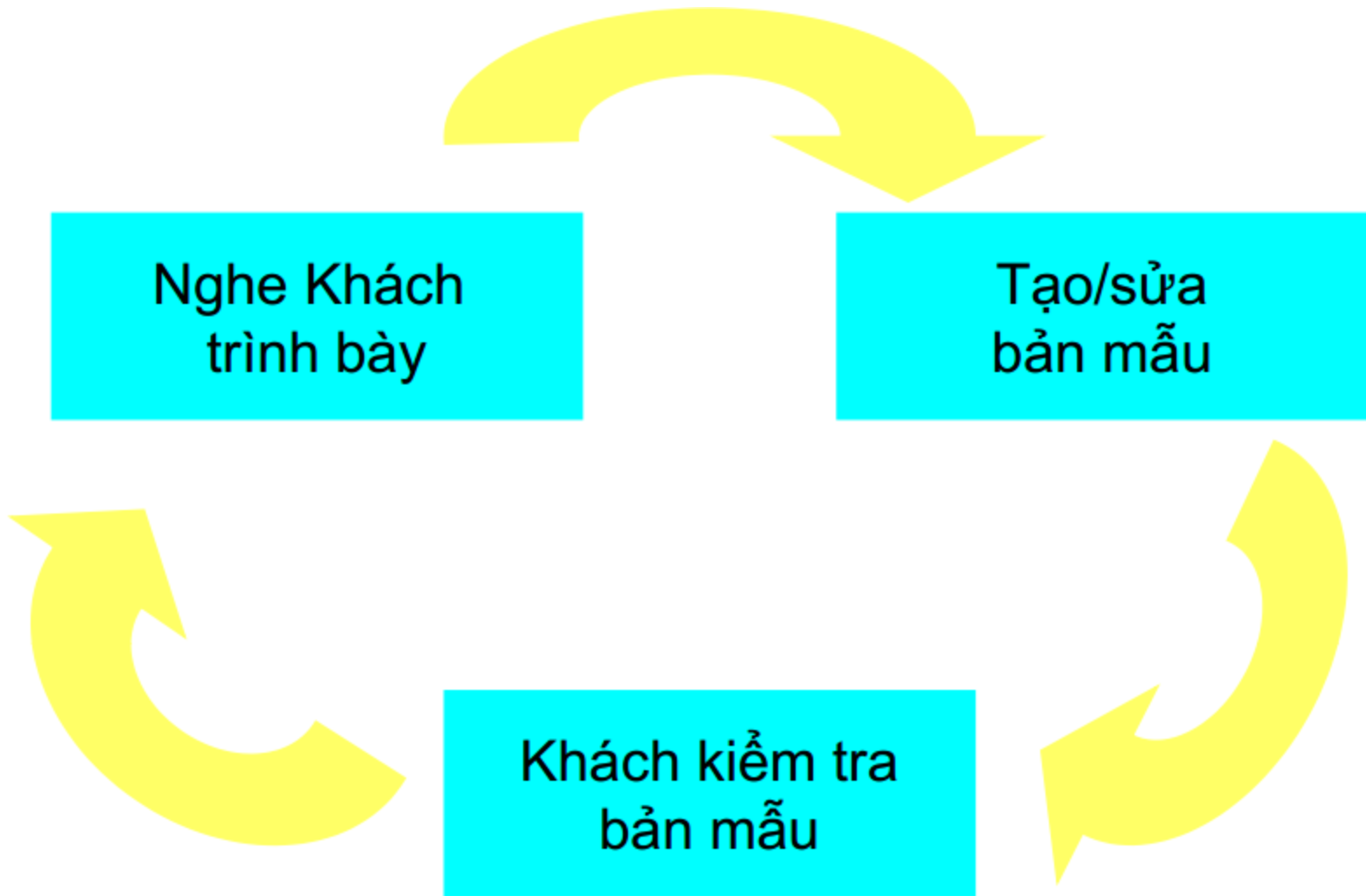
Mô hình chế thử

(Mô hình bản mẫu/prototyping model)





Mô hình chế thử





Mô hình chế thử

- Dùng mô hình bản mẫu khi cần tạo như một “hệ sơ khai” để thu thập yêu cầu người dùng qua các thiết kế nhanh khi:
 - Thông thường khách hàng đã xác định được mục tiêu tổng quát của phần mềm, nhưng chưa xác định được cái vào, xử lý hay yêu cầu cái ra; người phân tích chưa hiểu rõ nhu cầu của khách hàng.
 - Hay khi người phát triển có thể không chắc về tính hiệu quả của một thuật toán hay giải pháp, việc thích nghi hệ điều hành hay dạng giao diện người máy (HCI – *Human Computer Interface*) cần có.



Mô hình chế thử

- Mô hình bản mẫu có thể ở trong 3 dạng:
 1. Bản mẫu trên giấy hay mô hình dựa trên máy PC mô tả giao diện người máy dưới dạng làm cho người dùng hiểu được cách các tương tác xuất hiện.
 2. Bản mẫu làm việc cài đặt một tập con các chức năng của phần mềm mong muốn
 3. Một chương trình đã có thực hiện một phần hay tất cả các chức năng mong muốn nhưng cần phải cải tiến thêm các tính năng khác tùy theo khả năng phát triển.



Mô hình chế thử

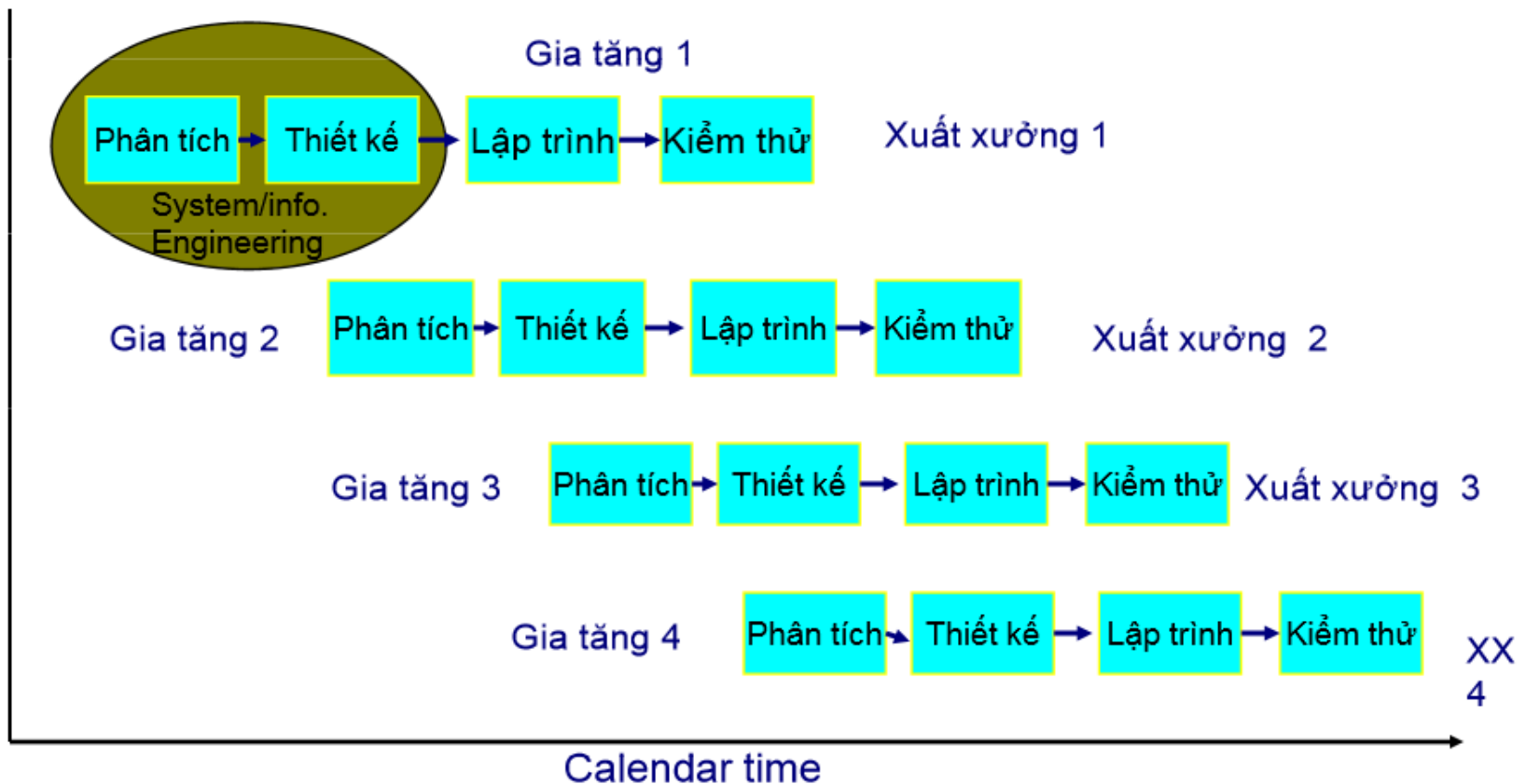
- Ví dụ một mô hình mô phỏng tương tác thi trắc nghiệm trên di động





Mô hình gia tăng

(The incremental model)





Ví dụ



- Trong ví dụ trên chúng ta phải làm việc gia tăng thêm các miếng ghép cho đến khi bức tranh ý tưởng được hoàn thành.



Mô hình gia tăng

- Là sự kết hợp giữa mô hình tuần tự và ý tưởng lặp lại của chế bản mẫu.
- Sản phẩm lỗi với những yêu cầu cơ bản nhất của hệ thống được phát triển trước. Các chức năng với những yêu cầu khác được phát triển thêm sau (gia tăng)
- Lặp lại quy trình để hoàn thiện dần



Mô hình gia tăng

- Mô hình này được đề xuất dựa trên ý tưởng thay vì phải xây dựng và chuyển giao hệ thống một lần thì sẽ chia thành nhiều giai đoạn tăng dần. mỗi giai đoạn là một phần kết quả chức năng được yêu cầu.
- Các yêu cầu của khách hàng được đánh thứ tự ưu tiên. Yêu cầu nào được đánh thứ tự ưu tiên càng cao thì càng ở trong những giai đoạn phát triển sớm và được kiểm tra kỹ lưỡng.



Mô hình gia tăng

- Sử dụng mô hình gia tăng khi nào?
 - Khi yêu cầu của dự án là rõ ràng, đầy đủ, và nắm rõ được các yêu cầu của dự án.
 - Khi sớm cần có một nguyên mẫu phần mềm để quảng bá, giới thiệu hoặc thử nghiệm.
 - Khi một công nghệ mới được áp dụng.
 - Tài nguyên và kỹ năng chuyên môn luôn sẵn sàng.
 - Khi có một tính năng hay các mục tiêu có nguy cơ lỗi cao.



Mô hình gia tăng

- Ưu điểm:
 - Có thể sớm tạo ra nguyên mẫu của sản phẩm
 - Độ linh hoạt cao hơn và khi thay đổi yêu cầu dự án thì chi phí sẽ ít hơn nhiều, vì những thay đổi thuộc về module nào thì module đó sẽ thay đổi mà các module khác không hề bị ảnh hưởng.
 - Làm cho việc test nhẹ nhàng hơn, những module đơn giản thì test cũng đơn giản, sớm kết thúc.
 - Giảm chi phí cho lần đầu giao sản phẩm.
 - Dễ dàng quản lý các rủi ro có thể phát sinh.



Mô hình gia tăng

- Nhược điểm:
 - Cần phải có những khả năng thiết kế tốt và phương pháp tốt.
 - Cần hiểu rõ được yêu cầu và biết cách phân chia nó ra như thế nào cho hợp lý.
 - Chi phí để phát triển theo phương pháp này là rất cao, cao hơn hẳn mô hình thác nước.

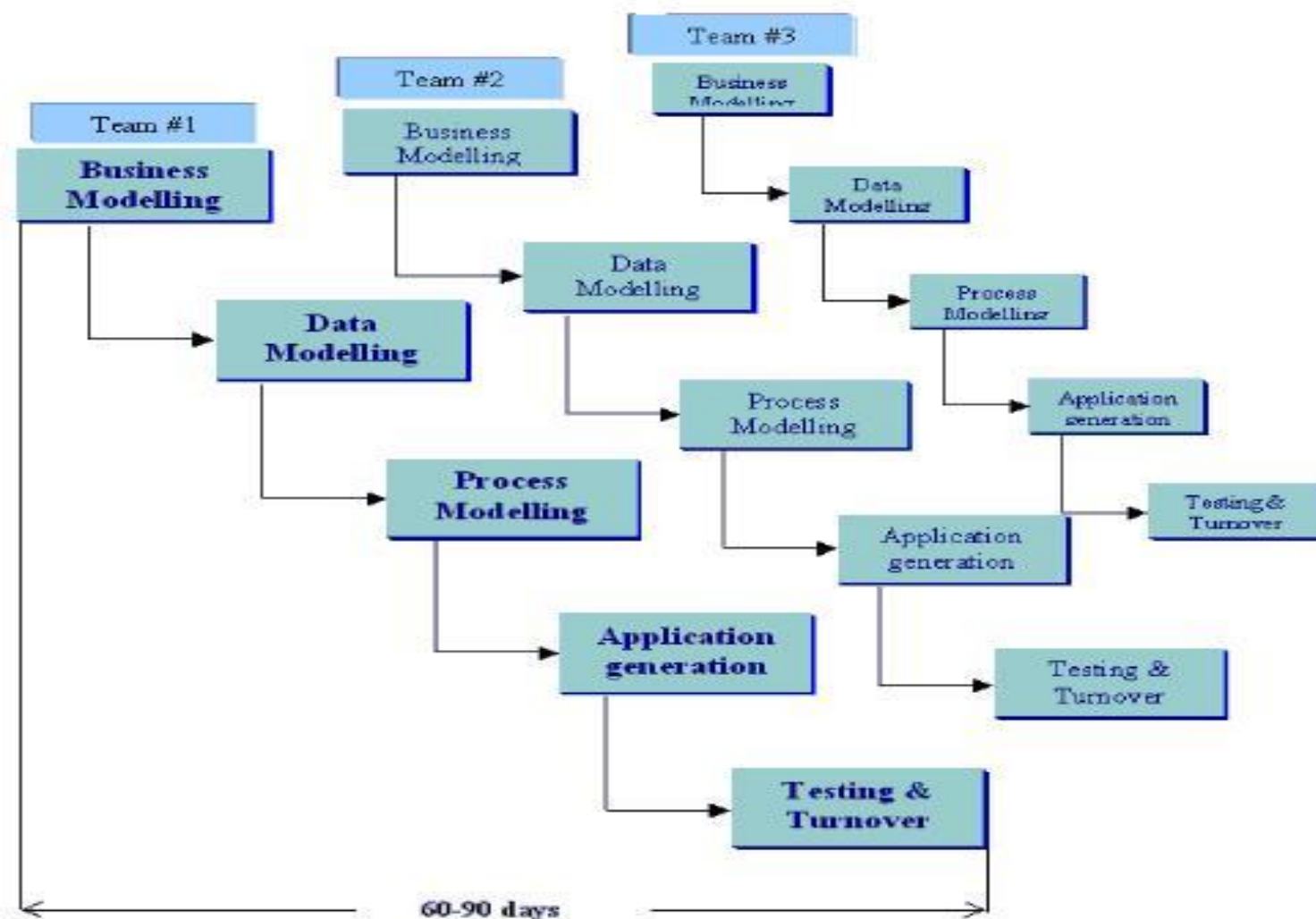


Mô hình RAD

- Là một dạng khác của mô hình tiến trình gia tăng(Incremental Process). Nó dựa trên việc tạo mẫu và phát triển lặp nhấn mạnh đến chu kỳ phát triển ngắn (60-90 ngày).
- Nó tập trung vào việc thu thập các yêu cầu của khách hàng, thử nghiệm sớm nguyên mẫu bằng cách sử dụng khái niệm lặp, tái sử dụng nguyên mẫu hiện có(thành phần), tích hợp liên tục và chuyển giao nhanh sản phẩm.



Mô hình RAD





Mô hình RAD

- Mỗi nguyên mẫu là một mô hình làm việc có chức năng tương đương với một thành phần của sản phẩm.
- Các môđun chức năng được phát triển song song như các nguyên mẫu, nghĩa là các nhóm nhỏ(team) hoạt động độc lập như các dự án nhỏ.



Mô hình RAD

- Mỗi nguyên mẫu đều được phát triển với chu trình phát triển lặp qua các pha: phân tích(Business Modeling), thiết kế(Data Modeling & Process Modeling), cài đặt (Application Generation), kiểm thử (Testing and turnover).
- Điều quan trọng nhất cho mô hình này thành công đó là nguyên mẫu được phát triển có thể tái sử dụng được.



Mô hình RAD

- Business modeling: Xác định các luồng thông tin giữa các chức năng khác nhau trong cùng một nguyên mẫu. Bao gồm:
 - Phân tích nghiệp vụ để tìm ra luồng thông tin chính cho các chức năng.
 - Làm thế nào để thu được luồng thông tin?
 - Làm thế nào và khi nào xử lý thông tin?
 - Những yếu tố nào thúc đẩy thành công luồng thông tin?



Mô hình RAD

- Data Modeling:
 - Luồng thông tin được phân tích ra trong pha trước sẽ được xem xét và phân tích để tìm ra các bộ đối tượng quan trọng cho các chức năng.
 - Xác định để tìm ra và định nghĩa các thuộc tính của tập dữ liệu.
 - Thiết lập và định nghĩa các mối quan hệ giữa các đối tượng dữ liệu trong mô hình nghiệp vụ.



Mô hình RAD

- Process Modeling:
 - Các đối tượng dữ liệu được tìm ra ở mô hình dữ liệu sẽ được xử lý chuyển đổi để thực hiện đúng theo luồng thông tin nghiệp vụ đã được xác định trong mô hình nghiệp vụ.
 - Các quá trình xử lý dữ liệu bao gồm: Thêm, xóa, truy xuất hoặc sửa đổi các đối tượng dữ liệu.
- Application Generation: Công đoạn cài đặt ra nguyên mẫu.



Mô hình RAD

- Testing and Turnover:
 - Thời gian kiểm thử tổng thể của mô hình RAD được giảm thiểu do đã được kiểm thử độc lập trong suốt mỗi lần lặp của mỗi nguyên mẫu.
 - Tuy nhiên luồng dữ liệu và giao diện giữa các nguyên mẫu thành phần cần phải được kiểm thử kỹ lưỡng khi kết hợp tạo ra sản phẩm hoàn chỉnh.
 - Vì các nguyên mẫu đều đã được kiểm thử nên nó làm giảm thiểu nguy cơ rủi ro của dự án.



Mô hình RAD

- So sánh với các mô hình truyền thống:
 - Điểm yếu của mô hình truyền thống là sự cứng nhắc với việc nhấn mạnh phân tích yêu cầu và thu thập dữ liệu trước khi đưa ra sản phẩm nguyên mẫu, vì vậy nó gây ra áp lực cho khách hàng khi phải ký kết hợp đồng trong khi họ chưa cảm thấy được sản phẩm.
 - Việc thay đổi theo yêu cầu của khách hàng trong MHTT là khá cứng nhắc hoặc có thể không khả thi. Trong khi đó mô hình RAD việc thay đổi rất linh hoạt nhờ việc tạo nguyên mẫu và sự độc lập môđun giữa các thành phần.



Mô hình RAD

- Khi nào nên áp dụng mô hình RAD:
 - Chỉ sử dụng với các sản phẩm phần mềm có tính môđun hóa để có thể chuyển giao theo cách gia tăng.
 - Nên sử dụng khi có sẵn nhân lực và các nhà thiết kế có trình độ cao, các nhân lực chuyên gia về lĩnh vực nghiệp vụ của sản phẩm.
 - Nên sử dụng khi có yêu cầu thay đổi của khách hàng trong suốt dự án và thời gian hoàn thành ngắn trong 2-3 tháng.



Mô hình xoắn ốc

- Là quy trình phát triển định hướng rủi ro cho các dự án phần mềm, mô hình xoắn ốc đưa ra cách áp dụng các yếu tố của một hoặc nhiều mô hình xử lý, chẳng hạn như mô hình thác nước, mô hình bản mẫu tiến hóa.
- Mô hình này lần đầu được Barry Boehm đưa ra. Mỗi cấp độ trong xoắn ốc liên quan đến việc lập kế hoạch, phân tích, phát hiện rủi ro, hoàn thiện hệ thống và tạo mẫu thêm.



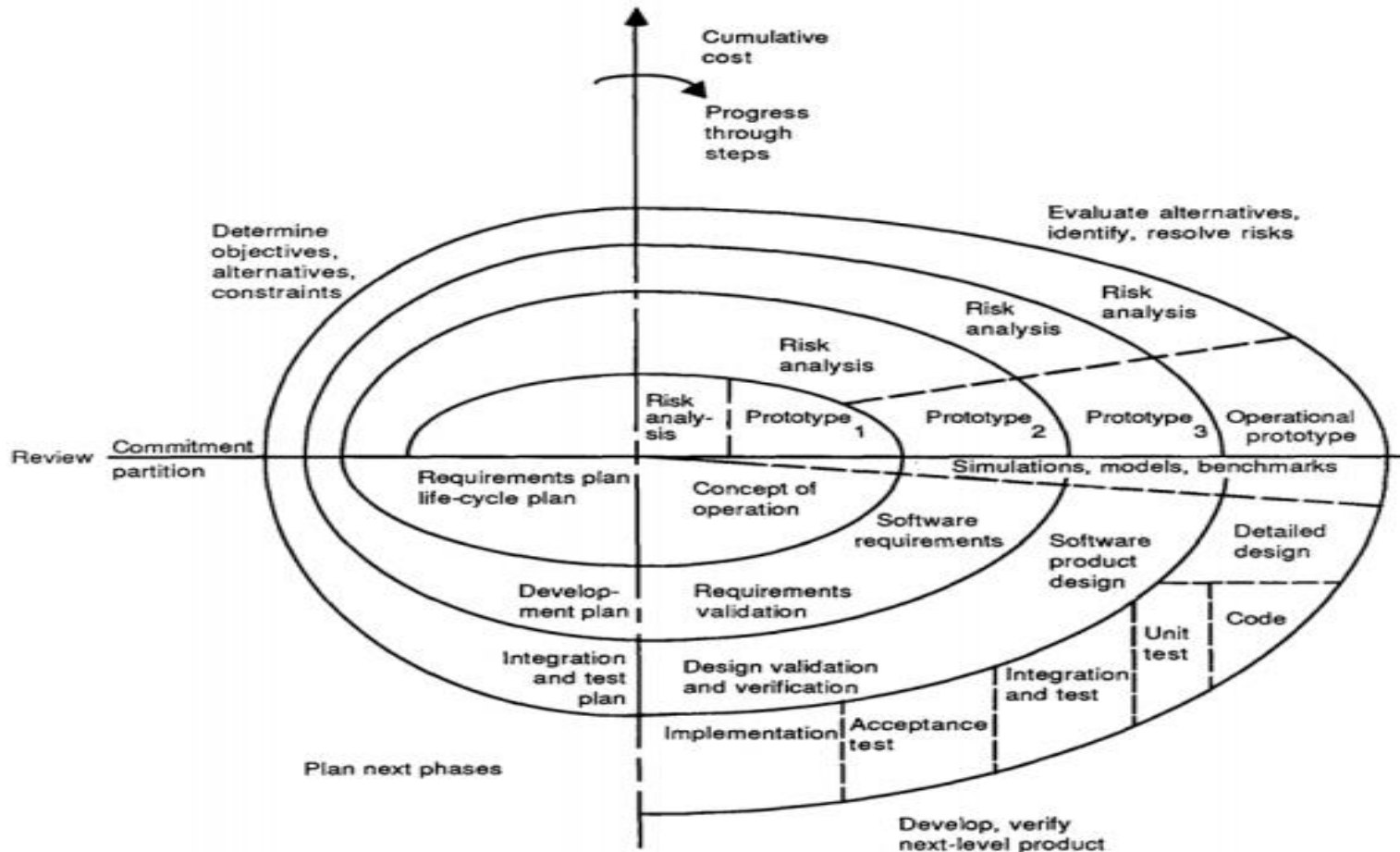
Mô hình xoắn ốc

- Về bản chất, nó mô tả sự phát triển của phần mềm qua các giai đoạn tiến hóa, mỗi giai đoạn được coi như một mô hình thác nước, bổ sung vào giai đoạn thực hiện PM mẫu (prototype) ngay sau khi xác định YC nhằm mục tiêu phát hiện nhanh các sai sót về yêu cầu.
- Trong mỗi vòng lặp của đường xoắn ốc, khách hàng đánh giá và đề nghị được sửa đổi. Thêm vào đó với mỗi vòng lặp, một phân tích rủi ro được thực hiện cho kết quả là tiếp tục dự án hay dừng lại.



Mô hình xoắn ốc

Mô hình xoắn ốc (spiral model)





Các hoạt động chính của mỗi bước lập trong quy trình

1. *Lập kế hoạch*: xác định mục tiêu, giải pháp và ràng buộc
2. *Phân tích rủi ro*: phân tích các phương án và xác định, giải quyết rủi ro.
3. *Kỹ nghệ*: phát triển sản phẩm “mức tiếp”
4. *Đánh giá của khách hàng*: khẳng định kết quả của kỹ nghệ.



Phân tích rủi ro

1. Hãy xem xét đến các điều kiện quan trọng nhất của các yếu tố liên quan từ đó xác định rủi ro và đặt độ ưu tiên cho các rủi ro.
2. Thiết lập các bản mẫu cho các rủi ro đã xác định và bắt đầu với cái có độ ưu tiên cao nhất.
3. Nếu một rủi ro giải quyết thành công, đánh giá kết quả và lập kế hoạch cho vòng tiếp theo
4. Nếu một rủi ro nào không giải quyết được, kết thúc dự án ngay lập tức



Spiral Model Template

- Phân tích những rủi ro và khả năng giải quyết thường là xây dựng bản mẫu.
- Để xác định rủi ro của mỗi giai đoạn trong mỗi xoắn ốc, Boehm sử dụng mẫu "Spiral Model Template".
- Được Barry Boehm lần đầu đưa ra trong bài báo năm 1968 với tựa đề "A Spiral Model of Software Development and Enhancement " (1)



Spiral Model Template

Template	Explanation	Example Phase
Objectives	The goals of the software project	Significantly improve software quality
Constraints	Limitations which the project must meet	Within three years Without large-scale capital investment Without radical change to company standards
Alternatives	Possible ways to achieve the objectives	Reuse existing certified software Introduce formal specification and verification Invest in testing and validation tools
Risks	Potential risks for this phase	No cost effective quality improvement possible Quality improvements may increase costs excessively New methods might cause existing staff to leave
Risk Resolution	Strategies for reducing the risks	Literature survey, Pilot project, Survey of potential reusable components, Assessment of available tool support, Staff training and motivation seminars
Results	Results of applying risk resolution strategies	Experience of formal methods is limited - very hard to quantify improvements Limited tool support available for company standard development system Reusable components available but little reuse tool support
Plans	Development plans for the next phase	Explore reuse option in more detail Develop prototype reuse support tools Explore component certification scheme
Commitment	Resources needed to achieve the plans	Fund further 18-month study phase



Kỹ nghệ

- Phát triển và triển khai
 - Dựa trên việc lập kế hoạch và phân tích rủi ro để từ đó phát triển hệ thống đồng thời phải kiểm tra lại. Giai đoạn này ta nên sử dụng mô hình thác nước để phát triển dự án.
- Lập kế hoạch cho pha tiếp theo
 - Chúng ta xem xét tiến độ và đánh giá thông qua các thông số đã đưa ra ở bước lập kế hoạch. Từ đó, tiếp tục triển khai giải quyết các vấn đề còn lại với qui trình được lặp lại tương tự.



Đặc điểm của mô hình

- Mô hình xoắn ốc cung cấp cách thức làm phần mềm bằng cách đưa ra các phiên bản:
 - Đây không phải là bổ sung thêm các thành phần mới như mô hình gia tăng.
 - Đây là sự tiến hóa: Cũng các đặc trưng ấy nhưng được làm mịn hơn, chi tiết hơn, cũng như nêu ra được các rủi ro mới cần giải quyết
 - Phiên bản sau cùng chính là phần mềm hoàn chỉnh có thể chuyển giao cho khách hàng sử dụng.



Những quan niệm sai lầm nguy hiểm

- Mô hình xoắn ốc đơn giản chỉ là một chuỗi sự phát triển của mô hình thác nước.
- Tất cả các hoạt động dự án theo một trình tự xoắn ốc đơn.
- Mọi hoạt động trong sơ đồ phải được thực hiện, và theo thứ tự hiển thị.



Ưu điểm của mô hình xoắn ốc

- Là mô hình hội tụ các tính năng tốt và khắc phục các yếu điểm của nhiều mô hình phát triển khác gặp phải.
- Giám sát dự án dễ dàng và hiệu quả.
- Rất phù hợp với dự án lớn cần phát triển nhiều phiên bản, các hệ thống có yêu cầu xác định chưa rõ ràng, có nguy cơ cao và giảm thiểu rủi ro, đối phó với những thay đổi trong quá trình thực hiện dự án.
- Dự đoán về thời hạn và chi phí sát với thực tế.



Nhược điểm của mô hình xoắn ốc

- Phân tích rủi ro khá tốn kém, chủ yếu áp dụng cho dự án lớn, có tiềm lực về tài chính
- Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn, phức tạp, cần có đội ngũ chuyên gia về phân tích rủi ro
- Chưa được áp dụng rộng rãi như mô hình thác nước, nguyên mẫu.



Mô hình xoắn ốc Win-Win

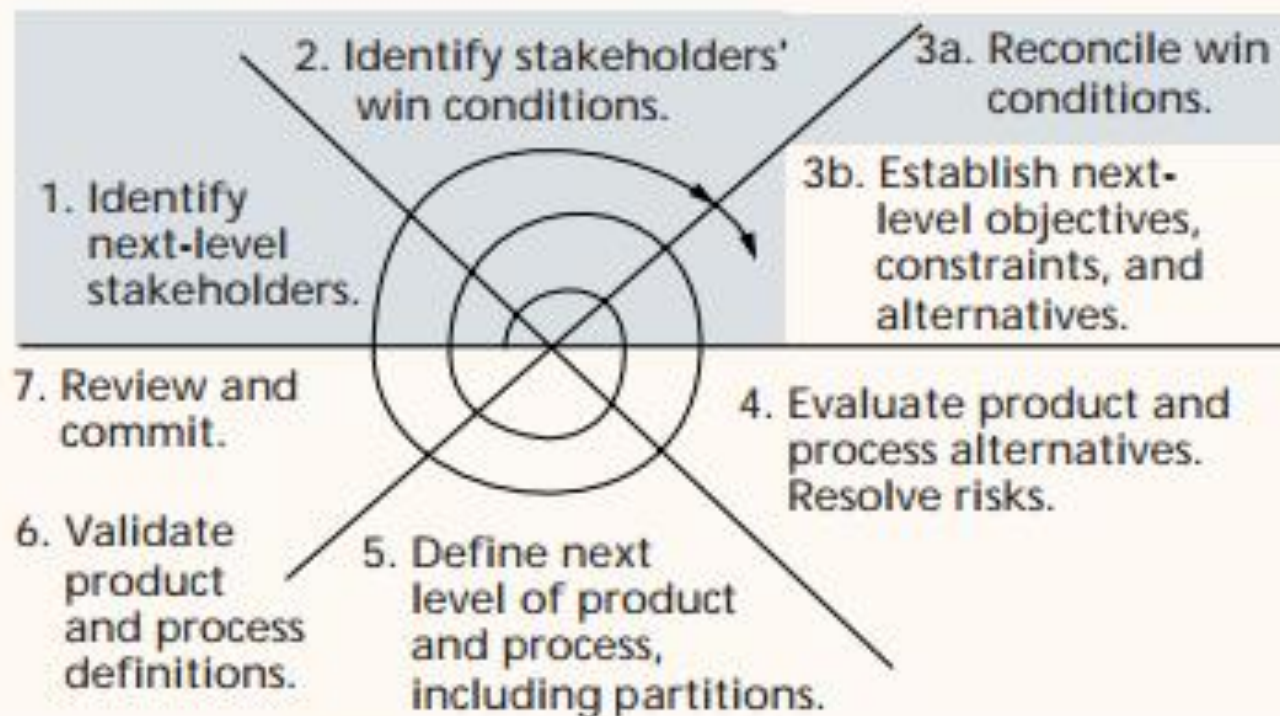


Figure A. How the WinWin spiral model differs from the original spiral model. The new model adds front-end activities (blue) that show where objectives, constraints, and alternatives come from. This lets users more clearly identify the rationale involved in negotiating win conditions for the product.



Mô hình xoắn ốc Win-Win

- Mô hình xoắn ốc Win-Win được đưa ra bởi Barry Boehm là sự mở rộng của mô hình xoắn ốc, nhằm thỏa hiệp giữa người phát triển và khách hàng, cả hai cùng thắng.
- Các giai đoạn trong mô hình này giống như mô hình xoắn ốc, chỉ khác ở chỗ thời điểm xác định yêu cầu, đội phát triển và khách hàng cùng thảo luận và đàm phán điều kiện thắng. (Là phần bổ sung màu xanh)



Phát triển Agile và mô hình 4GT

- Phát triển Agile là một phương pháp thực hiện các dự án phần mềm dựa trên các nguyên tắc:
 - Phát triển phân đoạn lặp (iterative)
 - Tăng trưởng (incremental)
 - Tiến hóa (evolutionary)
- Trong đó:
 - Khuyến khích sự thay đổi khi phát triển
 - Chuyển giao sản phẩm đến người dùng sao cho nhanh nhất.
 - Tập trung vào chất lượng



Phát triển Agile và mô hình 4GT

- Phương pháp phát triển phần mềm linh hoạt (Agile) khác biệt so với các phương pháp khác là:
 - Số lượng giấy tờ tài liệu ít hơn
 - Tập trung vào lập trình hơn
 - Khác biệt về nền tảng tư duy:
 - Thích ứng thay vì dự đoán
 - Hướng đến con người thay vì quy trình



Phát triển Agile và mô hình 4GT

- PPPTPMLH đề cao tính chủ động và sáng tạo của các cá nhân tham gia, đặc biệt là việc trao đổi thông tin giữa các thành viên.
- PPPTPMLH không khước từ sự tổ chức nhưng nó cố gắng
 - Cân bằng giữa sự tổ chức và sự linh hoạt
 - Cân bằng giữa việc không có quy trình và quy trình quá chi li, cứng nhắc.



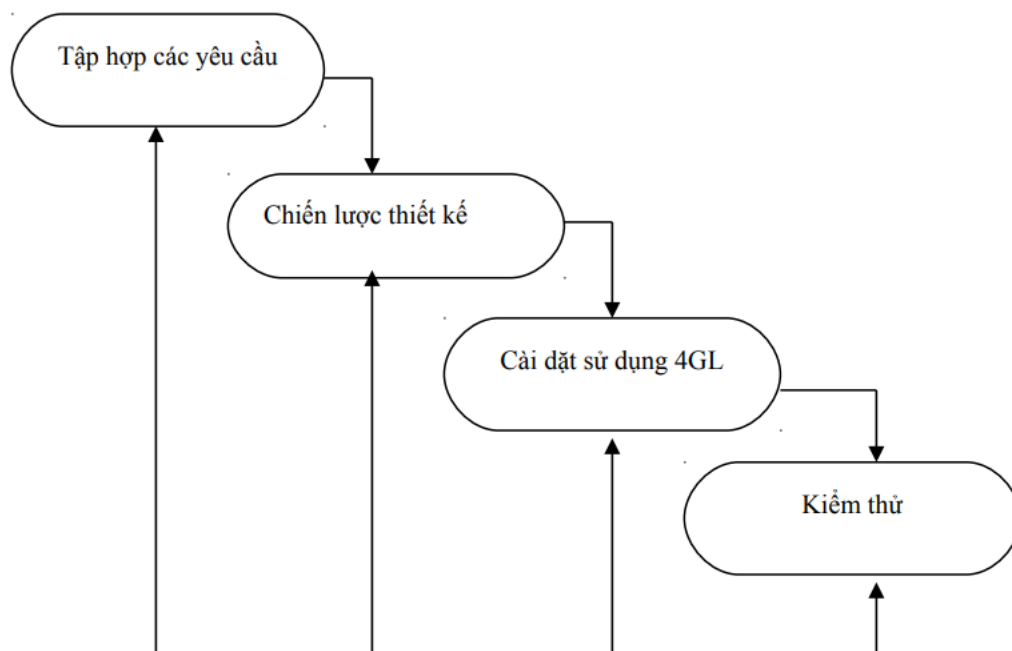
Phát triển Agile và mô hình 4GT

- Mô hình 4GT:
 - Là kỹ thuật dựa vào công cụ phần mềm, có thể đặc tả phần mềm ở mức khái niệm cao theo một cách thức định trước công cụ sẽ tự động sinh mã.
 - 4GT thích hợp cho ứng dụng vừa và nhỏ
 - 4GT tăng năng suất đáng kể
 - Một số ý kiến cho rằng:
 - Một số công cụ khó sử dụng
 - Mã sinh tự động tạo ra cồng kềnh
 - Việc bảo trì cho các hệ thống lớn là một vấn đề



Phát triển Agile và mô hình 4GT

- Mô hình 4GT:
 - Hiện nay dùng thành phần+4GT đang được ứng dụng mạnh nhằm giảm thời gian và chi phí cho dự án.





Nội dung



3. Các giai đoạn trong quy trình kỹ nghệ phần mềm

- Giai đoạn xác định
- Giai đoạn phát triển
- Giai đoạn bảo trì



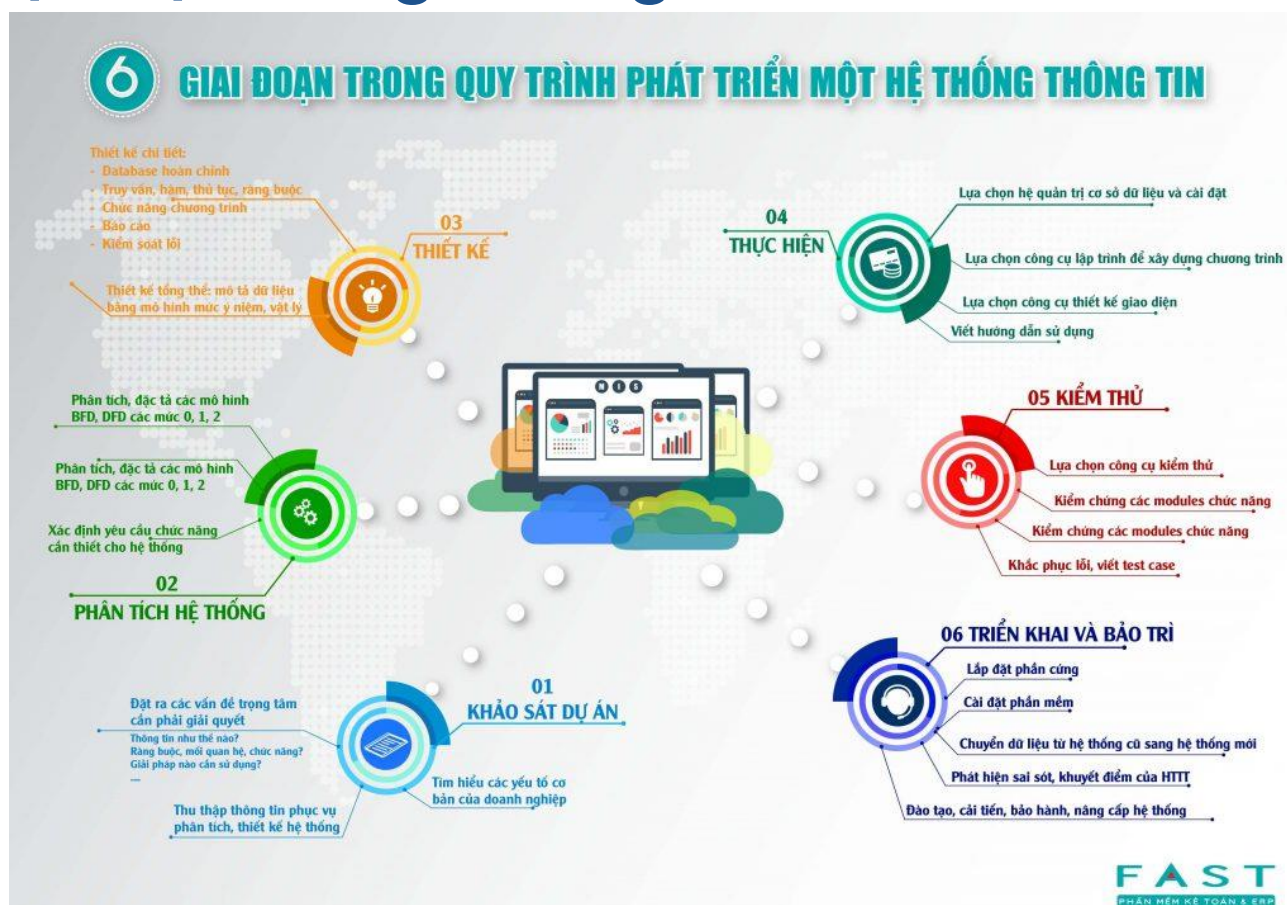
Hệ thống thông tin

- Hệ thống là một tập quy trình nghiệp vụ liên quan, kết hợp với nhau.
 - Hệ thống xử lý giao dịch
 - Hệ thống tự động hóa văn phòng
 - Hệ thống hỗ trợ quyết định
 - Hệ thống hỗ trợ điều hành quản lý
 - Hệ thống hỗ trợ quản lý, chia sẻ kiến thức, blog, forum ...
 - ...



Hệ thống thông tin

- Các giai đoạn trong quy trình phát triển một hệ thống thông tin





Giai đoạn xác định

- Giai đoạn này xác định làm cái gì?
- Bao gồm ba bước:
 - Phân tích ở mức hệ thống
 - Lập kế hoạch dự án phần mềm
 - Phân tích yêu cầu



Giai đoạn phát triển

- Giai đoạn này xác định làm như thế nào?
- Bao gồm ba bước:
 - Thiết kế phần mềm
 - Mã hóa
 - Kiểm thử phần mềm



Giai đoạn bảo trì

- Nhiệm vụ của bảo trì là đảm bảo cho phần mềm vận hành tốt.
- Trong giai đoạn bảo trì có thể phải:
 - Sửa đổi
 - Bổ sung



Thảo luận

