

## Chương 1

### Tổng quan về phát triển phần mềm

- 1.1 Máy tính số và công dụng
- 1.2 Chương trình máy tính & phần mềm
- 1.3 Đời sống phần mềm
- 1.4 Các mục tiêu của việc phát triển phần mềm
- 1.5 Các hoạt động chức năng trong phát triển phần mềm
- 1.6 Quy trình phát triển phần mềm
- 1.7 Vài thông tin về nắm bắt yêu cầu
- 1.8 Kết chương



## I. Máy tính số và công dụng

- ❑ **Các thiết bị cụ thể hóa** : Con người thông minh hơn các động vật khác nhiều. Ta đã chế tạo ngày càng nhiều công cụ, thiết bị để hỗ trợ mình trong hoạt động. Trong giai đoạn dài ban đầu, ta chế tạo theo hướng cụ thể hóa, nghĩa là mỗi công cụ, thiết bị chỉ thực hiện được 1 vài công việc cụ thể nào đó. Thí dụ, cây chổi để quét, radio để bắt và nghe đài audio... Các công cụ, thiết bị cụ thể hóa được chế tạo ngày càng tinh vi, phức tạp và thực hiện nhiều công việc hơn trước đây.
- ❑ Khuyết điểm của thiết bị cụ thể hóa là mỗi thiết bị chỉ giải quyết 1 công việc. Trong cuộc sống, ta cần thực hiện rất nhiều công việc, do đó ta phải chế tạo rất nhiều thiết bị và việc này không có dấu hiệu chấm dứt.



## I. Máy tính số và công dụng

- ❑ **Thiết bị tổng quát hóa - Máy tính số (digital computer)** : nó cũng là 1 thiết bị, nhưng thay vì chỉ thực hiện 1 vài chức năng cụ thể, sát với nhu cầu đòi thường của con người, nó có thể thực hiện 1 số hữu hạn các chức năng cơ bản (tập lệnh), mỗi lệnh rất sơ khai chưa giải quyết trực tiếp được nhu cầu đòi thường nào của con người, nhưng nếu chúng ta biết cách tổ chức, sắp xếp các lệnh với độ dài nhất định và thứ tự nhất định thì đoạn lệnh này sẽ giải quyết được công việc có nghĩa mà con người cần. Cơ chế thực hiện các lệnh là tự động, bắt đầu từ lệnh được chỉ định nào đó rồi tuần tự từng lệnh kế tiếp cho đến lệnh cuối cùng. Danh sách các lệnh được thực hiện này được gọi là chương trình.



## I. Máy tính số và công dụng

- ❑ Bất kỳ công việc (bài toán) ngoài đời nào cũng có thể được chia thành trình tự nhiều công việc nhỏ hơn. Trình tự các công việc nhỏ này được gọi là giải thuật giải quyết công việc ngoài đời. Mỗi công việc nhỏ hơn cũng có thể được chia nhỏ hơn nữa nếu nó còn phức tạp,... công việc ngoài đời có thể được miêu tả bằng 1 trình tự các lệnh máy (chương trình ngôn ngữ máy).
- ❑ Ưu điểm của máy tính số là ta chỉ cần chế tạo 1 lần, rồi theo thời gian ta nhờ nó chạy những chương trình (phần mềm) khác nhau để thực hiện các bài toán khác nhau.



## I. Máy tính số và công dụng

- ❑ Vấn đề mấu chốt của việc dùng máy tính giải quyết công việc ngoài đời là lập trình (được hiểu nôm na là qui trình xác định trình tự đúng các lệnh máy để thực hiện công việc). Cho đến nay, lập trình là công việc của con người (với sự trợ giúp ngày càng nhiều của máy tính).
- ❑ Với công nghệ phần cứng hiện nay, ta chỉ có thể chế tạo các máy tính mà tập lệnh máy rất sơ khai, mỗi lệnh máy chỉ có thể thực hiện 1 công việc rất nhỏ và đơn giản  $\Rightarrow$  công việc ngoài đời thường tương đương với trình tự rất lớn (hàng triệu) các lệnh máy. Lập trình bằng ngôn ngữ máy rất phức tạp, tốn nhiều thời gian, công sức, kết quả rất khó bảo trì, phát triển.



## I. Máy tính số và công dụng

- ❑ Ta muốn có máy luận lý với tập lệnh cao cấp và gần gũi hơn với con người (được đặc tả bởi ngôn ngữ lập trình). Ta hiện thực máy này bằng 1 máy có sẵn (lúc đầu là máy vật lý) + 1 chương trình dịch.
- ❑ Gọi ngôn ngữ máy vật lý là N0. Trình biên dịch ngôn ngữ N1 sang ngôn ngữ N0 sẽ nhận đầu vào là chương trình được viết bằng ngôn ngữ N1, phân tích từng lệnh N1 rồi chuyển thành danh sách các lệnh ngôn ngữ N0 có chức năng tương đương. Để viết chương trình dịch từ ngôn ngữ N1 sang N0 dễ dàng, độ phức tạp của từng lệnh ngôn ngữ N1 không quá cao so với từng lệnh ngôn ngữ N0.
- ❑ Có 2 loại chương trình dịch : trình biên dịch (compiler) và trình thông dịch (interpreter).



## I. Máy tính số và công dụng

- ❑ Sau khi có máy luận lý hiểu được ngôn ngữ luận lý N1, ta có thể định nghĩa và hiện thực máy luận lý N2 theo cách trên và tiếp tục đến khi ta có 1 máy luận lý hiểu được ngôn ngữ Nm rất gần gũi với con người, dễ dàng miêu tả giải thuật của bài toán cần giải quyết...
- ❑ Nhưng qui trình trên chưa có điểm dừng, với yêu cầu ngày càng cao và kiến thức ngày càng nhiều, người ta tiếp tục định nghĩa những ngôn ngữ mới với tập lệnh ngày càng gần gũi hơn với con người để miêu tả giải thuật càng dễ dàng, gọn nhẹ và trong sáng hơn.
- ❑ Hiện nay, ta thường viết phần mềm bằng các ngôn ngữ cấp cao như Java, C#...



## II. Chương trình máy tính – Phần mềm

- ❑ Chương trình máy tính là một giải thuật cụ thể để giải quyết công việc ngoài đời mà mỗi công việc trong giải thuật được miêu tả bằng 1 lệnh máy cụ thể.
- ❑ Chương trình máy tính có thể là chương trình ngôn ngữ máy, nó có thể chạy trực tiếp bởi CPU phần cứng của máy.
- ❑ Chương trình máy tính cũng có thể là chương trình ngôn ngữ cấp cao như C, C++, C#, Java,... Trong trường hợp này ta phải dùng chương trình dịch nó ra chương trình ngôn ngữ máy có hành vi tương đương trước khi có thể nhờ CPU máy tính chạy trực tiếp nó.
- ❑ Đối với các ứng dụng lớn và phức tạp, ta nên chọn ngôn ngữ cấp cao nhất để xây dựng nó.
- ❑ Hiện ta đã xây dựng và dùng rất nhiều chương trình trên nhiều lĩnh vực của đời sống xã hội.



## II. Phần mềm – Phân loại

- Tùy vào góc nhìn mà ta phân loại phần mềm như thế nào. Thí dụ, nếu dựa vào cách thức hoạt động của phần mềm đối với người dùng nó thì ta có thể phân phần mềm ra 2 loại sau :
  - Phần mềm đại chúng : được viết và bán cho bất kỳ khách hàng nào muốn dùng chúng. Thí dụ như phần mềm đồ họa PhotoShop, tool quản lý dự án, phần mềm CAD...
  - Phần mềm cá nhân hóa : được viết cho khách hàng đặc biệt, riêng lẻ theo yêu cầu riêng của họ. Thí dụ hệ thống điều khiển nhúng, phần mềm điều khiển không lưu, phần mềm kiểm soát giao thông.



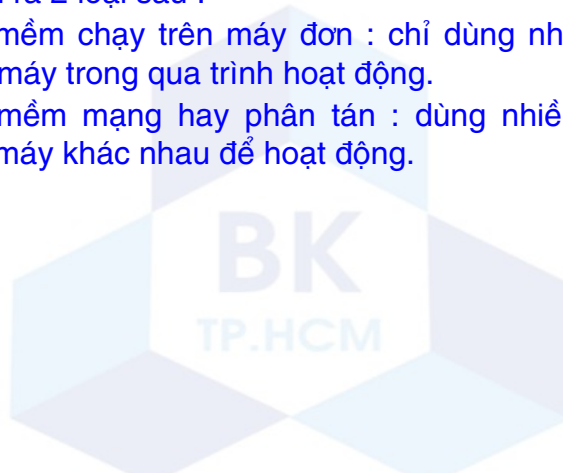
## II. Phần mềm – Phân loại

- Nếu dựa vào khả năng thay đổi hành vi của phần mềm đối với người dùng nó thì ta có thể phân phần mềm ra 2 loại sau :
  - Phần mềm cụ thể hóa : luôn cung cấp các tính năng và hành vi cố định, cho dù được chạy bởi ai và lần thứ mấy.
  - Phần mềm tổng quát hóa : có thể thay đổi tính năng và hành vi theo thời gian và theo yêu cầu từng người dùng. Thí dụ MS Word có thể là phần mềm soạn thảo văn bản, nhưng user có thể cá nhân hóa theo thời gian để biến nó thành ứng dụng có chức năng bất kỳ.



## II. Phần mềm – Phân loại

- ❑ Nếu dựa vào tài nguyên mà phần mềm dùng thì ta có thể phân phần mềm ra 2 loại sau :
  - Phần mềm chạy trên máy đơn : chỉ dùng những tài nguyên trên 1 máy trong quá trình hoạt động.
  - Phần mềm mạng hay phân tán : dùng nhiều tài nguyên ở nhiều máy khác nhau để hoạt động.



## II. Phần mềm – Phân loại

- ❑ Nếu dựa vào lĩnh vực xử lý thì ta có thể có những loại phần mềm sau :
  - **Hệ điều hành** : là phần mềm quản lý các tài nguyên cấp thấp (thường là phần cứng), che dấu các tính chất vật lý của chúng (thường rất khó hiểu và sử dụng), rồi cung cấp lại một interface sử dụng chúng với các lợi điểm như an toàn, tin cậy, thân thiện, hiệu quả và nhất là độc lập với tính chất vật lý của tài nguyên được sử dụng. Người ta còn gọi HĐH là máy ảo (máy luận lý)..
  - **Chương trình dịch** : sẽ dịch các file mã nguồn sang dạng mã máy tương đương (thường được link lại thành file khả thi - \*.exe). Mỗi lần chạy ứng dụng, ta chỉ kích hoạt file khả thi.



## II. Phần mềm – Phân loại

- **Ứng dụng văn phòng** : Cho phép người dùng thực hiện 1 số chức năng thông thường liên quan đến văn phòng. Microsoft Office là ứng dụng văn phòng được sử dụng phổ biến nhất. Open Office là ứng dụng văn phòng mã nguồn mở nhưng yếu hơn và thiếu ổn định hơn).
- **Ứng dụng nghiệp vụ & Database server** : Ứng dụng nghiệp vụ thực hiện các yêu cầu nghiệp vụ trong các cơ quan, đơn vị như quản lý nhân viên, quản lý tài sản, quản lý điểm, quản lý bệnh nhân và bệnh án,... Trong hầu hết các ứng dụng nghiệp vụ, dữ liệu cần lưu trữ và xử lý là rất lớn. Vấn đề lưu trữ và quản lý những dữ liệu lớn sao cho nhất quán, an toàn tin cậy,... đòi hỏi nhiều kiến thức chuyên sâu và nhiều thời gian công sức hiện thực. Database server giải phóng người lập trình khỏi các công việc liên quan đến database.



## II. Phần mềm – Phân loại

- **Biên tập & chơi multimedia** : Multimedia là dữ liệu đa phương tiện như văn bản, hình ảnh, âm thanh, film,... Dữ liệu đa phương tiện giúp người dùng thích thú hơn khi nghiên cứu về 1 vấn đề nào đó. Các ứng dụng biên tập dữ liệu đa phương tiện cho phép ta xây dựng, thêm/bớt/hiệu chỉnh thông tin và file đa phương tiện tương ứng. Thí dụ trình Photoshop cho ta xử lý ảnh tĩnh, SoundGold cho phép ta xử lý âm thanh, Photo Premiere cho ta biên tập film... Các ứng dụng chơi multimedia cho phép người dùng tham khảo file multimedia đã có. Thí dụ trình Window Multimedia Player của Microsoft cho ta chơi hầu hết các định dạng file multimedia khác nhau từ ảnh tĩnh, âm thanh hay film. File multimedia cần chơi có thể nằm trên máy đơn hay trên 1 server multimedia nào đó trong mạng Internet



## II. Phần mềm – Phân loại

- **Game** : Game (ứng dụng trò chơi) là những ứng dụng dễ lôi cuốn người dùng nhất. Có 2 thể loại game phổ biến : game hành động và game trí tuệ.
- **Các ứng dụng trên mạng Internet** : Thường ứng dụng mạng dùng mô hình hoạt động client/server, mỗi module sẽ đóng vai trò hoặc server, hoặc client. Module server sẽ quản lý các tài nguyên liên quan trên máy mình đang chạy và cung cấp dịch vụ truy xuất các tài nguyên này cho các module ở các máy khác. Module client sẽ chạy trên máy người dùng, cung cấp giao tiếp sử dụng thân thiện, dễ dàng, an toàn,... Các module server/client của 1 ứng dụng mạng thường tuân thủ 1 giao thức xác định nào đó. Giao thức (protocol) là tập các thông báo request/reply cùng định dạng cụ thể của từng thông báo mà client/server sẽ gửi/nhận cho nhau.



## II. Phần mềm – Phân loại

- **Các ứng dụng CAD (Computer-Aided Design)** : Giúp người thiết kế xây dựng/hiệu chỉnh bản thiết kế trực tiếp trên máy tính với sự giám sát/kiểm soát/trợ giúp ngày càng nhiều của phần mềm, nhờ đó ta có nhanh được bản thiết kế đúng đắn, khoa học, bài bản,... Bản vẽ có thể được phân tích tự động để xác định một số thông tin cần thiết. CAD được dùng để thiết kế chi tiết cơ khí, mạch điện tử, tàu thủy, xe hơi, nhà cửa, quần áo,§ Một số phần mềm tiêu biểu: OrCAD, AutoCAD,§





## II. Phần mềm – Phân loại

- **Ứng dụng CAM (Computer-Aided Manufacture)** : Phần mềm dùng máy tính hỗ trợ quá trình sản xuất và chế tạo linh kiện/thiết bị. Một số dạng CAM như Robot công nghiệp, CNC, CAD/CAM, hệ thống điều khiển... Robot công nghiệp thay thế con người trong việc thực hiện 1 công việc cụ thể xác định. Robot có 3 thành phần chính : tay máy, bộ điều khiển và bộ cung cấp năng lượng. Robot có khả năng cảm nhận môi trường xung quanh (nhờ các sensor) và đáp ứng kịp thời với sự thay đổi trạng thái của môi trường. CNC (Computer numerical control) được sử dụng rộng rãi trong các thiết bị máy, nó điều khiển hoạt động của máy dựa trên chương trình viết sẵn. Mạch vi xử lý sẽ thu thập các thông tin liên quan (thông qua các mạch D/A), dựa trên thông tin thu thập được, nó điều khiển thiết bị máy thay đổi hành vi theo sự tác động của bên ngoài.



## II. Phần mềm – Phân loại

- **Ứng dụng nhúng (embedded applications)** : Là các ứng dụng được cài đặt vào 1 thiết bị cụ thể có trang bị máy tính số với các tài nguyên hạn chế. Nhiệm vụ của ứng dụng nhúng là điều khiển quá trình hoạt động của thiết bị. Thí dụ chương trình điều khiển máy in laser, máy vẽ, máy photocopy, máy giặt,...

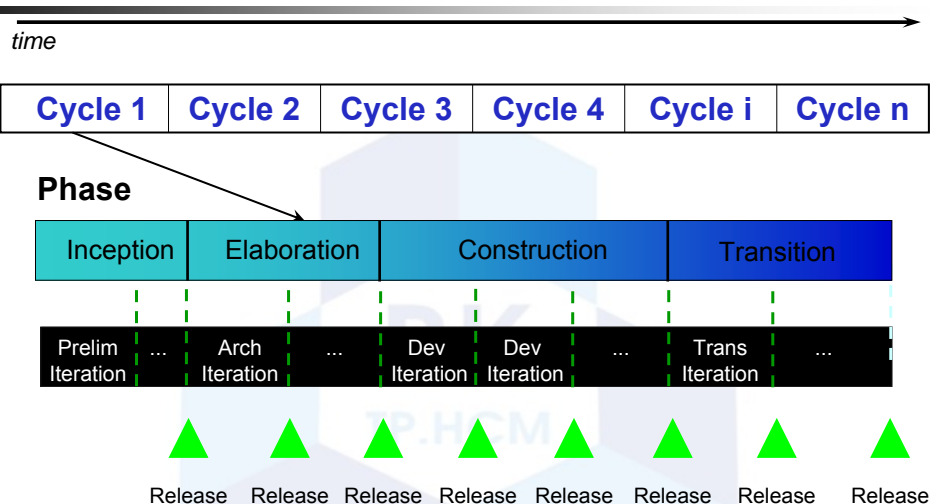


## II. Phần mềm – Phân loại

- **Mô hình hóa & mô phỏng** : Mô hình hóa được dùng để xem xét, phân tích hoặc lên kế hoạch cho một công việc phức tạp. Mô hình hóa thường dùng các công thức toán học để mô tả một công việc hay một quá trình phức tạp. Ví dụ :  $s = 0.5 at^2$  với  $s$  là quãng đường đi được,  $a$  là gia tốc và  $t$  là thời gian. Mô phỏng dùng máy tính để mô hình hóa và xem xét hệ thống : Kiểm tra, An toàn, Dự đoán, Nhanh chóng và linh hoạt. Thường phần mềm mô hình hóa & mô phỏng được sử dụng nhiều trong các lĩnh vực Giao thông, Xã hội học, Dự đoán thời tiết, Thủy lực học, Sinh học, Giáo dục học, Mô hình 3 chiều.



## III. Đời sống phần mềm



### III. Đời sống phần mềm

- ❑ Đời sống phần mềm (software life) : từ lúc xuất hiện lần đầu đến khi tác giả không hỗ trợ nữa (có thể vẫn còn dùng bởi nhiều người).
- ❑ Chu kỳ (Cycle) : đơn vị đo lường đời sống, bắt đầu khi có yêu cầu (ban đầu hay mới) cần giải quyết đến khi có version mới giải quyết tốt yêu cầu đó.
- ❑ Công đoạn (Phase) : hoạt động chức năng nhỏ cần thực hiện trong từng chu kỳ, thí dụ nằm bắt yêu cầu (Inception), phân tích và thiết kế (Elaboration), hiện thực và kiểm thử (Construction), chuyển giao (transition).
- ❑ Bước lặp (Iteration) : mỗi hoạt động, dù ngắn hay dài, dù nhỏ hay lớn, đều có thể phải lặp nhiều lần theo cơ chế tăng tiến để đạt đến mục tiêu đề ra.



### IV. Các mục tiêu của việc phát triển phần mềm

- ❑ Hiện nay, phần mềm thường phải có nhiều tính năng hữu dụng và dễ dàng sử dụng. Do đó độ phức tạp thường cao, hơn nữa phần mềm sẽ được dùng lâu dài qua nhiều thế hệ người dùng, trong quá trình dùng có thể phần mềm cần được nâng cấp hiệu chỉnh theo yêu cầu mới phát sinh.
- ❑ Ta dùng thuật ngữ "phát triển phần mềm" (software development) để chỉ tất cả các hoạt động giúp phần mềm tồn tại và phát triển theo thời gian.



## IV. Các mục tiêu của việc phát triển phần mềm

- ❑ Chi phí "phát triển phần mềm" :
  - Thường rất lớn và chiếm phần rất lớn trong chi phí xây dựng và duy trì hệ thống tính toán, cao hơn nhiều so với chi phí trang bị phần cứng.
  - Chi phí duy trì tính hữu dụng của phần mềm theo thời gian sử dụng thường rất cao so với chi phí xây dựng phiên bản đầu tiên của phần mềm.



## IV. Các mục tiêu của việc phát triển phần mềm

- ❑ Phát triển phần mềm cần đạt 2 mục tiêu lớn sau đây :
  - **Mục tiêu trước mắt:**
    - Mục tiêu về chức năng : Phần mềm phải thực hiện đúng và đủ các chức năng đề ra (tính đúng đắn và tin cậy của phần mềm). Đây là yêu cầu về chức năng.
    - Mục tiêu phi chức năng : Mỗi chức năng phải thực hiện 1 cách hiệu quả, tin cậy trong thời gian qui định... Tốn chi phí thấp nhất, thời gian nhanh nhất để có phần mềm chạy được, dùng được...



## IV. Các mục tiêu của việc phát triển phần mềm

- Phát triển phần mềm cần đạt 2 mục tiêu lớn sau đây :

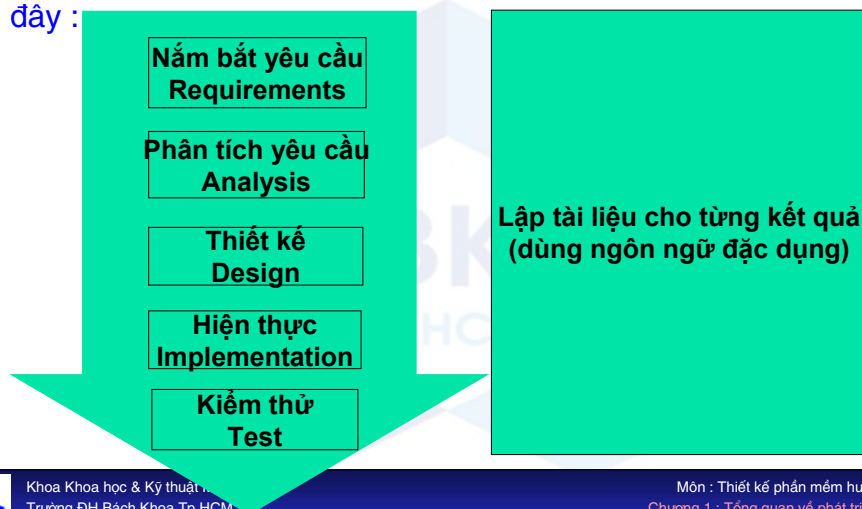
- **Mục tiêu lâu dài :**

- Không cần thay đổi theo thời gian và theo yêu cầu mới phát sinh (để hạn chế chi phí nâng cấp, hiệu chỉnh). Nếu trường hợp cần hiệu chỉnh, nâng cấp thì chỉ cần thay đổi ít nhất -> tối thiểu chi phí hiệu chỉnh, nâng cấp.
- Nếu cần thay đổi, nâng cấp thì phần mềm hiện hành phải hỗ trợ tối đa cho việc thay đổi, nâng cấp này -> tối thiểu chi phí hiệu chỉnh, nâng cấp.



## V. Các hoạt động chức năng $\subset$ phát triển phần mềm

- Việc phát triển phần mềm được tiến hành theo nhiều chu kỳ, trong mỗi chu kỳ ta phải thực hiện các hoạt động chức năng sau đây :



## V. Các hoạt động chức năng $\subset$ phát triển phần mềm

- **Nắm bắt yêu cầu (Requirements)** : Nhiệm vụ của workflow này là xác định chính xác, rõ ràng và đầy đủ các thông tin sau liên quan đến chương trình :
  - các chức năng của chương trình cần đáp ứng, mỗi chức năng cần xác định rõ phục vụ cho ai, họ cung cấp thông tin gì ở dạng nào cho phần mềm, phần mềm xuất kết quả gì và ở dạng nào?
  - chương trình sẽ tương tác với các thành phần nào : loại người nào, phần mềm nào, thiết bị nào,...



## V. Các hoạt động chức năng $\subset$ phát triển phần mềm

- **Phân tích yêu cầu (Analysis)** : Nhiệm vụ của workflow này là phát họa sơ lược cách giải quyết từng chức năng của chương trình. Thường ta dựa vào phương pháp phân tích từ trên xuống :
  - Nội dung của phương pháp này là xét xem, muốn giải quyết vấn đề nào đó thì cần phải làm những công việc nhỏ hơn nào. Mỗi công việc nhỏ hơn tìm được lại được phân thành những công việc nhỏ hơn nữa, cứ như vậy cho đến khi những công việc phải làm là những công việc thật đơn giản, có thể thực hiện dễ dàng.
  - Thí dụ việc học lấy bằng kỹ sư CNTT khoa CNTT ĐHBK TP.HCM có thể bao gồm 9 công việc nhỏ hơn là học từng học kỳ từ 1 tới 9, học học kỳ i là học n môn học của học kỳ đó, học 1 môn học là học m chương của môn đó,...



## V. Các hoạt động chức năng $\subset$ phát triển phần mềm

- ❑ Thiết kế (Design) : Nhiệm vụ của workflow này là chi tiết hóa cách giải quyết từng chức năng của chương trình đến mức độ dễ dàng viết code nhất có thể có. Nếu 1 người làm, ta lập thiết kế từng chức năng theo 1 thứ tự xác định.
- ❑ Hiện thực (Implementation) : Nhiệm vụ của workflow này là dịch bản thiết kế chi tiết thành mã nguồn của ngôn ngữ lập trình xác định, từ đó dịch ra mã máy để tạo thành chương trình khả thi có thể chạy trên máy tính. Nếu 1 người làm, ta lập hiện thực từng chức năng thiết kế (hay từng phần nhỏ của chức năng) theo 1 thứ tự xác định.
- ❑ Kiểm thử (Testing) : Nhiệm vụ của workflow này kiểm tra và thử nghiệm chương trình thực thi xem nó có lỗi không, nếu có thì lỗi cụ thể nằm ở lệnh nào, tại sao bị lỗi và sửa lỗi. Nếu 1 người làm, ta lập kiểm thử từng hàm chức năng theo 1 thứ tự xác định.



## V. Các hoạt động chức năng $\subset$ phát triển phần mềm

- ❑ Lập tài liệu cho từng hoạt động chức năng : Thực hiện mỗi workflow sẽ tạo ra nhiều kết quả, ta phải quản lý, bảo trì các kết quả này theo thời gian nhằm phục vụ cho việc nghiên cứu, hiệu chỉnh, nâng cấp phần mềm sau này. Một trong các việc quản lý, bảo trì các kết quả tạo được là lập tài liệu. Ta phải dùng 1 ngôn ngữ thích hợp để lập tài liệu cho các kết quả sao cho việc quản lý, bảo trì, chuyển giao phần mềm được dễ dàng, tin cậy và hiệu quả... Hiện nay, ngôn ngữ mô hình UML (Unified Modeling Language) được sử dụng rất phổ biến để đặc tả, quản lý các tài liệu trong quá trình phát triển phần mềm.



## VI. Quy trình phát triển phần mềm

- ❑ Nói nôm na là trình tự các công việc mà ta phải tuân thủ thực hiện để xây dựng và duy trì phần mềm hữu dụng theo thời gian.
- ❑ Có nhiều quy trình phát triển phần mềm khác nhau đã được đề nghị và sử dụng. Mỗi quy trình có những ưu, nhược điểm riêng và thích hợp trong từng loại phần mềm cụ thể.
- ❑ Cho dù dùng quy trình phát triển phần mềm nào thì cũng phải thực hiện các hoạt động chức năng đã được giới thiệu ở các slide trước, chỉ có đều cách thức và thời điểm thực hiện chúng như thế nào mà thôi. Các hoạt động chức năng phải làm là nắm bắt yêu cầu phần mềm, phân tích yêu cầu, thiết kế (kiến trúc và chi tiết), hiện thực, kiểm thử và chuyển giao cho user.



## VI. Quy trình phát triển phần mềm

- ❑ Quy trình RUP (Rational Unified Process) : quy trình phát triển phần mềm hợp nhất của hãng Rational, do hãng Rational khởi xướng và duy trì. Nó dựa trên việc dùng và kết hợp các đối tượng để cấu thành phần mềm. Nó có nhiều ưu điểm so với các quy trình khác, do đó được dùng chủ yếu hiện nay trong các công ty phát triển phần mềm. Trong các chương còn lại, ta sẽ tập trung giới thiệu các thông tin cơ bản và chính yếu của quy trình phát triển này.





## VI. Quy trình phát triển phần mềm

- ❑ Thuật ngữ "Công nghệ phần mềm" (Software engineering) để nói về các lý thuyết, phương pháp, qui trình và tiện ích được dùng trong việc phát triển phần mềm chuyên nghiệp hầu tối thiểu chi phí, nâng cao tính đúng đắn, bền vững của phần mềm.
- ❑ Trong các nội dung của công nghệ phần mềm thì qui trình phát triển phần mềm là chủ yếu nhất.
- ❑ Trong các qui trình phát triển được dùng thì hiện tại qui trình RUP được dùng phổ biến nhất. Qui trình này dựa trên nền tảng hướng đối tượng.
- ❑ Ngôn ngữ UML sẽ được dùng để đặc tả hầu hết các tài liệu tạo ra trong qui trình phát triển RUP.



## VI. Quy trình phát triển phần mềm RUP

- ❑ Quy trình phát triển là gì ? Nói theo kiểu thực dụng thì qui trình phát triển gồm tất cả tài liệu miêu tả những ai (who) tham gia, mỗi người phải làm gì (what), mỗi công việc sẽ được làm khi nào (when), khi làm thì làm theo cách nào (how) để đạt miêu tiêu đề ra.

Các yêu cầu  
ban đầu hay  
yêu cầu mới về  
phần mềm

Quy trình phát triển  
phần mềm

Hệ thống phần  
mềm ban đầu  
hay hệ thống kế  
tiếp

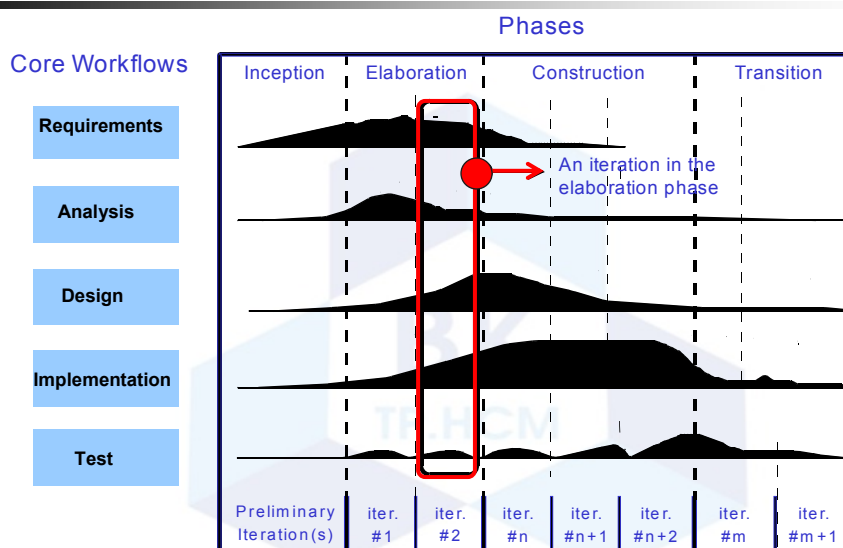


## VI. Quy trình phát triển phần mềm RUP

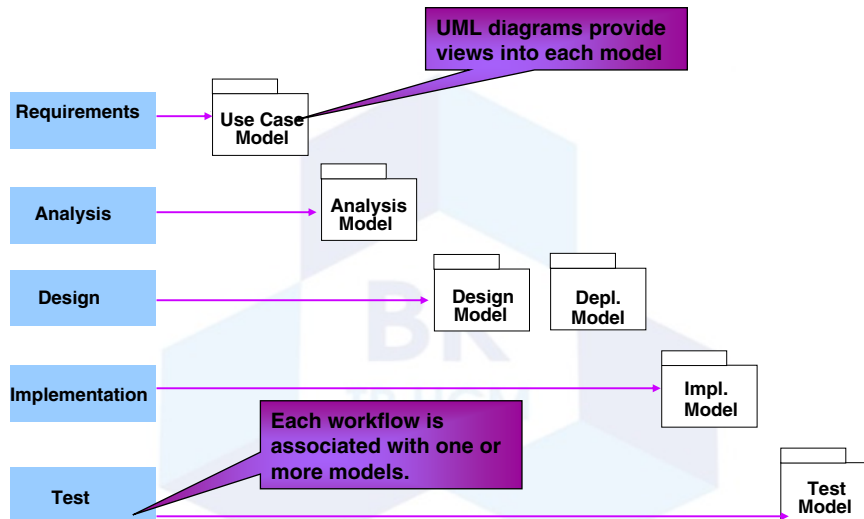
- ❑ Artifacts : miêu tả mọi kết quả được tạo ra sau 1 hoạt động chức năng hay sau 1 bước lặp nào đó. Ta phải duy trì artifacts theo thời gian. Artifacts thường được miêu tả dưới nhiều định dạng hy ngôn ngữ khác nhau, trong đó ngôn ngữ UML được dùng chủ yếu nhất. Tùy theo mức độ ta có Vision, Baseline Architecture, initial capability, product release (version).
- ❑ Workers : tất cả phần tử tham gia phát triển phần mềm, cho dù họ là ai. Ở đây ta quan tâm chủ yếu đến vai trò của worker, chứ không quan tâm đến người cụ thể và số lượng người cụ thể. Thí dụ architects, analysts, designers, implementers, testers,§



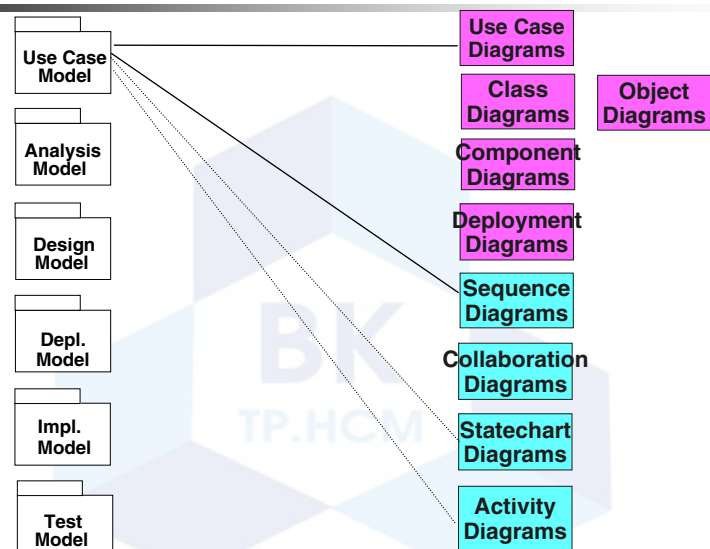
## VI. Quy trình phát triển phần mềm RUP



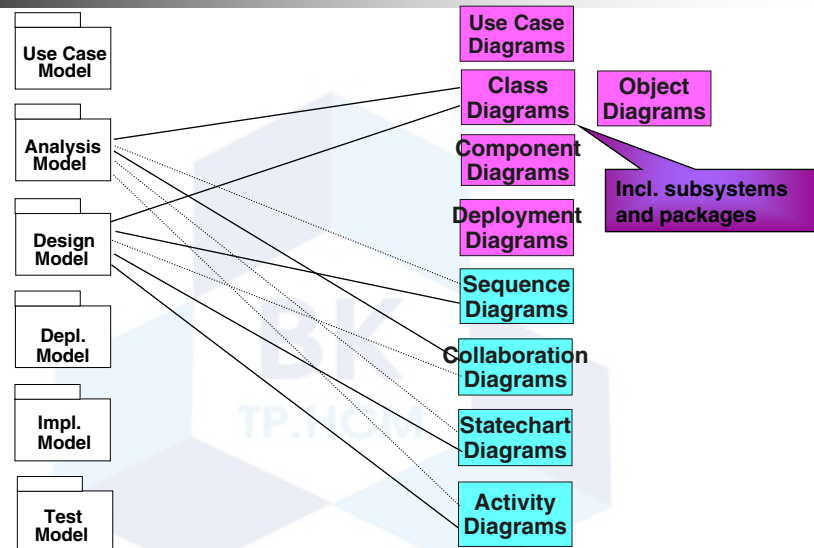
## VI. Qui trình phát triển phần mềm RUP



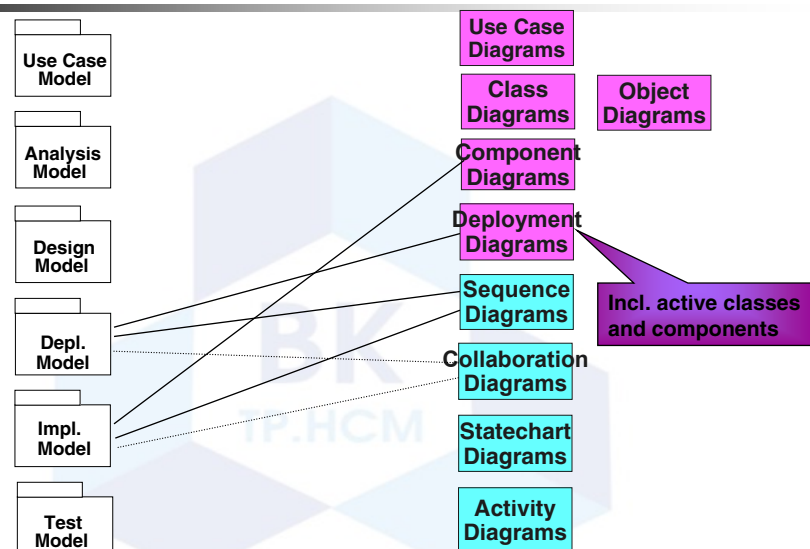
## VI. Qui trình phát triển phần mềm RUP



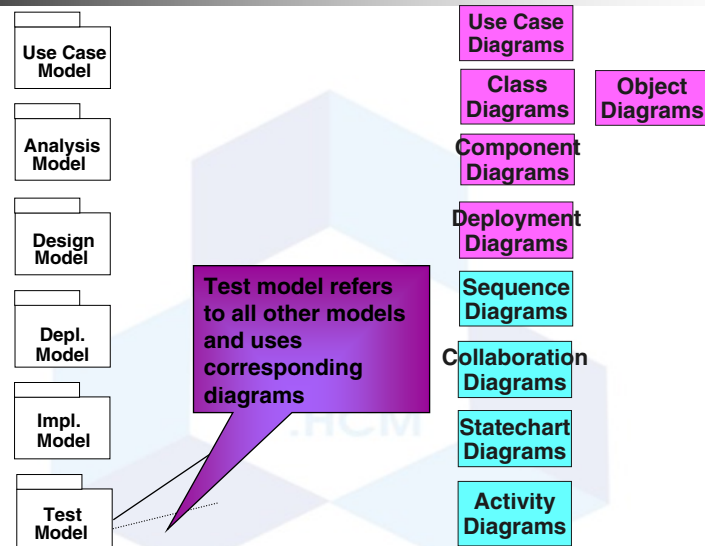
## VI. Qui trình phát triển phần mềm RUP



## VI. Qui trình phát triển phần mềm RUP



## VI. Qui trình phát triển phần mềm RUP



## VII. Vài thông tin về nắm bắt yêu cầu

- ❑ Nắm bắt yêu cầu (Requirements) là xác định chính xác, rõ ràng và đầy đủ các yêu cầu chức năng và phi chức năng của chương trình cần xây dựng. Sau khi nắm bắt được từng yêu cầu, ta phải lập tài liệu để đặc tả yêu cầu đó.
- ❑ Ta dùng use-case và bảng đặc tả use-case để đặc tả yêu cầu chức năng.
- ❑ Ta dùng kịch bản thuộc tính chất lượng để miêu tả thuộc tính chất lượng phần mềm tương ứng.



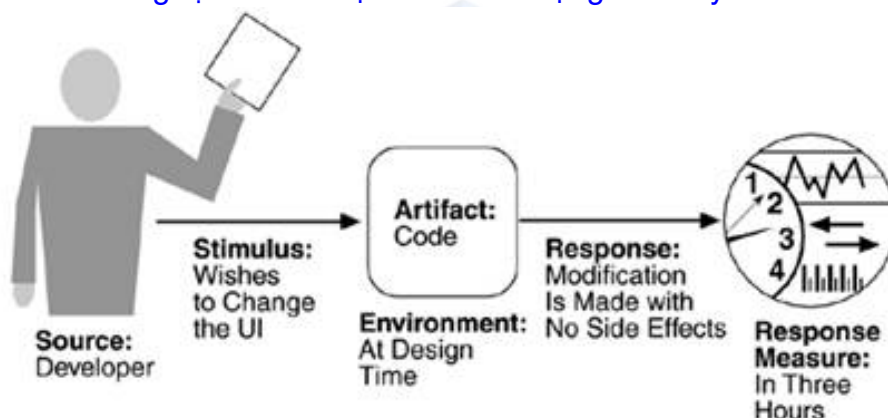
## VII. Vài thông tin về nắm bắt yêu cầu

- Thí dụ ta có thể đặc tả yêu cầu chức năng đăng ký khóa học của SV bằng bảng đặc tả chức năng như sau (xem file Word đính kèm).



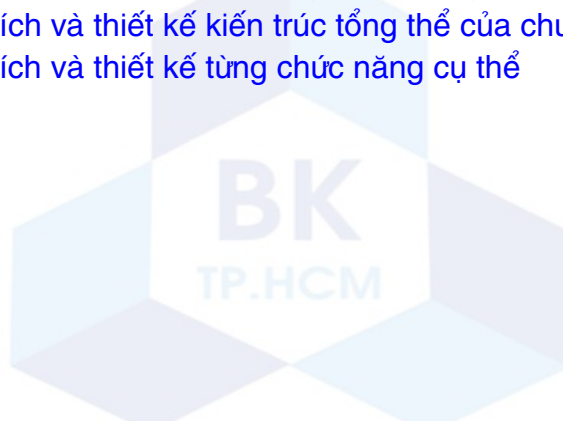
## VII. Vài thông tin về nắm bắt yêu cầu

- Thí dụ ta có thể đặc tả yêu cầu về tính dễ hiệu chỉnh của chương trình bằng kịch bản thuộc tính chất lượng sau đây :



## VII. Vài thông tin về nắm bắt yêu cầu

- ❑ Các yêu cầu chức năng và phi chức năng nắm bắt được sẽ được dùng làm thông tin đầu vào cho hoạt động phân tích và thiết kế.
- ❑ Hoạt động phân tích và thiết kế gồm 2 mảng chính :
  - Phân tích và thiết kế kiến trúc tổng thể của chương trình
  - Phân tích và thiết kế từng chức năng cụ thể



## VIII. Kết chương

- ❑ Chương này đã giới thiệu cho chúng ta bức tranh toàn cục về thiết bị tổng quát hóa là máy tính số, cách thức nhờ máy tính thực hiện dùm 1 bài toán, chi phí xây dựng và duy trì phần mềm, các hoạt động chức năng của phát triển phần mềm, qui trình phát triển phần mềm và ý nghĩa của thuật ngữ công nghệ phần mềm.

