

Chương 3 : CÁC THÀNH PHẦN ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM TRONG VÒNG ĐỜI DỰ ÁN



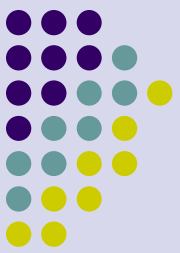
- 1) Thảo luận cách tích hợp các thành phần SQA vào vòng đời dự án.
- 2) Thảo luận về các thành phần xét duyệt của giai đoạn thiết kế: xét duyệt thiết kế hình thức, xét duyệt ngang hàng và ý kiến chuyên gia.

Chương 3 : CÁC THÀNH PHẦN ĐAM BAO CHẤT LƯỢNG PHẦN MỀM TRONG VÒNG ĐỜI DỰ ÁN



- 3) Trình bày các thành phần SQA được dùng cho giai đoạn vận hành-bảo trì.

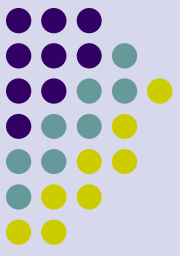
1) Thảo luận cách tích hợp các thành phần SQA vào vòng đời dự án.



Mô hình của quy trình phát triển phần mềm phổ biến:

- Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)
- Mô hình bản mẫu (Prototyping model)
- Mô hình xoắn ốc (Spiral model)
- Mô hình hướng đối tượng (Object-oriented model).

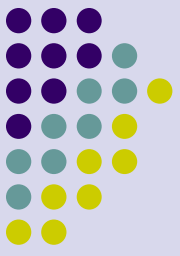
Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



Mô hình cổ điển (vẫn đang được sử dụng hiện nay) nó cung cấp một mô tả toàn diện nhất về quy trình.

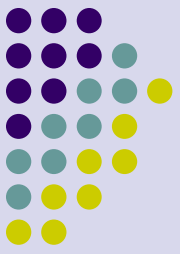
Trong giai đoạn đầu của quy trình phát triển phần mềm, các tài liệu thiết kế của sản phẩm sẽ được chuẩn bị.

Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



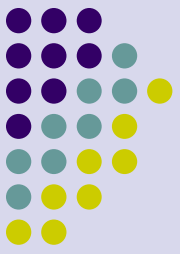
Định nghĩa yêu cầu. Khách hàng phải định nghĩa các yêu cầu của họ. Trong nhiều trường hợp, một hệ thống phần mềm là một phần của một hệ thống lớn hơn. Thông tin về các phần khác của hệ thống được mở rộng sẽ giúp thiết lập sự hợp tác giữa các nhóm phát triển và phát triển các giao diện thành phần.

Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



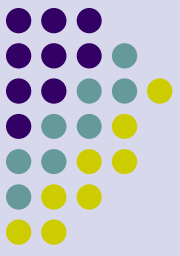
phân tích. Nỗ lực chính ở đây là để phân tích ý nghĩa các yêu cầu để xây dựng mô hình ban đầu cho hệ thống phần mềm.

Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



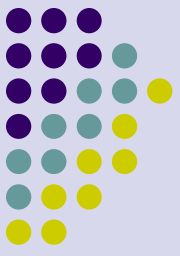
Thiết kế. Giai đoạn này liên quan đến việc định nghĩa chi tiết các đầu ra, đầu vào và thủ tục xử lý, bao gồm cả cấu trúc dữ liệu và cơ sở dữ liệu, cấu trúc phần mềm,

Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



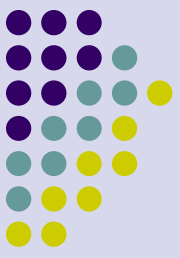
Lập trình. Các thiết kế được chuyển sang mã lệnh. Việc viết mã lệnh liên quan đến các hoạt động đảm bảo chất lượng như thanh tra, kiểm thử đơn vị và kiểm thử tích hợp.

Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



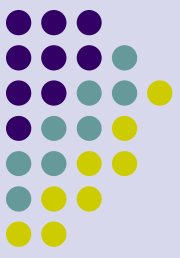
Kiểm thử hệ thống. Các kiểm thử hệ thống được thực hiện một khi giai đoạn lập trình đã hoàn tất. Mục tiêu chính của kiểm thử là để tìm ra càng nhiều lỗi càng tốt để đạt được một mức độ chất lượng chấp nhận được cho phần mềm.

Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



Cài đặt và chuyển đổi. Sau khi hệ thống phần mềm đã được phê duyệt, hệ thống sẽ được cài đặt để được sử dụng. Nếu hệ thống phần mềm mới được dùng để thay thế cho hệ thống hiện tại, thì sẽ có một quá trình chuyển đổi phần mềm được thực hiện để đảm bảo rằng các hoạt động của công ty có thể tiếp tục mà không bị gián đoạn trong suốt thời gian chuyển đổi.

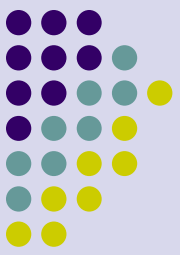
Mô hình vòng đời phát triển phần mềm (Software Development Life Cycle model)



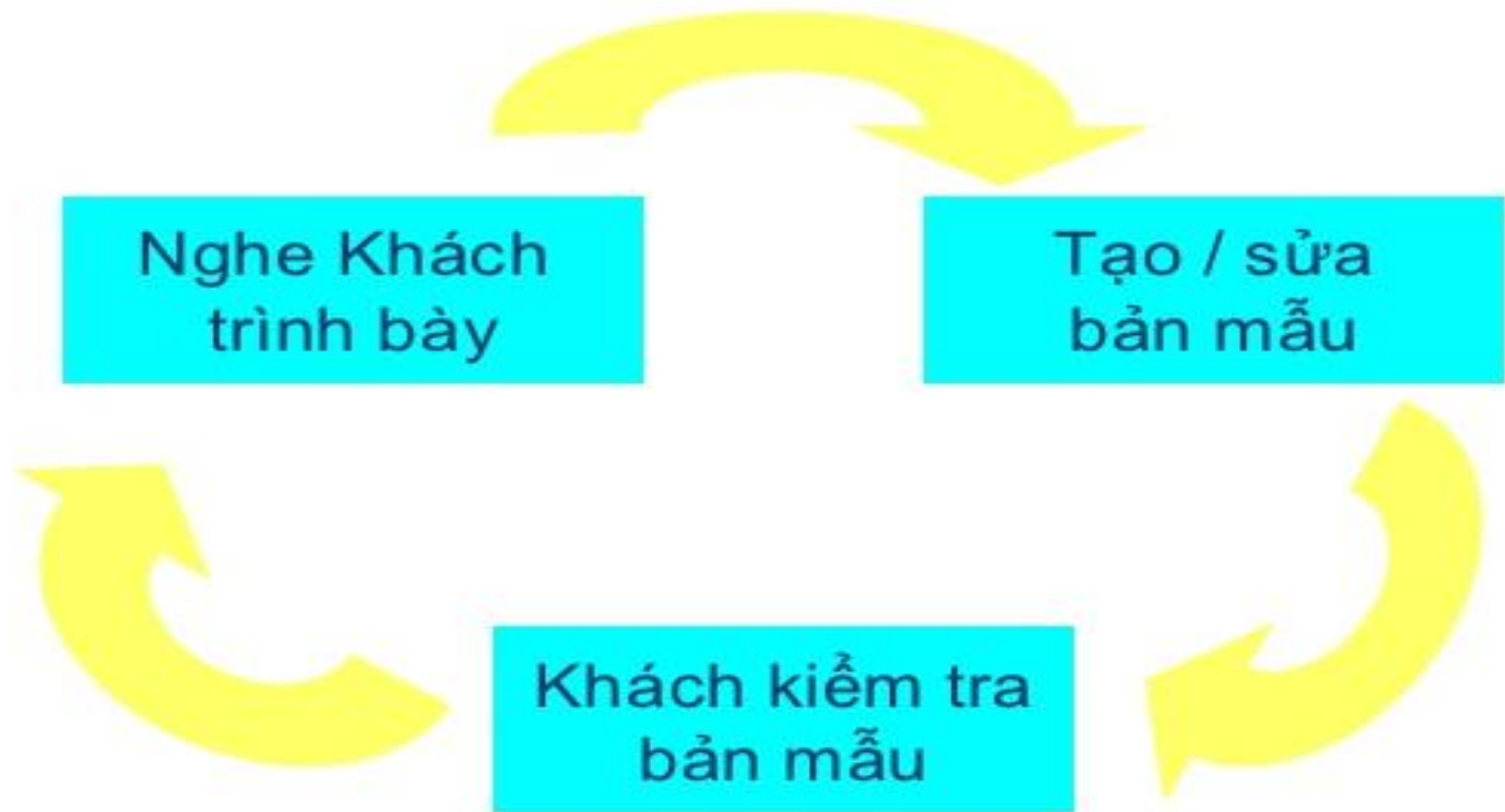
Vận hành và bảo trì:

Phần mềm được đưa vào sử dụng (vận hành) một khi đã hoàn thành việc cài đặt và chuyển đổi.

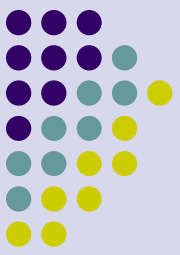
Đến khi một thể hệ phần mềm mới xuất hiện hoặc có lỗi, thì việc bảo trì được cần đến



Mô hình bản mẫu (Prototyping model)

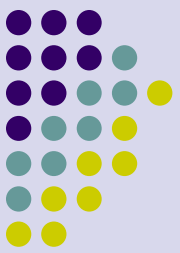


Mô hình bản mẫu (*Prototyping model*)

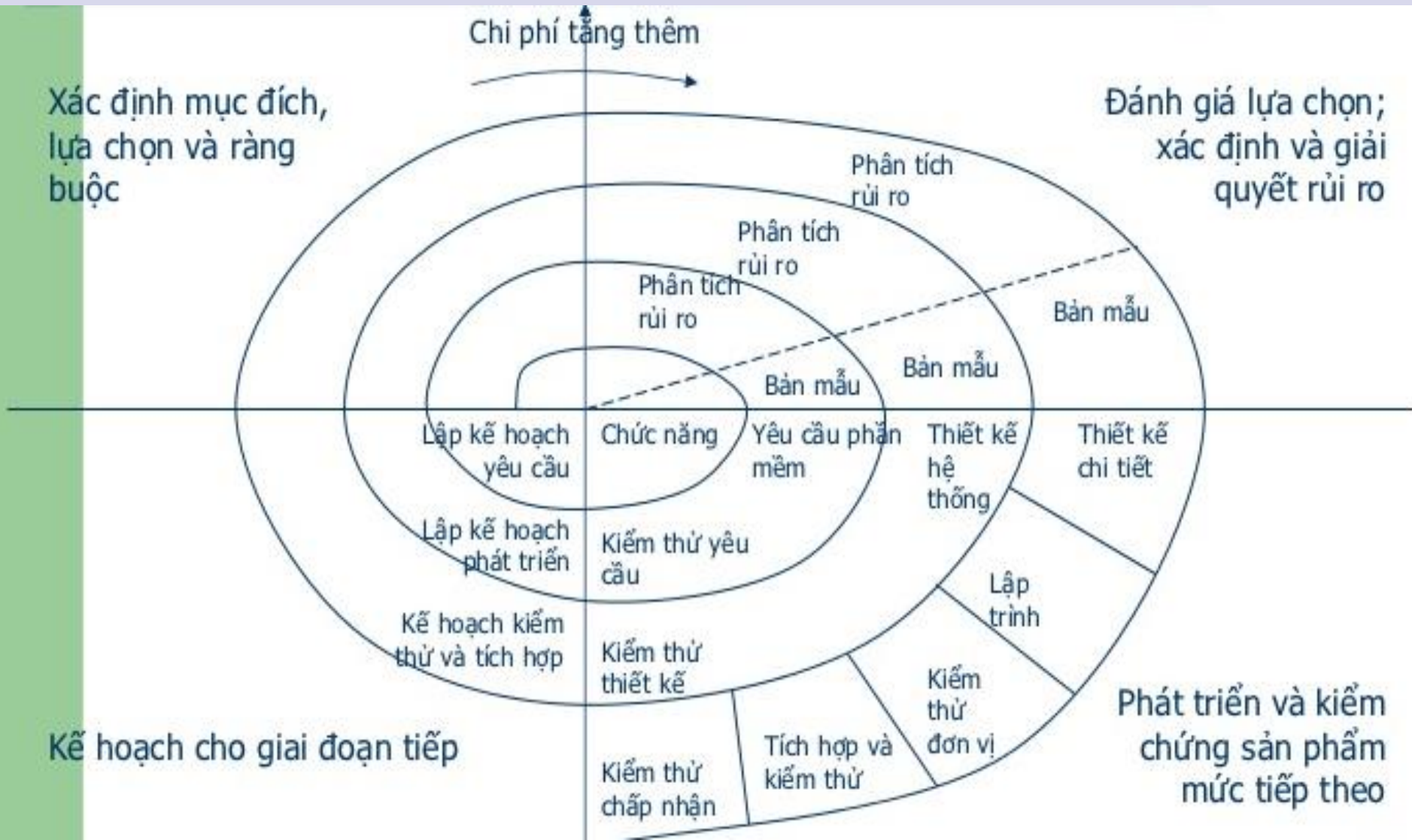


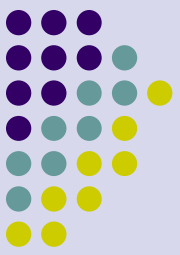
Mô hình bản mẫu (Prototyping model)

Việc tạo bản mẫu có thể được sử dụng kết hợp với các phương pháp khác hoặc được sử dụng "độc lập". Nói cách khác, mức độ tạo bản mẫu có thể khác nhau, từ việc chỉ thay thế một giai đoạn của mô hình thác nước (hoặc của một phương pháp khác) cho đến việc tạo ra bản mẫu cho toàn bộ hệ thống phần mềm.



Mô hình xoắn ốc (Spiral model)

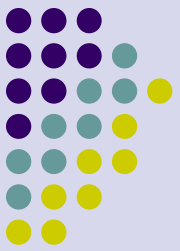




Mô hình xoắn ốc (Spiral model)

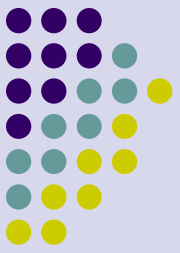
Mô hình xoắn ốc cung cấp một phương pháp cải tiến để giám sát các dự án lớn và phức tạp thông qua việc phô bày ra các viễn cảnh thất bại, nhất là các dự án được thực hiện trong nhiều năm.

Mô hình hướng đối tượng (Object-oriented model).



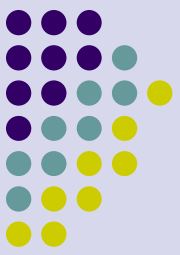
Mô hình hướng đối tượng khác với các mô hình khác ở chỗ nó tập trung vào việc tăng cường tái sử dụng lại phần mềm. Phương pháp này có đặc trưng là dễ dàng tích hợp các mô-đun phần mềm hiện có (đối tượng hoặc thành phần) vào hệ thống phần mềm đang được phát triển

Mô hình hướng đối tượng (Object-oriented model).

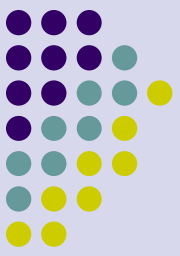


- Kinh tế
- Cải thiện chất lượng
- Rút ngắn thời gian phát triển

Các yếu tố ảnh hưởng đến cường độ sử dụng các hoạt động SQA trong quy trình



SQA được thực hiện sau khi một giai đoạn hoặc một cột mốc của dự án được hoàn thành và bị ảnh hưởng bởi các yếu tố dự án và nhóm.



Các yếu tố ảnh hưởng đến cường độ sử dụng các hoạt động SQA trong quy trình

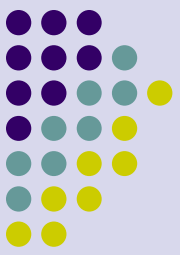
Yếu tố dự án:

Độ lớn của dự án

Độ phức tạp và khó khăn về mặt kỹ thuật

Mức độ tái sử dụng lại các thành phần phần mềm

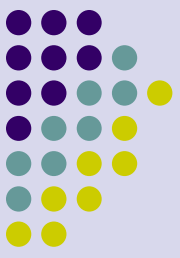
Mức độ nghiêm trọng của các đầu ra nếu như dự án bị hỏng hóc



Các yếu tố ảnh hưởng đến cường độ sử dụng các hoạt động SQA trong quy trình

Yếu tố nhóm

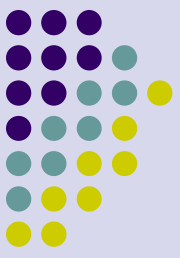
- Trình độ chuyên môn của thành viên nhóm
- Mức độ hiểu biết về dự án và kinh nghiệm của nhóm đối với lĩnh vực ứng dụng
- Mức độ sẵn sàng của đội ngũ nhân viên - những người có thể hỗ trợ chuyên môn cho nhóm
- Mức độ thân mật giữa các thành viên trong nhóm, nói cách khác là tỷ lệ phần trăm của các thành viên mới trong nhóm



Các yếu tố ảnh hưởng đến cường độ sử dụng các hoạt động SQA trong quy trình

Ba khía cạnh đảm bảo chất lượng của sản phẩm phần mềm : kiểm chứng, thẩm định, và kiểm tra sự đủ tiêu chuẩn:

Kiểm chứng (Verification) Xác định xem các sản phẩm của một giai đoạn phát triển nào đó có thỏa các điều kiện đã được đặt ra vào lúc bắt đầu của giai đoạn đó hay không.

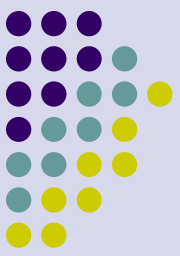


Các yếu tố ảnh hưởng đến cường độ sử dụng các hoạt động SQA trong quy trình

Thẩm định (Validation): Đánh giá một hệ thống hoặc một thành phần trong suốt hoặc vào cuối của quy trình phát triển để xác định xem nó có thỏa các yêu cầu đã được xác định hay không.

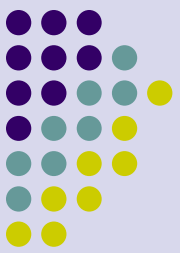
Kiểm tra sự đủ tiêu chuẩn (Qualification): Quy trình được dùng để xác định xem một hệ thống hoặc một thành phần có phù hợp cho sử dụng hay không.

2) Các thành phần xét duyệt của giai đoạn thiết kế



Xét duyệt (IEEE, 1990) “Một quá trình hoặc một cuộc họp mà trong đó một sản phẩm công việc hoặc một tập các sản phẩm công việc được đưa ra tới toàn thể cá nhân tham gia vào dự án, các giám đốc, người dùng, khách hàng và các bên quan tâm đến dự án nhằm lấy . kiến phê bình và phê chuẩn”

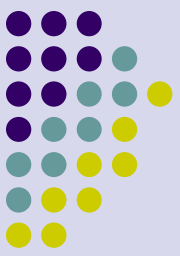
2) Các thành phần xét duyệt của giai đoạn thiết kế



Mục đích:

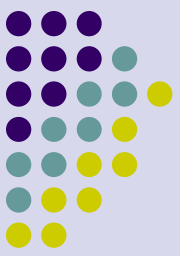
- Mục đích trực tiếp:
 - Phát hiện lỗi phân tích và thiết kế.
 - Xác định các rủi ro mới.
 - Xác định sự sai lệch so với mẫu, các kiểu thủ tục và qui ước.
 - Để phê chuẩn sản phẩm của phân tích hoặc thiết kế.

2) Các thành phần xét duyệt của giai đoạn thiết kế

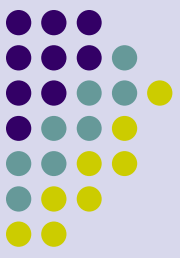


- Mục đích gián tiếp:
 - Nơi họp mặt không chính thức để trao đổi về những kiến thức chuyên môn.
 - Ghi lại những lỗi phân tích và thiết kế sẽ hỗ trợ một cơ sở cho những hoạt động
 - Sửa chữa lỗi trong tương lai.

2) Các thành phần xét duyệt của giai đoạn thiết kế



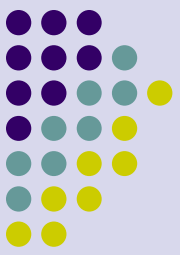
- a) Xét duyệt thiết kế hình thức
- b) Xét duyệt ngang hàng
- c) Ý kiến chuyên gia.



a) Xét duyệt thiết kế hình thức

Rà soát thiết kế hình thức(DRs-formal Design Reviews) là rà soát duy nhất cần thiết cho việc phê duyệt sản phẩm thiết kế.

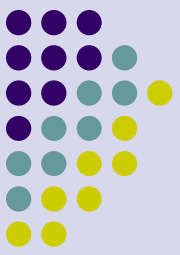
Rà soát thiết kế hình thức có thể được thực hiện tại bất cứ mốc phát triển nào yêu cầu sự hoàn thiện của tài liệu phân tích hay thiết kế.



a) Xét duyệt thiết kế hình thức

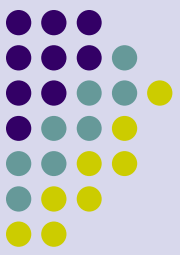
Dách các review thiết kế chính thức :

- DPR – Development Plan Review : Review kế hoạch phát triển
- SRSR – Software Requirement Specification Review : Review đặc tả yêu cầu phần mềm
- PDR – Preliminary Design Review : Review thiết kế sơ bộ



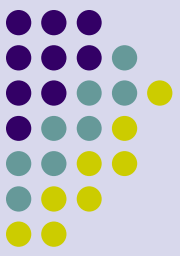
a) Xét duyệt thiết kế hình thức

- DBDR- Detailed Design Review : Review thiết kế chi tiết
- TPR – Test Plan Review : Review kế hoạch kiểm thử
- STPR – Software Test Procedure Review : Review thủ tục kiểm thử phần mềm
- VDR- Version Description Review : Review mô tả phiên bản
- OMR- Operator Manual Review : Review vận hành thủ công



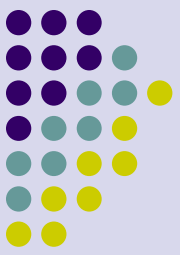
a) Xét duyệt thiết kế hình thức

- SMR- Support Manual Review :Review trợ giúp thủ công
- - TRR- Test Readiness Review : Review sự sẵn sàng kiểm thử
- - PRR- Product Release Review : Review bản phát hành sản phẩm
- - IPR-Installation Plan Review : Review kế hoạch cài đặt



a) Xét duyệt thiết kế hình thức

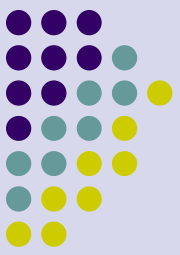
- **Các nhân tố ảnh hưởng tới DRs**
 - Những người tham gia
 - Sự chuẩn bị trước
 - Phiên DR
 - Các hoạt động sau DR được đề xuất



a) Xét duyệt thiết kế hình thức

Những người tham gia rà soát thiết kế

- Review leader
- Review team.



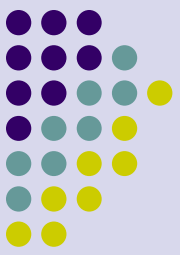
a) Xét duyệt thiết kế hình thức

Sự chuẩn bị cho một phiên bản DR

Được hoàn thành bởi 3 thành viên: review leader, review team và development team..

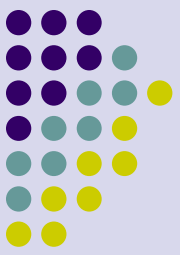
- **Chuẩn bị của Review leader**

- Bổ nhiệm các thành viên nhóm
- Lập lịch các phiên review
- Phân chia tài liệu thiết kế cho các thành viên của nhóm



a) Xét duyệt thiết kế hình thức

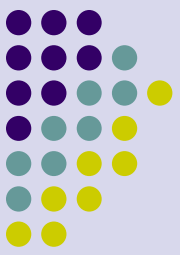
- **Chuẩn bị của Review team**
 - Xem lại tài liệu thiết kế
 - Danh sách bình luận
- **Chuẩn bị của Development team**
 - Trình diễn ngắn tài liệu thiết kế
 - Checklist các công việc review



a) Xét duyệt thiết kế hình thức

- **Phiên DR:**

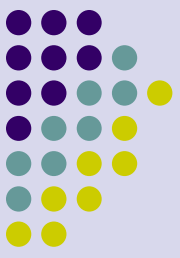
1. Trình diễn ngắn gọn về tài liệu thiết kế
2. Các bình luận của các thành viên review team
3. Kiểm tra và xác nhận thảo luận mỗi bình luận



a) Xét duyệt thiết kế hình thức

4. Các quyết định về tài liệu thiết kế để xác định tiến trình dự án. Các quyết định có thể có 3 loại :

- Phê duyệt đầy đủ
- Phê duyệt từng phần
- Từ chối phê duyệt

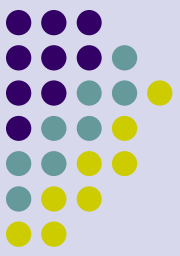


a) Xét duyệt thiết kế hình thức

Các hoạt động hậu review

Báo cáo Review:

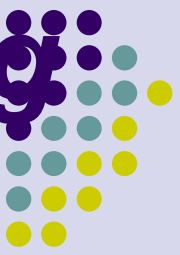
- Review leader thực hiện sau phiên review
- Bao gồm :
 - o Tổng kết các thảo luận review
 - o Quyết định về sự tiếp tục của dự án
 - o Danh sách các hoạt động cần thiết phải làm
 - o Tên thành viên chịu trách nhiệm theo sát việc hiệu chỉnh



a) Xét duyệt thiết kế hình thức

- Tiến trình theo dõi
 - Review leader thực hiện
 - Chắc rằng việc hiệu chỉnh được thực hiện đúng đắn
 - Việc theo dõi cần được ghi lại

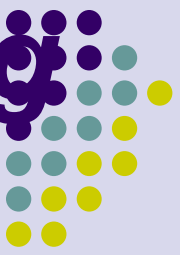
b) Các rà soát ngang hàng (peer review)



Mục đích chính của peer review là xác định lỗi và độ lệch dựa vào các chuẩn.

- Có hai phương pháp peer reviews:
 - xét duyệt (inspection)
 - kiểm tra từng bước (walkthrough).
- Walkthrough phát hiện sai sót và ghi chú lên tài liệu.

b) Các rà soát ngang hàng (peer review)



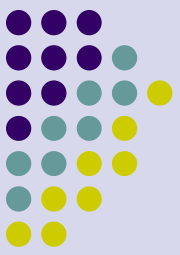
- **Những người tham gia vào peer reviews**

Một đội peer review tối ưu 3-5 người tham gia.

Tất cả những người tham gia nên là những người cùng địa vị của nhà thiết kế hệ thống phần mềm.

Một đội peer review đề cử bao gồm:

- Một leader review.
- Một người thực thi (author).
- Các chuyên gia đặc biệt (specialized professionals).

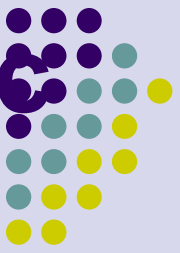


c) Các ý kiến của chuyên gia

Lợi ích:

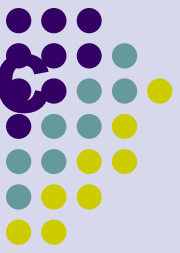
- Thiếu sự hiểu biết đầy đủ về lĩnh vực nào đó.
- Tạm thời thiếu những người chuyên nghiệp để tham gia vào đội xem xét lại
- Các thành viên chuyên nghiệp cao cấp trong tổ chức không thống nhất được với nhau.
- Trong các tổ chức nhỏ số lượng ứng viên phù hợp cho đội xem xét lại là không đủ.

c) Đảm bảo chất lượng của các thành phần bảo trì phần mềm



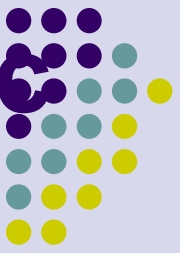
- 3 thành phần của dịch vụ bảo trì và nó được xem như là bản chất của sự thành công:
 - Bảo trì sửa lỗi: những dịch vụ hỗ trợ người dùng và sửa lỗi phần mềm.
 - Bảo trì thích ứng: làm cho các gói phần mềm thích ứng với những yêu cầu mới của khách hàng và điều kiện môi trường thay đổi.

c) Đảm bảo chất lượng của các thành phần bảo trì phần mềm



- Bảo trì cải thiện chất lượng: kết hợp (1) bảo trì hoàn thiện những chức năng mới được thêm vào phần mềm cũng như nâng cao hiệu suất, cùng với (2) bảo trì phòng ngừa – những hoạt động cải thiện độ tin cậy và cơ sở hạ tầng hệ thống cho dễ dàng làm cho việc bảo trì trong tương lai hiệu quả hơn.

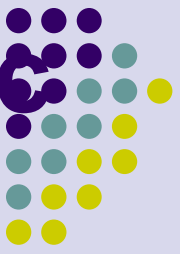
c) Đảm bảo chất lượng của các thành phần bảo trì phần mềm



- Những khó khăn của người dùng có thể được gây ra bởi nguyên nhân sau:

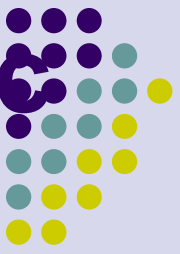
Lỗi code (thường được dùng với thuật ngữ “thất bại phần mềm”- software failure”).

c) Đảm bảo chất lượng của các thành phần bảo trì phần mềm



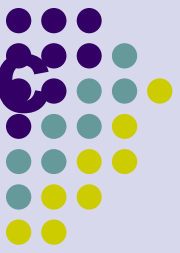
- Lỗi viết tài liệu thủ công, giúp những màn ảnh hoặc các hình thức tài liệu hướng dẫn được chuẩn bị sẵn sàng cho người dùng. Trong trường hợp này, dịch vụ hỗ trợ có thể cung cấp cho người dùng với những chỉ dẫn đúng (mặc dù không có sự chính xác trong chính tài liệu phần mềm được thực hiện).

c) Đảm bảo chất lượng của các thành phần bảo trì phần mềm



- Không đầy đủ, mơ hồ hoặc tài liệu không chính xác.
- Thiếu kiến thức về hệ thống phần mềm của người dùng hoặc thất bại của họ khi sử dụng tài liệu được cung cấp. Trong những trường hợp này không có xét đến thất bại hệ thống phần mềm.

c) Đảm bảo chất lượng của các thành phần bảo trì phần mềm

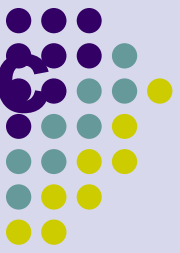


● *Mục tiêu hoạt động QA bảo trì phần mềm:*

(1) Đảm bảo, với mức độ tin cậy được chấp nhận, rằng những hoạt động bảo trì phù hợp với những yêu cầu kỹ thuật chức năng.

(2) Đảm bảo, với mức độ tin cậy được chấp nhận, rằng những hoạt động bảo trì phù hợp với những yêu cầu quản lý lập lịch và ngân sách.

c) Đảm bảo chất lượng của các thành phần bảo trì phần mềm

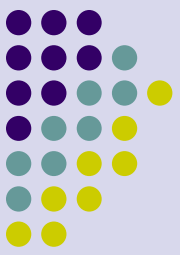


(3) Những hoạt động khởi đầu và quản lý. nhằm cải thiện và tăng hiệu quả cho bảo trì phần mềm và những hoạt động SQA. Điều này liên quan đến việc cải thiện cái nhìn toàn cảnh để đạt được những yêu cầu về chức năng và quản lý trong khi giá thành giảm.

Các thành phần chất lượng phần mềm tiền bảo trì



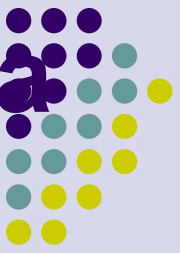
- Xem xét lại hợp đồng bảo trì
- Xây dựng kế hoạch bảo trì.



Xem xét lại hợp đồng bảo trì

Trước khi bắt đầu cung cấp những dịch vụ bảo trì phần mềm tới từng nhóm khách hàng trên thì hợp đồng bảo trì tương ứng phải được hoàn thành để nó đánh giá tổng số những trách nhiệm bảo trì theo những điều kiện liên quan.

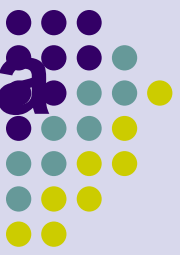
Sự rõ ràng trong yêu cầu của khách hàng



- **Những vấn đề sau đây đặc biệt được quan tâm:**

- Loại dịch vụ bảo trì sửa đổi được yêu cầu: Danh sách những dịch vụ từ xa và những dịch vụ tại chỗ được cung cấp, giờ phục vụ, thời gian phản hồi
- Số lượng người sử dụng và kiểu ứng dụng được dùng.

Sự rõ ràng trong yêu cầu của khách hàng



- Những người dùng địa phương, ở khoảng cách xa (hoặc qua đại dương) và các kiểu dịch vụ được cài đặt tại mỗi nơi đó.
- Cung cấp bảo trì cải thiện chức năng, khả năng thích ứng và thủ tục cho những yêu cầu của dịch vụ cũng như đề xuất và phê duyệt việc thực thi cho những dịch vụ này.



Xem xét lại những phương pháp tiếp cận khác về các điều khoản trong văn bản bảo trì

Xem xét những suy xét cụ thể dưới đây để đưa ra lựa chọn phù hợp:

- Những hợp đồng con cho những site (địa điểm) hoặc kiểu dịch vụ
- Việc thực thi một vài dịch vụ thông qua khách hàng cùng với sự hỗ trợ từ nhóm bảo trì của nhà cung cấp.



Xem xét lại những phương pháp tiếp cận khác về các điều khoản trong văn bản bảo trì

Nhìn lại sự ước lượng về tài nguyên bảo trì được yêu cầu:

- Đầu tiên, những ước lượng nên được kiểm tra lại trên cơ sở của những dịch vụ được yêu cầu, sắp xếp theo đề xuất của nhóm. Sau đó, dựa vào năng lực của công ty để xem có đáp ứng được những khía cạnh chuyên môn cũng như là tính sẵn sàng của nhóm bảo trì đã phân tích hay không.



Xem xét lại những phương pháp tiếp cận khác về các điều khoản trong văn bản bảo trì

- **Xem xét lại những dịch vụ bảo trì được cung cấp bởi những hợp đồng con và/hoặc khách hàng.**

Việc nhìn lại này đề cập tới việc định danh lại những dịch vụ được cung cấp bởi mỗi người tham gia, số tiền trả cho những hợp đồng con, đảm bảo chất lượng và những thủ tục tiếp theo được đáp ứng.



Xem xét lại những phương pháp tiếp cận khác về các điều khoản trong văn bản bảo trì

- **Xem xét lại những ước lượng về giá thành bảo trì :**

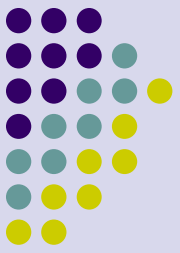
Những ước lượng này nên được nhìn lại dựa trên cơ sở của những tài nguyên được yêu cầu.

Các công cụ đảm bảo lượng bảo trì phần mềm



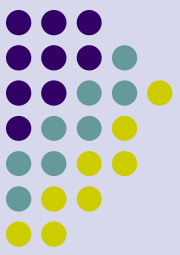
- 1) Các công cụ SQA dùng cho bảo trì khắc phục.**
- 2) Các công cụ cơ sở hạ tầng SQA dùng cho bảo trì phần mềm**

1) Các công cụ SQA dùng cho bảo trì khắc phục.



- a) Các dịch vụ hỗ trợ người dùng.
- b) Các sửa chữa phần mềm (sửa chữa lỗi).

a) Các dịch vụ hỗ trợ người dùng.

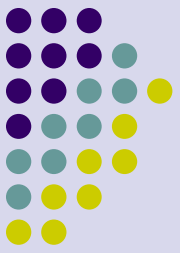


Đối phó với các trường hợp có sai sót/lỗi trong mã lệnh hoặc trong tài liệu của phần mềm.

Tài liệu không đầy đủ hoặc không rõ ràng.

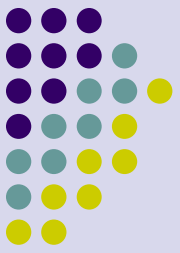
Người không có đủ kiến thức về phần mềm hoặc sử dụng sai tài liệu hướng dẫn.

b) Các sửa chữa phần mềm (sửa chữa lỗi).

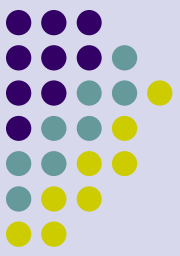


Sửa chữa lỗi và tài liệu - được yêu cầu trong trường hợp phần mềm xảy ra hỏng hóc.

1) Các công cụ SQA dùng cho bảo trì khắc phục.



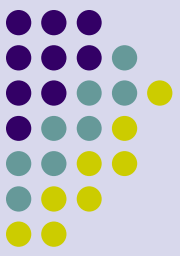
Các dịch vụ này thường được cung cấp trong giai đoạn đầu sử dụng/vận hành phần mềm.



2) Các công cụ cơ sở hạ tầng SQA dùng cho bảo trì phần mềm

Các công cụ cơ sở hạ tầng SQA góp phần làm nâng cao chất lượng bảo trì theo hai cách:

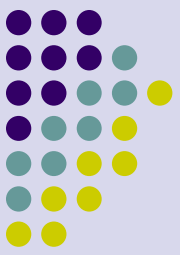
- + Hỗ trợ các nhóm phát triển phần mềm để tạo ra các sản phẩm có chất lượng cao.
- + Bằng cách hỗ trợ các nhóm bảo trì đang chịu trách nhiệm để bảo trì các sản phẩm phần mềm đó.



2) Các công cụ cơ sở hạ tầng SQA dùng cho bảo trì phần mềm

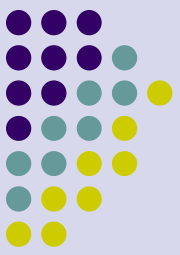
Các công cụ phổ biến là:

- Các thủ tục bảo trì và hướng dẫn công việc.
- Đào tạo và cấp chứng chỉ cho nhóm bảo trì.
- Ngăn chặn và khắc phục.



2) Các công cụ cơ sở hạ tầng SQA dùng cho bảo trì phần mềm

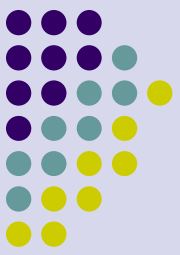
- Quản lý cấu hình.
- Kiểm soát tài liệu và hồ sơ chất lượng.
- Các thiết bị hỗ trợ chất lượng.



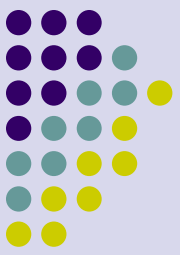
Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Kiểm thử phần mềm là công cụ đảm bảo chất lượng phần mềm đầu tiên được áp dụng để kiểm soát chất lượng sản phẩm phần mềm trước khi giao hoặc cài đặt sản phẩm cho khách hàng.

Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử



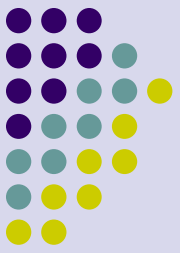
Khi các chuyên gia xem kiểm thử là một phần của đảm bảo chất lượng phần mềm, điều này dẫn đến sự hình thành của kiểm thử đơn vị và kiểm thử tích hợp.



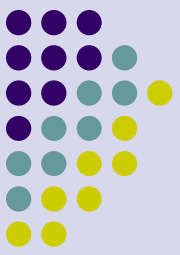
Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Kiểm thử không phải là công cụ SQA duy nhất được áp dụng cho đảm bảo chất lượng phần mềm giai đoạn kiểm thử. Ngoài ra còn có thanh tra và rà soát mã - các phương pháp này được thực hiện cho các phần mã được in ra mà không thật sự chạy chương trình.

Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử



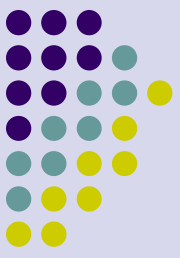
Tuy nhiên, bởi vì những công cụ này chỉ dựa trên việc xét duyệt các tài liệu, nên chúng không bao giờ có thể thay thế cho kiểm thử.



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

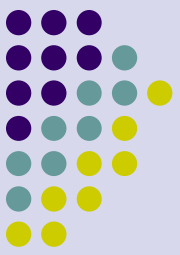
Một định nghĩa cổ điển của Myers:

"Kiểm thử là quy trình thực thi một chương trình với mục đích để tìm kiếm lỗi."



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

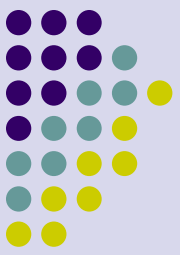
Kiểm thử đơn vị tập trung vào việc xác minh trên đơn vị nhỏ nhất của thiết kế phần mềm. Sử dụng các mô tả thiết kế thử tục để hướng dẫn, các đường dẫn điều khiển quan trọng được kiểm thử để phát hiện lỗi trong phạm vi modul.



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Kiểm thử đơn vị nhằm phát hiện các lỗi trong các phạm vi của modul bao gồm:

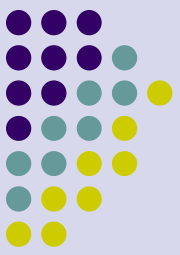
- ❑ Giao diện modul
- ❑ Cấu trúc dữ liệu cục bộ
- ❑ Điều kiện biên
- ❑ Đường dẫn độc lập
- ❑ Đường dẫn xử lý lỗi



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Kiểm thử đơn vị nhằm phát hiện các lỗi trong các phạm vi của modul bao gồm:

- ❑ Giao diện modul
- ❑ Cấu trúc dữ liệu cục bộ
- ❑ Điều kiện biên
- ❑ Đường dẫn độc lập
- ❑ Đường dẫn xử lý lỗi

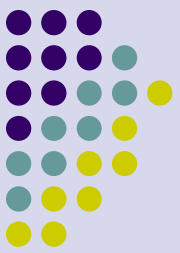


Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Kiểm thử đơn vị thường được xem như 1 phần phụ cho bước mã hóa sau khi mã nguồn được phát triển.

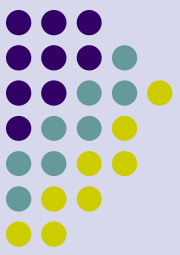
Khi chỉ 1 chức năng được gọi bởi một modul, số các trường hợp kiểm thử được giảm xuống và các lỗi có thể dự đoán và phát hiện sớm hơn

Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

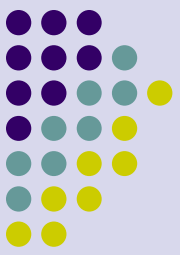


Unit Test thường do lập trình viên thực hiện. Unit Test đòi hỏi kiểm tra viên có kiến thức về thiết kế và code của chương trình mục đích của Unit Test.

Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

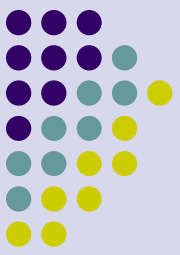


Kiểm thử tích hợp là 1 kỹ thuật có hệ thống để xác định cấu trúc chương trình trong khi thực hiện các kiểm thử nhằm phát hiện các lỗi liên quan đến điều kiện. Mục tiêu là lấy các thành phần đã được kiểm thử và xây dựng cấu trúc chương trình đã được mô tả bởi thiết kế.



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

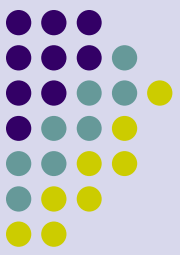
- Integration Test có 2 mục tiêu chính
 - + Phát hiện lỗi giao tiếp xảy ra giữa các unit
 - + Tích hợp các Unit đơn lẻ thành các hệ thống nhỏ và cuối cùng là nguyên hệ thống hoàn chỉnh chuẩn bị cho kiểm tra ở mức hệ thống.



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Có 4 loại kiểm tra trong Integration Test:

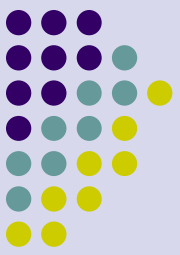
1) Kiểm tra chức năng: Kiểm tra chỉ chú trọng đến chức năng của chương trình, không quan tâm đến cấu trúc bên trong, chỉ khảo sát chức năng của chương trình theo yêu cầu kĩ thuật



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Có 4 loại kiểm tra trong Integration Test:

- 2) Kiểm tra cấu trúc: Nhằm bảo đảm các thành phần bên trong của 1 chương trình chạy đúng
- 3) Kiểm tra hiệu năng: Kiểm tra việc vận hành của hệ thống
- 4) Kiểm tra khả năng chịu tải: Kiểm tra các giới hạn của hệ thống

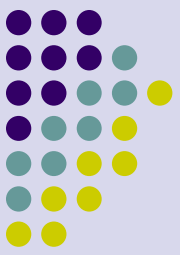


Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Mục tiêu cuối cùng của quá trình kiểm thử:

Mục tiêu trực tiếp

- Xác định ra càng nhiều lỗi càng tốt.
- Đưa phần mềm được kiểm thử, sau khi đã sửa chữa các lỗi được xác định và kiểm thử lại, đến một mức độ chất lượng chấp nhận.
- Thực hiện kiểm thử một cách hiệu quả với hiệu suất cao, trong giới hạn ngân sách và thời gian.



Đảm bảo chất lượng phần mềm trong giai đoạn kiểm thử

Mục tiêu gián tiếp

- Lập hồ sơ về các lỗi phần mềm (để sử dụng cho các hoạt động khắc phục và ngăn chặn lỗi).