

Chương 4

Quy trình phát triển phần mềm RUP

- 4.1 Quy trình phát triển phần mềm RUP
- 4.2 Các mô hình cần tạo trong mỗi workflow
- 4.3 Các lược đồ cần tạo trong mỗi mô hình
- 4.4 Xây dựng các lược đồ như thế nào ?
- 4.5 Kết chương



4.1 Quy trình phát triển phần mềm RUP

- Quy trình phát triển là gì ? Nói theo kiểu thực dụng thì quy 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 mục tiêu đề ra.

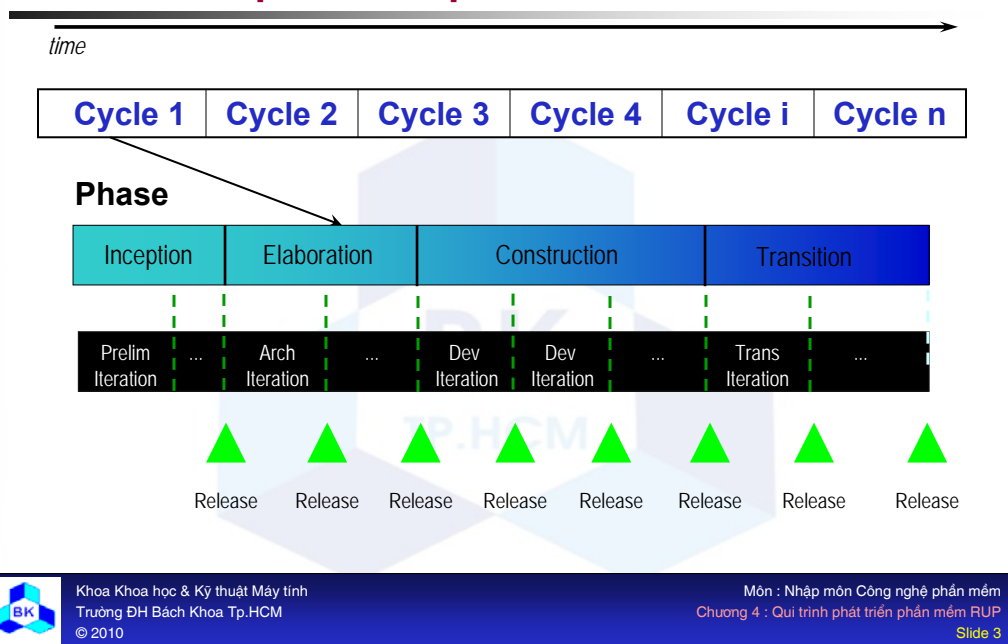
Các yêu cầu
ban đầu hay
yêu cầu mới về
phần mềm

Software Engineering
Process

Hệ thống phần
mềm ban đầu
hay hệ thống kế
tiếp



4.1 Qui trình phát triển phần mềm RUP



4.1 Qui trình phát triển phần mềm RUP

- ❑ 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 phần mềm, bắt đầu khi có các yêu cầu cần giải quyết (ban đầu hay mới) đến khi có version mới giải quyết tốt các yêu cầu đó.
- ❑ Công đoạn (Phase) : hoạt động chức năng nhỏ cần thực hiện trong từng chu kỳ, thí dụ như Inception (nhắm bắt yêu cầu), Elaboration (phân tích và thiết kế), Construction (hiện thực và kiểm thử), Transition (chuyển giao).
- ❑ 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.



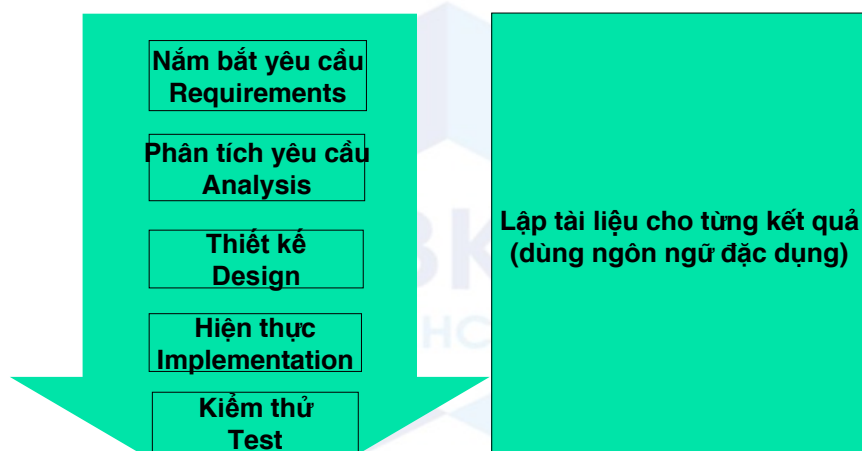
4.1 Quy trình phát triển phần mềm RUP

- ❑ 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 hay ngôn ngữ khác nhau, trong đó ngôn ngữ UML được dùng chủ yếu nhất. Tùy theo mức độ ta có dạng artifacts như 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,...



4.1 Quy trình phát triển phần mềm RUP

- ❑ Như đã nói trong chương 1, cho dù dùng quy trình phát triển phần mềm nào thì ta cũng phải thực hiện các workflows sau :



4.1 Qui trình phát triển phần mềm RUP

- ❑ Nếu chỉ 1 người thực hiện trên các ứng dụng nhỏ thì ta sẽ dùng cách thực hiện tuần tự từng workflows : nắm bắt yêu cầu phân tích từng yêu cầu, thiết kế, viết code, kiểm thử, chuyển giao.
- ❑ Tuy nhiên, nếu chỉ thực hiện tuần tự 1 lần các workflows trên thì kết quả rất khó đúng và đủ. Ta cần phải lặp các workflows trên nhiều lần :
 - Lặp từng workflow đến khi kết quả của nó tốt nhất có thể có trước khi dùng cho workflow sau.
 - Thực hiện tuần tự 5 workflow, nếu cần làm lại thì lặp lại chúng...
 - Kết hợp giữa 2 kỹ thuật trên theo yêu cầu cụ thể.



4.1 Qui trình phát triển phần mềm RUP

- ❑ Qui trình phát triển phần mềm RUP sẽ thích hợp trong trường hợp ta thực hiện ứng dụng lớn, phức tạp cần rất nhiều loại người tham gia và tốn 1 thời gian dài để thực hiện. Qui trình RUP dựa trên ý tưởng lặp tăng tiến các hoạt động như sau :
 - Lúc đầu, các workers chịu trách nhiệm workflow nắm bắt yêu cầu phần mềm sẽ tiến hành nắm bắt yêu cầu sơ khởi để có được kết quả ban đầu (thường chỉ xác định được 1 ít yêu cầu chính yếu). Kết quả sẽ được chuyển cho nhóm workers phân tích. Lúc này workers nắm bắt yêu cầu sẽ lặp lại công việc của mình để tạo được kết quả tốt hơn lần trước.
 - Khi nhận được các yêu cầu phần mềm cần thực hiện, workers phân tích yêu cầu sẽ thực hiện việc phân tích từng yêu cầu, phát họa sơ lược cách giải quyết nó. Khi có được kết quả nào thì chuyển ngay cho đội thiết kế rồi tiếp tục phân tích yêu cầu khác hay lặp lại việc phân tích yêu cầu cũ.

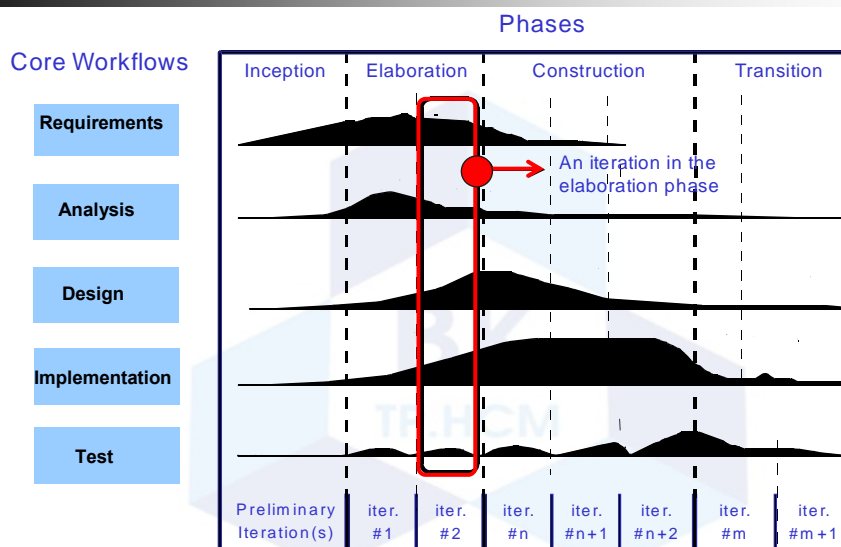


4.1 Qui trình phát triển phần mềm RUP

- Khi nhận được bản phân tích của 1 yêu cầu nào đó, workers thiết kế sẽ thực hiện việc thiết kế chi tiết nó, khi thiết kế chi tiết được class chức năng nào thì gửi ngay kết quả cho workers hiện thực rồi tiếp tục thiết kế các class còn lại hay lặp lại việc thiết kế class đã thiết kế rồi.
- Khi nhận được bản thiết kế của 1 class chức năng nào đó, workers hiện thực sẽ viết code cho class đó, khi viết code xong tác vụ chức năng nào thì gửi ngay kết quả cho workers kiểm thử rồi tiếp tục viết code cho các tác vụ còn lại hay lặp lại việc code cho tác vụ đã viết rồi.
- Khi nhận được code của 1 hàm hay 1 class nào đó, workers kiểm thử sẽ kiểm thử đơn vị nó, rồi kiểm thử tích hợp, kiểm thử hệ thống...



4.1 Qui trình phát triển phần mềm RUP



4.2 Các mô hình cần tạo trong mỗi workflow

- ❑ Mục tiêu của qui trình phát triển phần mềm RUP là phải xây dựng được đặc tả đầy đủ và chính xác về phần mềm.
- ❑ Phần mềm lớn và phức tạp chỉ được đặc tả đầy đủ thông qua nhiều góc nhìn. Trong toán học, ta có không gian 3 chiều để miêu tả 1 vật thể.
- ❑ Hiện nay, nhiều người đồng ý các dạng góc nhìn sau :
 - Góc nhìn theo thời gian : lúc bắt đầu, lúc ở phân nửa, lúc cuối cùng...
 - Góc nhìn không gian : từ ngoài vào, nhìn nội bộ bên trong,...
 - Góc nhìn theo độ trừu tượng : vĩ mô, vi mô...
 - Góc nhìn tĩnh động : góc nhìn cấu trúc tĩnh, góc nhìn hành vi động...
 - Kết hợp nhiều dạng góc nhìn theo cơ chế phân cấp.



4.2 Các mô hình cần tạo trong mỗi workflow

- ❑ Để đặc tả đầy đủ phần mềm theo 1 góc nhìn nào đó, ta dùng thuật ngữ mô hình (Model). Mỗi mô hình sẽ đặc tả đầy đủ phần mềm theo 1 góc nhìn mà mô hình muốn. Qui trình RUP cố gắng xây dựng các mô hình chính yếu sau :
 - Mô hình use-case : thể hiện góc nhìn từ ngoài nhìn vào phần mềm. Mô hình này được xây dựng và duy trì bởi workflow nắm bắt yêu cầu.
 - Mô hình phân tích : thể hiện góc nhìn bên trong, nó cho ta thấy mức độ sơ lược ban đầu về cấu trúc phần mềm (vĩ mô nhất). Mô hình này được xây dựng và duy trì bởi workflow phân tích yêu cầu.
 - Mô hình thiết kế : thể hiện góc nhìn bên trong, nó cho ta thấy mức độ chi tiết về cấu trúc phần mềm (vi mô). Mô hình này được xây dựng và duy trì bởi workflow thiết kế.

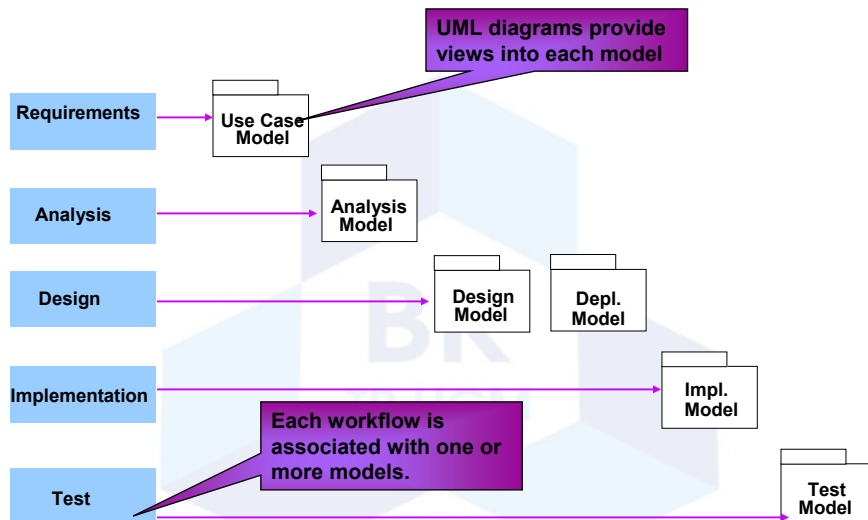


4.2 Các mô hình cần tạo trong mỗi workflow

- Mô hình triển khai : thể hiện góc nhìn bên trong, nó cho ta thấy yêu cầu về các tài nguyên phần cứng và mối quan hệ giao tiếp giữa chúng để vận hành phần mềm. Mô hình này được xây dựng và duy trì bởi workflow thiết kế, hiện thực.
- Mô hình hiện thực : thể hiện góc nhìn bên trong, nó cho ta thấy cụ thể về công nghệ, ngôn ngữ và mã nguồn của từng phần tử trong bản thiết kế (class). Mô hình này được xây dựng và duy trì bởi workflow hiện thực.
- Mô hình kiểm thử : thể hiện góc nhìn bên trong, nó cho ta thấy các thông tin về việc kiểm thử phần mềm. Mô hình này được xây dựng và duy trì bởi workflow kiểm thử.



4.2 Các mô hình cần tạo trong mỗi workflow

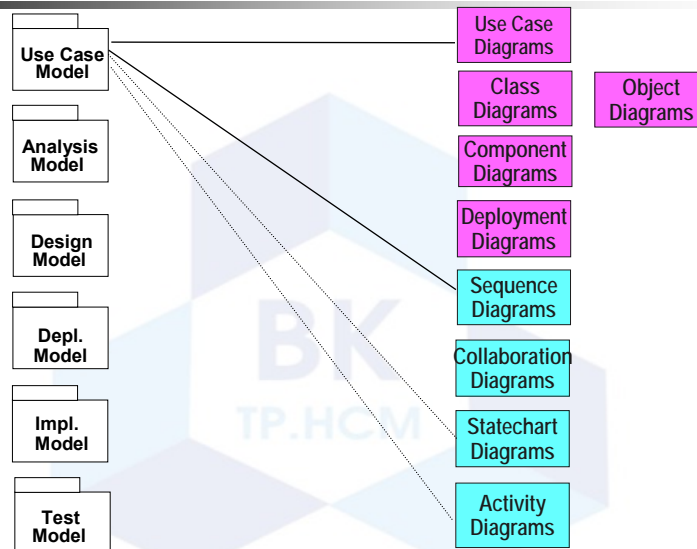


4.3 Các lược đồ cần tạo trong mỗi mô hình

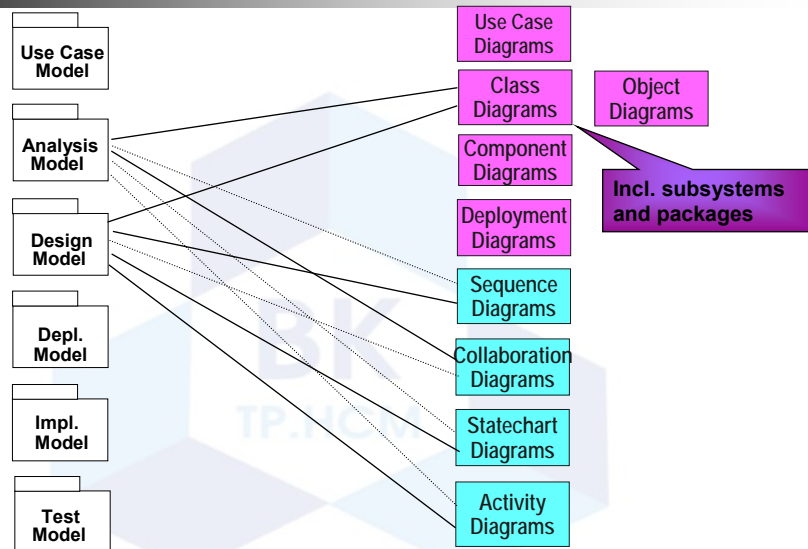
- ❑ Như chúng ta đã nói, mỗi mô hình sẽ đặc tả đầy đủ phần mềm theo 1 góc nhìn mà mô hình muốn. Đối với phần mềm lớn và phức tạp, mỗi mô hình miêu tả nó thường lớn và phức tạp, khó lòng được miêu tả vật lý bởi 1 phần tử.
- ❑ Ta phân ra mô hình thành nhiều phần tử nhỏ hơn để dễ xây dựng, quản lý và duy trì theo thời gian. Lược đồ (Diagram) là loại phần tử nhỏ hơn, nó được dùng chủ yếu để xây dựng mô hình.
- ❑ Mỗi lược đồ cho ta đặc tả được 1 bộ phận nhỏ của phần mềm theo góc nhìn xác định :
 - Góc nhìn cấu trúc tĩnh : lược đồ class, lược đồ đối tượng, lược đồ thành phần, lược đồ triển khai.
 - Góc nhìn hành vi động : lược đồ trình tự, lược đồ cộng tác, lược đồ chuyển trạng thái, lược đồ hoạt động.



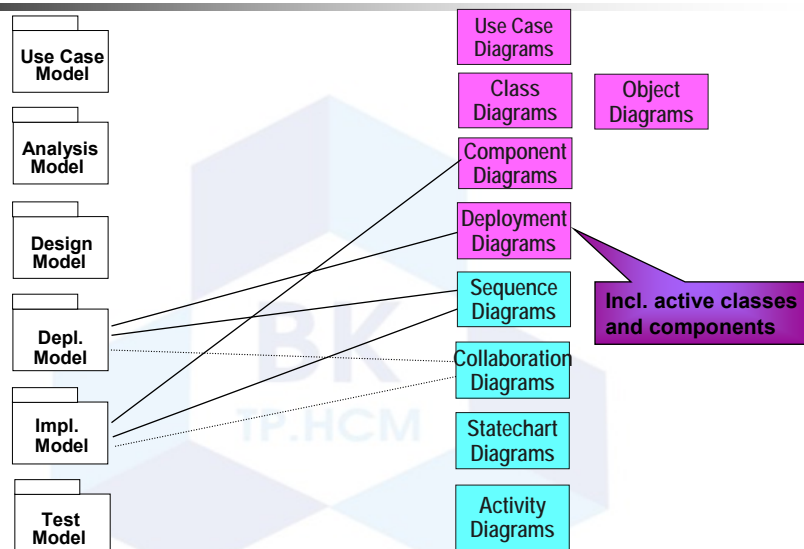
4.3 Các lược đồ cần tạo trong mỗi mô hình



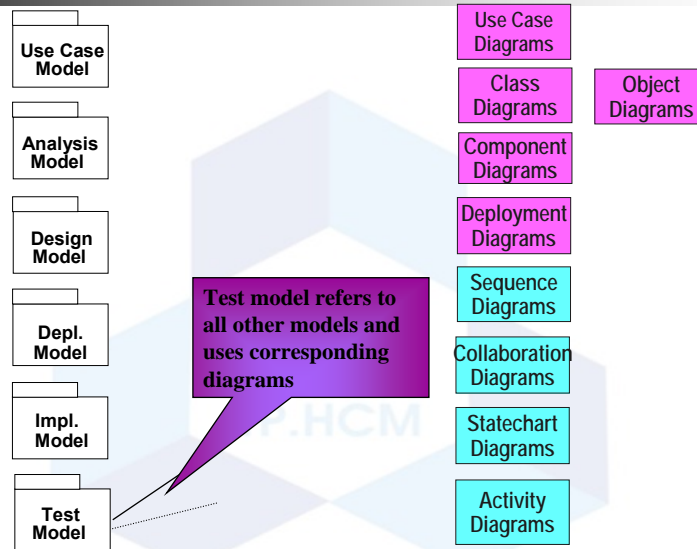
4.3 Các lược đồ cần tạo trong mỗi mô hình



4.3 Các lược đồ cần tạo trong mỗi mô hình



4.3 Các lược đồ cần tạo trong mỗi mô hình



4.4 Xây dựng các lược đồ như thế nào ?

- ❑ Ta sẽ xây dựng các lược đồ như thế nào để thỏa mãn các mục tiêu sau :
 - Nhanh chóng, dễ dàng trong việc xây dựng.
 - Kết quả phải thể hiện đúng, rõ ràng, đơn nghĩa về thông tin cần đặc tả.
 - Đọc, hiểu dễ dàng, đơn nghĩa và nhanh chóng bởi nhiều người, cho dù họ dùng ngôn ngữ tự nhiên nào.
- ❑ Ngôn ngữ mô hình UML mà ta sẽ trình bày trong chương 5 tiếp theo có những tính chất giúp ta xây dựng các lược đồ thỏa mãn các mục tiêu trên.



4.4 Kết chương

- Chương này đã giới thiệu cho chúng ta các thông tin cơ bản về đời sống phần mềm, về quy trình phát triển phần mềm RUP, về các workflows cần thực hiện, về các mô hình cũng như các lược đồ cần xây dựng và duy trì.

