
Chương 6

Kiểm tra yêu cầu

(Validating the Requirements)

Nội dung

- ❑ Xác định độ ưu tiên
- ❑ Kiểm tra tính hợp lệ yêu cầu - Requirement validation
 - Review
 - Informal review
 - Formal review (inspection)
 - Testing

Tại sao phải ưu tiên hóa yêu cầu

- Dự án phần mềm có tài nguyên hạn chế cần xác định độ ưu tiên (priorities) các chức năng của sản phẩm
- Việc ưu tiên hóa giúp cho nhà quản lý dự án giải quyết các xung đột, cân đối giữa các yêu cầu
- Xếp thứ tự ưu tiên nghĩa là cân bằng các lợi ích nghiệp vụ (business benefits) của mỗi yêu cầu dựa trên chi phí và ảnh hưởng của nó tới kiến trúc và sự tiến hóa của sản phẩm.

Tại sao phải ưu tiên hóa yêu cầu

- ❑ Người quản lý dự án phải cân đối giữa phạm vi dự án và các ràng buộc về thời gian, ngân sách, nhân viên và chất lượng.
- ❑ Cách giải quyết: bỏ qua hay trì hoãn sang phiên bản kế tiếp các yêu cầu có độ ưu tiên thấp .
- ❑ Nếu khách hàng không phân biệt được yêu cầu nào quan trọng hơn thì người quản lý dự án phải quyết định thay. Và nếu khách hàng không đồng ý thì phải chỉ ra yêu cầu nào cần trước và yêu cầu nào có thể đợi được.

Tại sao phải ưu tiên hóa yêu cầu

- ❑ Cả khách hàng và nhà phát triển đều phải cung cấp thông tin để ưu tiên hóa yêu cầu.
- ❑ Khách hàng thường đặt độ ưu tiên cao cho chức năng nào liên quan tới nghiệp vụ quan trọng nhất hay có lợi cho việc sử dụng. Tuy nhiên ngay khi nhà phát triển chỉ ra chi phí, các khó khăn, rủi ro kỹ thuật, khách hàng có thể chấp nhận thay đổi độ ưu tiên yêu cầu.
- ❑ Nhà phát triển có thể quyết định thực thi chức năng có độ ưu tiên thấp trước vì chức năng này ảnh hưởng đến kiến trúc toàn hệ thống

Phương pháp xác định cấp độ ưu tiên thông dụng

- ❑ Nhóm các yêu cầu thành 3 nhóm :
high, medium, và low priority

Tên	Ý nghĩa
Cao (High)	Một yêu cầu đáp ứng một mục tiêu then chốt (mission – critical); đòi hỏi phải có để thực hiện phiên bản tiếp theo.
Trung bình (Medium)	Hỗ trợ các hoạt động hệ thống cần thiết; đòi hỏi cuối cùng cũng phải có nhưng có thể đợi cho đến phiên bản sau nếu cần thiết
Thấp (Low)	Một nâng cấp về mặt chức năng hoặc chất lượng; có thể là tốt nếu có vào một lúc nào đó nếu nguồn lực cho phép.

➔ Phương pháp này chủ quan và không chính xác .

Phương pháp xác định cấp độ ưu tiên thông dụng

- ❑ Nhóm các yêu cầu thành 3 nhóm : high, medium, và low priority

Tên	Ý nghĩa	Tham chiếu
Nền tảng (Essential)	Phần mềm là không thể được chấp nhận trừ khi yêu cầu này được thực hiện theo một cách thức thỏa thuận nào đó.	IEEE 1998
Có điều kiện (Conditional)	Có thể nâng cao chất lượng sản phẩm, nhưng sản phẩm vẫn có thể được chấp nhận nếu không có.	
Lựa chọn (Optional)	Một lớp các chức năng có thể có hoặc không thể có cũng được.	
3	Cần phải được thi công hoàn hảo	Kovitz 1999
2	Các nhu cầu cần phải làm, nhưng không nhất thiết phải làm tốt	
1	Có thể chứa bugs.	

➔ Phương pháp này chủ quan và không chính xác .

Phương pháp xác định cấp độ ưu tiên

- ❑ Đánh giá độ ưu tiên theo 2 chiều :
importance và *urgency*
- ❑ Mỗi yêu cầu được khảo sát theo cả 2 chiều này để xếp loại độ ưu tiên cho yêu cầu đó.

	Important	Not Important
Urgent (Cần thiết, khẩn cấp)	High Priority	Don't do these !
Not Urgent	Medium Priority	Low Priority

Các phương pháp ưu tiên hóa yêu cầu

- ❑ Có nhiều phương pháp khác nhau nhưng đều dựa vào ước tính giá trị tương đối (relative value) và chi phí tương đối (relative cost) của mỗi yêu cầu .
- ➔ Các phương pháp này không thực tế nếu hệ thống có quá nhiều yêu cầu

Phương pháp ưu tiên hóa TQM (Total Quality Management)

- ❑ Xếp loại mỗi yêu cầu dựa theo nhiều tiêu chuẩn có gán trọng số (weighted) của dự án và tính điểm để xếp hạng độ ưu tiên cho các yêu cầu .

Phương pháp ưu tiên hóa QFD (Quality Function Deployment)

- ❑ Dựa vào giá trị của khách hàng (customer) với từng tính năng được đề xuất của sản phẩm (proposed product features)
- ❑ Chỉ áp dụng cách ưu tiên hóa này cho các yêu cầu thu thập được, không áp dụng cho các yêu cầu ở mức top priority (như chức năng nghiệp vụ chính của hệ thống, các quy định của chính quyền ...)

Ví dụ áp dụng phương pháp QFD cho hệ thống Chemical Tracking

- Giá trị khách hàng bao gồm lợi ích (benefit) mà khách hàng có được khi 1 tính năng của hệ thống được thực thi và thiệt hại (penalty paid) nếu thiếu tính năng đó.
- Tính năng của hệ thống thì tỷ lệ thuận (directly proportional) với giá trị mà nó đem lại và tỷ lệ nghịch (inversely proportional) với chi phí và rủi ro kỹ thuật khi thực thi nó.
- Khi tất cả các yếu tố cân bằng nhau, các tính năng có tỷ lệ *giá trị/chi phí được điều chỉnh rủi ro cao nhất (highest risk – adjusted value/cost)* thì sẽ có mức ưu tiên cao nhất.

Ví dụ áp dụng phương pháp QFD cho hệ thống Chemical Tracking

Những người tham gia quy trình xếp thứ tự ưu tiên gồm:

- Nhà quản lý dự án, người lãnh đạo thực hiện quy trình, phân giải các xung đột, điều chỉnh ý kiến tham gia của các thành viên khác nếu cần thiết.
- Những người đại diện khách hàng chính, ví dụ người đại diện sản phẩm, là những người đưa ra xếp hạng lợi ích và trừng phạt.
- Đại diện những người phát triển, ví dụ trưởng nhóm kỹ thuật, người cung cấp thông tin về xếp hạng chi phí và rủi ro.

Ví dụ áp dụng phương pháp QFD cho hệ thống Chemical Tracking

Thực hiện các bước sau khi sử dụng quy trình xếp thứ tự ưu tiên này:

1. Liệt kê trên bảng tính (spreadsheet) tất cả các yêu cầu, các tính năng, hoặc các use cases mà bạn muốn xếp thứ tự ưu tiên;
2. Ước lượng lợi ích tương đối mà mỗi tính năng mang lại cho khách hàng hoặc cho hoạt động nghiệp vụ trên theo thang từ 1 đến 9, trong đó 1 là lợi ích không đáng kể, 9 là lợi ích vô cùng lớn.
3. Ước lượng các trừng phạt tương đối mà khách hàng hoặc nhu cầu kinh doanh bị vi phạm nếu tính năng này không được đưa vào. Cũng sử dụng một thang đánh giá từ 1 đến 9, 1 là không trừng phạt và 9 là trừng phạt nặng.

Ví dụ áp dụng phương pháp QFD cho hệ thống Chemical Tracking

4. Cột Tổng giá trị (Total Value) là tổng lợi ích tương đối và trừng phạt tương đối. Một cách ngầm định, lợi ích và trừng phạt có trọng số như nhau.
5. Ước lượng chi phí tương đối để thực thi mỗi tính năng, căn cứ trên thang từ 1 (thấp) đến 9 (cao).
6. Các nhà phát triển ước lượng mức tương đối về rủi ro kỹ thuật hoặc rủi ro khác để thực hiện mỗi tính năng theo thang từ 1 đến 9.
7. Sau khi bạn nhập tất cả các ước lượng vào bảng tính, thì giá trị ưu tiên của mỗi tính năng được tính theo công thức sau:

$$\text{Ưu tiên} = (\text{Giá trị \%}) / (\text{Chi phí \%} * \text{Trọng số chi phí} + \text{Rủi ro \%} * \text{Trọng số rủi ro})$$

8. Sắp xếp danh sách tính năng theo thứ tự giảm dần của mức ưu tiên đã được tính toán. Tính năng ở đỉnh của danh sách có sự cân bằng tốt nhất của giá trị, chi phí, rủi ro và vì vậy cần có mức ưu tiên cao nhất

Ví dụ áp dụng phương pháp QFD cho hệ thống Chemical Tracking

Relative Weights	2	1			1		0.5		
Feature	Relative Benefit	Relative Penalty	Total Value	Value %	Relative Cost	Cost %	Relative Risk	Risk %	Priority
1. Print a material safety data sheet.	2	4	8	5.2	1	2.7	1	3.0	1.22
2. Query status of a vendor order.	5	3	13	8.4	2	5.4	1	3.0	1.21
3. Generate a Chemical Stockroom inventory report.	9	7	25	16.1	5	13.5	3	9.1	0.89
4. See history of a specific chemical container.	5	5	15	9.7	3	8.1	2	6.1	0.87
5. Search vendor catalogs for a	9	8	26	16.8	3	8.1	8	24.2	0.83

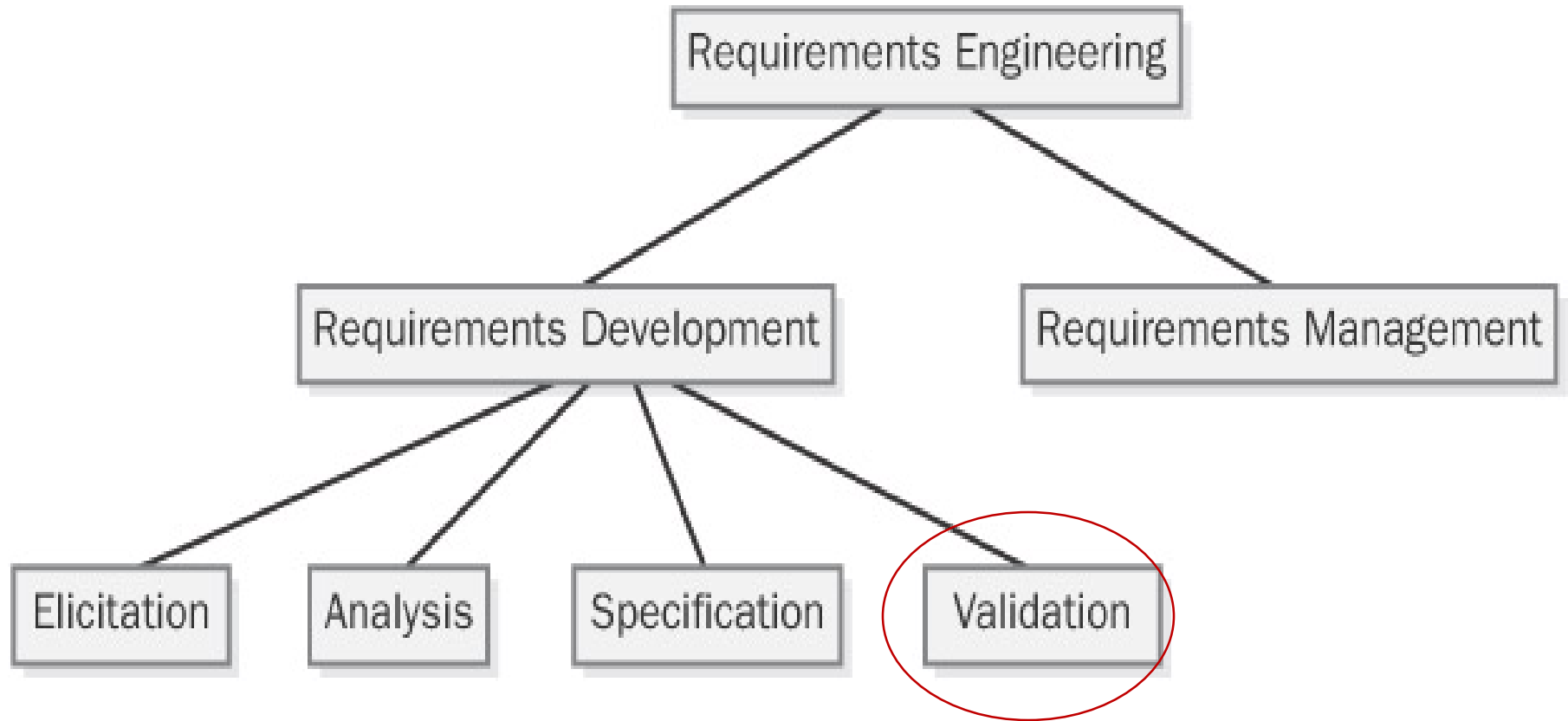
Hạn chế của phương pháp QFD

- ❑ Độ chính xác của kỹ thuật này bị phụ thuộc vào khả năng của nhóm khi ước tính lợi nhuận, chi phí và rủi ro cho mỗi tính năng.
- ❑ Khách hàng nên xem xét lại bảng tính để cùng thỏa thuận với việc xếp loại độ ưu tiên này.

Assignment

- ❑ Ứng dụng mô hình xếp thứ tự ưu tiên ở đây cho 10 hoặc 15 tính năng hoặc use cases trong dự án hiện tại của bạn. Hãy so sánh xem mức độ cải thiện hoạt động thực tế khi ước lượng mức độ ưu tiên theo cách này với cách mà bạn đang làm hiện tại?
- ❑ Hãy so sánh xem cách này tốt như thế nào so với sự ước lượng bằng trực giác?
- ❑ Nếu không có sự liên hệ nào giữa những gì mà mô hình dự đoán với những gì bạn nghĩ là đúng, hãy phân tích xem phần nào của mô hình không cho kết quả hợp lý.
- ❑ Hãy cố gắng định trọng số khác nhau cho các yếu tố lợi ích, trừng phạt, chi phí và rủi ro. Điều chỉnh mô hình cho đến khi nó cung cấp kết quả nhất quán với những gì bạn kỳ vọng.
- ❑ Sau khi bạn đã thích ứng và điều chỉnh mô hình này, hãy ứng dụng nó cho dự án mới của bạn. Tích hợp các ưu tiên đã tính toán vào quy trình ra quyết định và xem liệu dự án có đạt được các kết quả làm hài lòng các stakeholders hơn so với cách tiếp cận xếp thứ tự ưu tiên trước đây.

Các thành phần của requirement engineering



Một vài thống kê

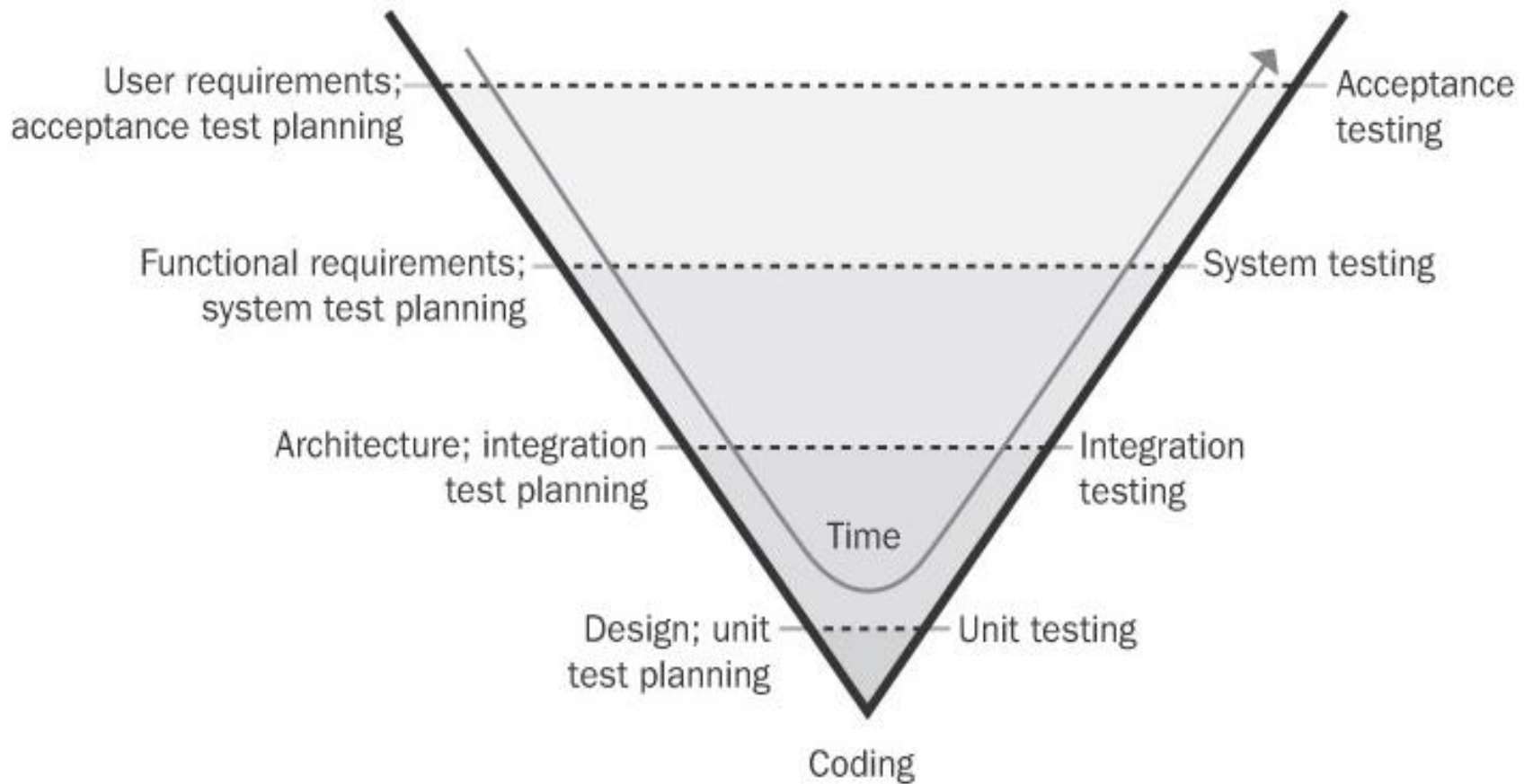
- ❑ Có thể chi phí gấp 100 lần để sửa sai 1 khiếm khuyết (defect) của chương trình so với việc sửa sai 1 lỗi (error) trong giai đoạn thu thập yêu cầu (Grady 1999).
- ❑ Mất 30 phút để sửa lỗi được phát hiện trong giai đoạn yêu cầu nhưng phải mất từ 5 – 17 giờ để sửa lỗi (defect) trong giai đoạn thử nghiệm hệ thống. (Kelly, Sherif, and Hops 1992).

Sửa lỗi trong giai đoạn yêu cầu sẽ tiết kiệm đáng kể thời gian và tiền bạc.

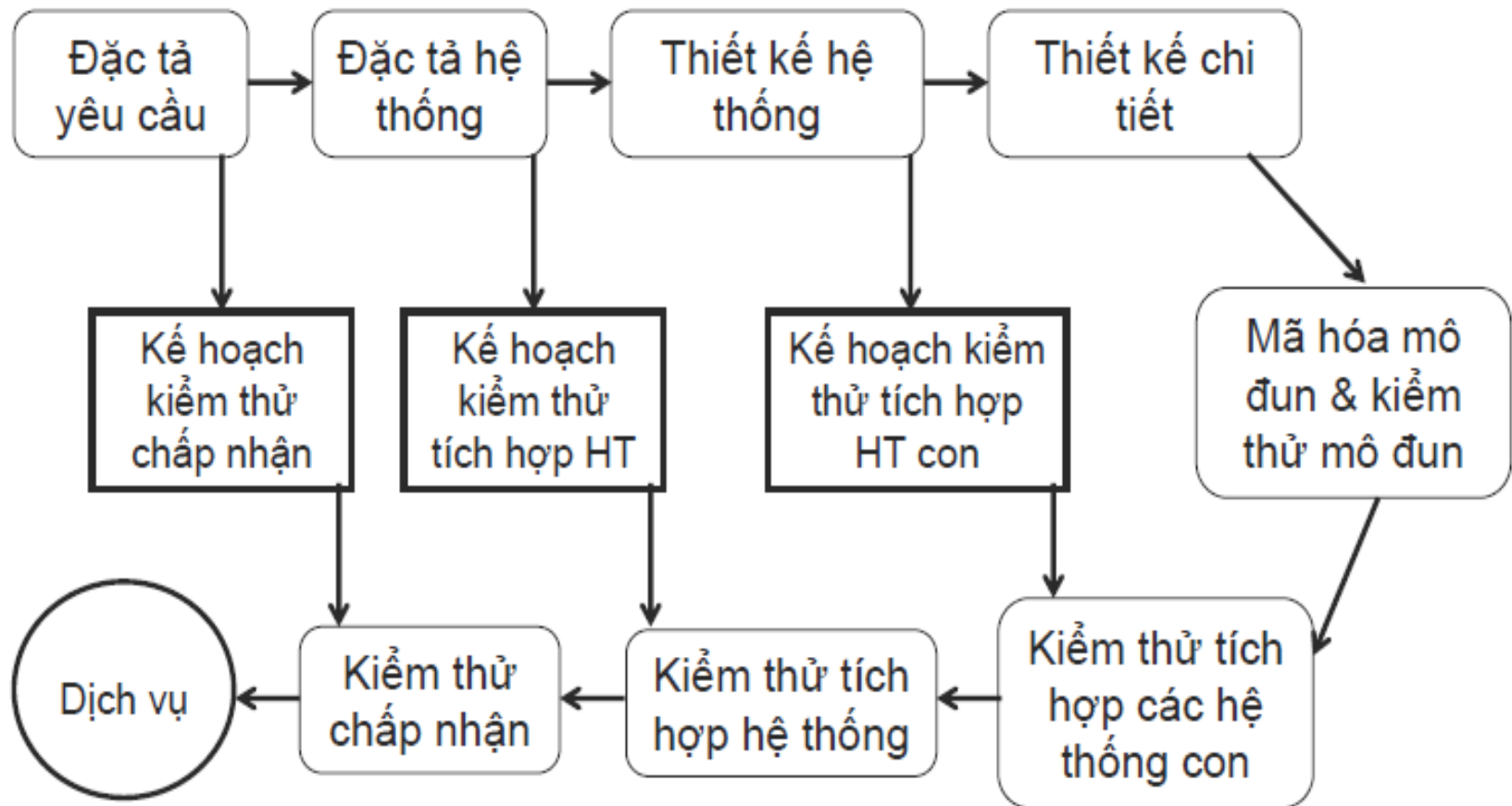
Mô hình chữ V

- ❑ Nếu bắt đầu kế hoạch kiểm thử và phát triển các test-case sớm, sẽ dễ phát hiện lỗi ngay khi chúng vừa xuất hiện, tránh tổn thất và giảm chi phí thử nghiệm và chi phí bảo trì.
- ❑ Trong mô hình chữ V, các hoạt động kiểm thử bắt đầu song song với các hoạt động phát triển hệ thống.
 - Acceptance testing dựa vào user requirements
 - System testing dựa vào functional requirements
 - Integration testing dựa vào system's architecture.

Mô hình chữ V



Mô hình chữ V



Requirements validation

- ❑ Là thành phần thứ tư – sau elicitation, analysis, và specification—của quy trình **phát triển yêu cầu**
- ❑ Mục đích của requirements validation là để bảo đảm:
 - SRS mô tả đúng các khả năng và đặc tính của hệ thống thỏa mãn nhu cầu của các stakeholder khác nhau
 - Yêu cầu phần mềm được suy dẫn đúng từ yêu cầu hệ thống và các quy tắc nghiệp vụ.
 - Yêu cầu là toàn diện và chất lượng cao.
 - Tất cả yêu cầu đều tương thích cùng nhau.
 - Yêu cầu cung cấp cơ sở đủ cho quy trình thiết kế và xây dựng và kiểm thử sản phẩm

Verification vs Validation

❑ Kiểm chứng (Verification)

- có đúng đặc tả không, có đúng thiết kế không
- phát hiện lỗi lập trình

❑ Thẩm định (Validation)

- có đáp ứng nhu cầu người dùng không
- có hoạt động hiệu quả không
- phát hiện lỗi phân tích, lỗi thiết kế (lỗi mức cao)

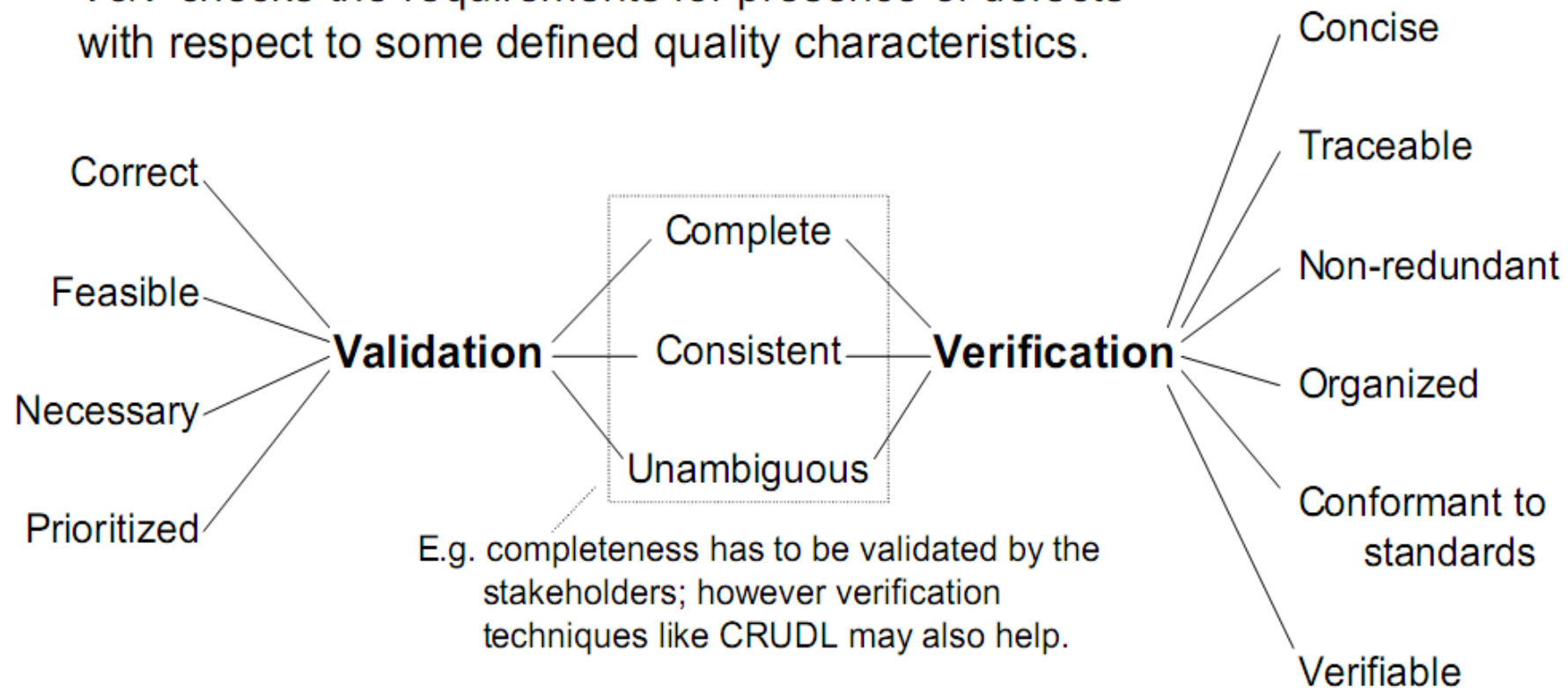
❑ V&V = Verification and Validation

- Mục tiêu là phát hiện và sửa lỗi phần mềm, đánh giá tính đúng đắn được của phần mềm

❑ Thứ tự thực hiện: Verification -> Validation

Validation vs. verification (2)

- V&V checks the requirements for presence of defects with respect to some defined quality characteristics.



- Henceforth, we will draw no distinction and call both “validation”.

Lợi ích của requirement validation

- ❑ Nhiều thành viên dự án rất miễn cưỡng khi phải tốn thời gian để “reviewing and testing an SRS”
- ❑ Lợi ích của validation:
 - Giảm việc phải làm lại (rework) do xác định yêu cầu sai
 - Tăng tốc việc tích hợp và kiểm thử hệ thống
 - Yêu cầu tốt hơn sẽ cho sản phẩm chất lượng hơn và thỏa mãn nhu cầu khách hàng cao hơn.
 - Giảm chi phí bảo trì, mở rộng.

Validation techniques

- ❑ Soát xét yêu cầu chính thức (formal) và phi chính thức,
- ❑ Các test cases (conceptual) được phát triển từ use cases và yêu cầu chức năng.

Reviewing Requirements

- ❑ Soát xét kỹ thuật (technical review): Là một ai đó không phải tác giả của một sản phẩm phần mềm (software work product) kiểm tra sản phẩm để tìm kiếm khiếm khuyết.
- ❑ Soát xét các tài liệu yêu cầu là kỹ thuật rất hữu hiệu để xác định các yêu cầu nhập nhằng, hoặc không thể kiểm tra, các yêu cầu không sáng sủa đủ để làm cơ sở cho việc thiết kế, và “yêu cầu” đó trên thực tế chính là các đặc tả thiết kế (design specifications).
- ❑ Soát xét yêu cầu cũng là một cách để các stakeholders thảo luận về mức độ hệ trọng của một chi tiết cụ thể nào đó.

Phân loại Software review

- ❑ Software peer reviews
- ❑ Software management reviews
- ❑ Software audit reviews

Đánh giá của chuyên gia

Peer review

- ❑ Peer review xảy ra khi ai đó không phải là người của đội phát triển dự án xem xét sản phẩm để xác định lỗi.
- ❑ Peer review được thực hiện bởi chính tác giả, một hay nhiều đồng nghiệp với tác giả cùng xem xét đánh giá sản phẩm để xác định lỗi.
- ❑ Việc khảo sát tài liệu SRS là 1 kỹ thuật rất hiệu quả để tìm xem yêu cầu nào còn mơ hồ hay không thể xác minh được. Yêu cầu nào chưa đủ rõ khi chuyển sang thiết kế và các vấn đề liên quan khác.
- ❑ Hai loại peer review:
 - Informal reviews
 - Formal review

Các phương pháp Informal peer review

- ❑ *Peer deskcheck*: khi nhờ 1 đồng nghiệp xem qua (look over) sản phẩm của mình.
- ❑ *Pass around*: mời nhiều đồng nghiệp cùng xem xét thành phẩm của mình
- ❑ *Walkthrough*: tác giả mô tả thành phẩm của mình và đưa ra các chú thích (comment) về nó.

Formal review

- ❑ Sau khi đánh giá, cần tạo ra báo cáo kết quả đánh giá và kết luận của đội xét duyệt xem sản phẩm có thể chấp nhận được không.
- ❑ Báo cáo chỉ rõ các khuyết điểm (defect) được tìm thấy và các vấn đề phát sinh. Thành viên của đội formal review chịu trách nhiệm về chất lượng của review, mặc dù người tạo ra sản phẩm phải là người cuối cùng chịu trách nhiệm về chất lượng.
- ❑ Formal peer review còn được gọi là *inspection* (thanh tra).

Inspection – Thanh Tra

- ❑ Bất kỳ một sản phẩm phần mềm nào cũng đều có thể kiểm tra, bao gồm tài liệu về yêu cầu và thiết kế, mã nguồn, tài liệu kiểm thử, và cả kế hoạch chung của dự án.
- ❑ Inspection là 1 quy trình nhiều giai đoạn (multistage process). Nó liên quan đến 1 đội nhỏ các thành viên đã qua đào tạo có trách nhiệm kiểm tra cẩn thận các khuyết điểm sản phẩm và các cơ hội cải thiện.
- ❑ Inspections là cửa ngõ mà các tài liệu cần phải đi qua trước khi nó trở thành **baseline (chuẩn)**.

Bốn dạng thành viên của đội inspection

- ❑ Tác giả (author) hay đồng tác giả của sản phẩm
- ❑ Tác giả của bất kỳ sản phẩm nào đã được tạo trước đó hay đặc tả cho các mục đang được kiểm tra đánh giá (phải bao gồm cả đại diện khách hàng).
- ❑ Những người sẽ làm việc dựa vào các đề mục đang được kiểm tra, có thể là developer, tester, project manager, và người viết tài liệu cho người dùng.
- ❑ Những người có trách nhiệm với sản phẩm, quan tâm đến các yêu cầu liên quan đến bên ngoài, hiệu quả khi yêu cầu thay đổi sẽ làm ảnh hưởng đến các hệ thống khác như thế nào.

Bốn dạng thành viên của đội inspection

- ❑ Nên giới hạn đội gồm 6 hay ít hơn các kiểm tra viên (inspector), có nghĩa là 1 vài lĩnh vực sẽ không có người trong đội đại diện.
- ❑ Các đội lớn sẽ dễ bị sa lầy vào các cuộc bàn luận ngoài lề, lo giải quyết vấn đề và tranh cãi xem cái gì thật sự lỗi
→ dẫn đến là giảm tốc độ duyệt tài liệu và tăng chi phí tìm lỗi.

Inspection Roles

- ☐ **Author**
- ☐ **Moderator**
- ☐ **Reader**
- ☐ **Recorder**

Author

- ❑ Tác giả là người tạo ra sản phẩm cần kiểm tra.
- ❑ Tác giả của SRS thường là analyst, người thu thập yêu cầu và viết đặc tả.
 - Trong informal review thì author thường điều khiển cuộc thảo luận.
 - Trong inspection, author giữ vai trò thụ động hơn. Author có thể lắng nghe các chú thích (comment) từ các inspector khác, trả lời nhưng không tranh cãi. Author cũng thường phát hiện lỗi mà các inspector khác không nhìn thấy.

Moderator – Người điều phối

- ❑ Còn gọi là *inspection leader*, lên kế hoạch kiểm tra cùng với author, điều phối các hoạt động, và tổ chức cuộc họp. Moderator phân phát tài liệu cho các thành viên nhiều ngày trước cuộc họp.
- ❑ Trách nhiệm của moderator bao gồm bắt đầu cuộc họp đúng giờ, động viên mọi thành viên góp ý, và giữ cho cuộc họp **tập trung việc tìm khuyết điểm thay vì tìm cách giải quyết**, báo cáo kết quả kiểm tra với ban quản lý, theo dõi các thay đổi cùng với tác giả để bảo đảm phát hiện được lỗi và các rắc rối

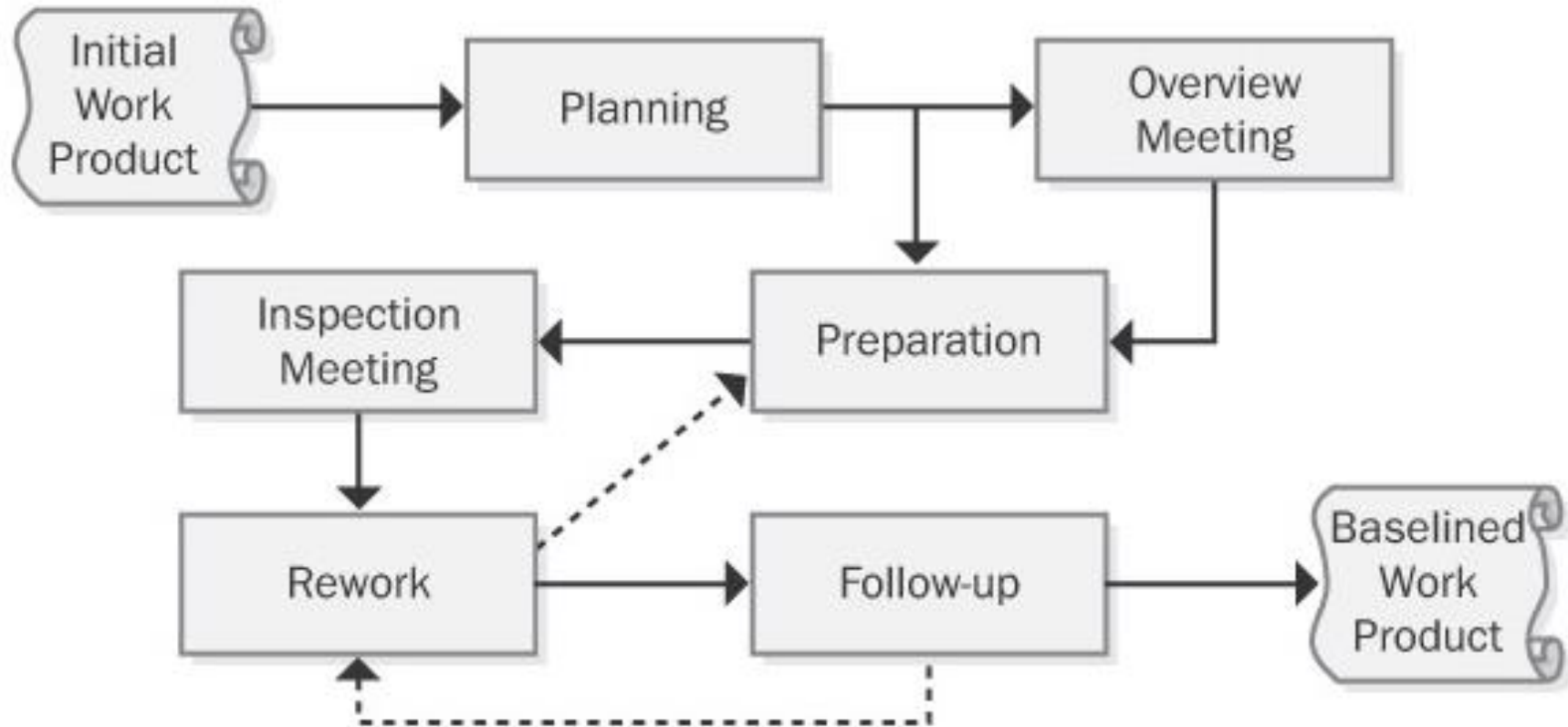
Reader – Người đọc

Trong các cuộc họp kiểm tra, reader sẽ đọc từng yêu cầu một trong SRS, các thành viên khác sẽ chỉ ra các khiếm khuyết và vấn đề cần xem xét

Recorder – Người ghi chép

- ❑ Recorder, hay *scribe*, ghi lại thành văn các vấn đề phát sinh và các khiếm khuyết được tìm thấy trong khi thanh tra theo một khuôn dạng văn bản chuẩn.
- ❑ Các thanh tra viên cần giúp người ghi chép nắm bắt được cốt lõi của mỗi vấn đề nảy sinh theo một cách thuyết phục để truyền thông cho tác giả về vị trí và bản chất của vấn đề.

Các giai đoạn của inspection



Các giai đoạn của inspection

- ❑ Mỗi inspector sẽ tự làm việc với tài liệu cần thẩm tra (inspected document) trong giai đoạn “preparation”. Đây là giai đoạn để phát hiện các vấn đề.
- ❑ Trong suốt cuộc họp, các inspectors sẽ thảo luận các vấn đề đã phát hiện được và có thể phát hiện thêm các vấn đề mới.

Inspection meeting

- ❑ Trong cuộc họp inspection, reader sẽ hướng dẫn với các inspector khác khi duyệt SRS, mô tả mỗi lần 1 yêu cầu. Khi các inspector đưa ra các lỗi và vấn đề, recorder ghi lại theo biểu mẫu tạo thành 1 danh sách các mục.
- ❑ Mục đích của cuộc họp nhằm xác định các lỗi chính trong tài liệu yêu cầu.
- ❑ Cuộc họp không nên kéo dài hơn 2 giờ vì khi mọi người mệt mỏi sẽ làm việc không còn hiệu quả. Nếu cần thêm thời gian thì nên lên kế hoạch cho 1 cuộc họp khác.
- ❑ Cuối cuộc họp, đội sẽ quyết định xem có chấp nhận tài liệu về yêu cầu hay không? Hay chấp nhận nhưng cần chỉnh sửa, nếu chỉnh sửa thì ít hay nhiều

Tiêu chuẩn đầu vào và tiêu chuẩn đầu ra (Entry and Exit Criteria)

- ❑ Điều kiện tiên quyết mà tài liệu về các yêu cầu cần phải thỏa mãn trước khi bắt đầu thẩm tra được gọi là *entry criteria*
- ❑ Các tiêu chuẩn cần phải được thỏa mãn trước khi moderator tuyên bố kiểm tra đã hoàn tất : **Exit criteria.**

Entry Criteria

1. Là 1 tập các tiêu chuẩn mà các author cần tuân theo trong lúc chuẩn bị inspection.
2. Giúp đội inspection không mất thời gian vào các vấn đề mà nên giải quyết trước khi bắt đầu inspection.
3. Moderator sử dụng entry criteria như một checklist trước khi quyết định có nên tiếp tục inspection hay không
4. Tài liệu phải phù hợp với các mẫu theo tiêu chuẩn
5. Tài liệu phải được kiểm tra chính tả.

Entry Criteria (tt)

6. Tác giả phải kiểm tra các lỗi bố trí (layout) trong tài liệu.
7. Các tài liệu tham khảo phải luôn sẵn sàng
8. Số dòng được in trong tài liệu để dễ tham chiếu đến các vị trí trong cuộc họp.
9. Tất cả các **open issues** cần phải đánh dấu như TBD (to be determined).
10. Moderator không tìm được nhiều hơn 3 lỗi chính trong cuộc khảo sát nhanh trong 10 phút của một tài liệu mẫu tiêu biểu.

Exit Criteria

1. Tất cả các vấn đề xuất hiện trong lúc kiểm tra đã được giải quyết
2. Bất kỳ thay đổi nào trong tài liệu đều đã được sửa đổi cẩn thận.
3. Tài liệu chỉnh sửa đã được kiểm tra chính tả.
4. Tất cả các TBDs đều đã được phân giải, hoặc quy trình phân giải mỗi TBD, ngày thực hiện, và người thực hiện đều được ghi nhận vào tài liệu.
5. Tài liệu đã được check in vào hệ thống quản lý cấu hình của dự án.
6. Các số liệu thanh tra đã được báo cáo tới người tổng hợp số liệu.

Các checklists thanh tra yêu cầu (Requirements Inspection Checklists)

- ❑ Để giúp các thanh tra viên tìm kiếm các kiểu lỗi trong sản phẩm mà họ phải thanh tra, bạn hãy xây dựng một checklist cho mỗi kiểu tài liệu mà tổ chức của bạn tạo ra.
- ❑ Các checklist đó lôi kéo sự chú ý của các thanh tra viên vào các vấn đề về yêu cầu vẫn thường xảy ra trong quá khứ.

Các checklists thanh tra yêu cầu (Requirements Inspection Checklists)

Stt	Tiêu chuẩn	Đáp ứng
KẾT CẤU CỦA SRS VÀ TÍNH ĐẦY ĐỦ		
1	Tất cả các tham chiếu chéo nội bộ tới các yêu cầu khác là đúng đắn?	
2	Tất cả các yêu cầu đều được mô tả ở cùng một mức chi tiết nhất quán và thích hợp?	
3	Các yêu cầu tạo cơ sở đầy đủ cho việc thiết kế?	
4	Mức ưu tiên thực thi của mỗi yêu cầu đều được xác định?	
5	Tất cả các giao diện phần cứng, phần mềm và truyền thông đều được định nghĩa?	
6	Các thuật toán để cài đặt các yêu cầu chức năng đều được định nghĩa?	
7	SRS đã mô tả tất cả các khách hàng hoặc hệ thống đã biết cần thiết của phần mềm?	
8	Một yêu cầu bị khuyết thông tin nào đó? Nếu đúng như vậy hãy đánh dấu yêu cầu đó là TBD.	
9	Hành vi được kỳ vọng đã được mô tả đầy đủ cho tất cả các điều kiện lỗi tiên đoán?	

Các checklists thanh tra yêu cầu (Requirements Inspection Checklists)

TÍNH ĐÚNG ĐẮN		
10	Có những yêu cầu nào xung đột với nhau hay không, hay mô tả cùng một nội dung hay không?	
11	Mỗi yêu cầu đều được viết bằng một ngôn ngữ sáng sủa, chính xác, không nhập nhằng?	
12	Mỗi yêu cầu đều có thể được kiểm tra bằng kiểm thử, chứng minh, soát xét hoặc phân tích?	
13	Mỗi yêu cầu đều thuộc phạm vi của dự án?	
14	Mỗi yêu cầu đều không có lỗi về nội dung hoặc lỗi về ngữ pháp?	
15	Có thể tất cả các yêu cầu đều được thi công trong các ràng buộc đã biết?	
16	Tất cả các messages báo lỗi cụ thể đều là duy nhất và có ý nghĩa?	

Các checklists thanh tra yêu cầu (Requirements Inspection Checklists)

CÁC THUỘC TÍNH CHẤT LƯỢNG		
17	Tất cả các mục tiêu hiệu năng đều được đặc tả một cách hợp lý?	
18	Tất cả các cân nhắc về an ninh và an toàn đều được đặc tả một cách hợp lý?	
19	Tất cả các mục tiêu chất lượng khác đều được mô tả và định lượng sáng sủa, với các đánh đổi có thể chấp nhận và được đặc tả?	
KHẢ NĂNG LẤN VẾT		
20	Mỗi yêu cầu đều được định danh đúng đắn và duy nhất?	
21	Mỗi yêu cầu chức năng đều được lần vết tới một yêu cầu mức cao hơn? (Ví dụ yêu cầu hệ thống hoặc use case)	
CÁC VẤN ĐỀ ĐẶC BIỆT		
22	Tất cả các yêu cầu đang có đều không kèm theo các giải pháp thiết kế hoặc thi công?	
23	Tất cả các chức năng then chốt về mặt thời gian (time-critical functions) đều được định danh và tiêu chuẩn thời gian được mô tả rõ?	
24	Các vấn đề quốc tế hoá đều được xác định rõ?	
Hình 14-4. Checklist thanh tra tài liệu SRS		

Các checklists thanh tra yêu cầu (Requirements Inspection Checklists)

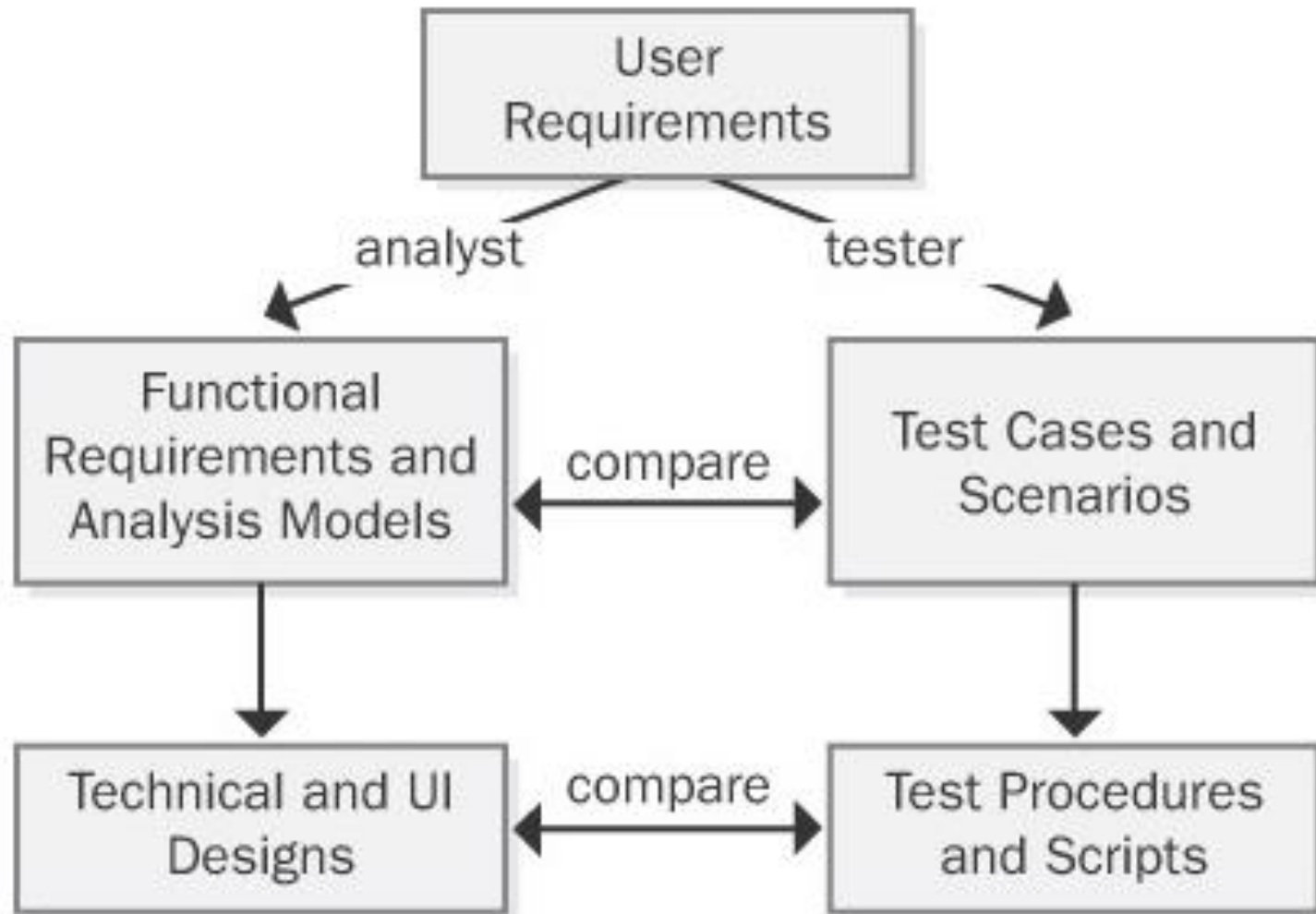
Stt	Tiêu chuẩn	Đáp ứng
1	Mỗi use case là một tác vụ độc lập và tách rời?	
2	Mục tiêu hoặc giá trị có thể đo lường của use case được xác định rõ?	
3	Actor tương tác với use case được xác định rõ?	
4	Use case được mô tả ở mức cơ bản (trừu tượng), thay vì như một kịch bản cụ thể?	
5	Use case không xác định các chi tiết thiết kế và thi công?	
6	Tất cả các tiến trình thay thế dự đoán (anticipated alternative courses) đều được mô tả đầy đủ?	
7	Tất cả các điều kiện loại trừ đã biết (known exception conditions) đều được mô tả đầy đủ?	
8	Có chuỗi hành động chung nào có thể được tách ra từ các use cases? (Để hình thành use case mới)	
9	Chuỗi đối thoại của mỗi tiến trình được mô tả sáng sủa, không nhập nhằng và đầy đủ?	
10	Mỗi actor và bước trong use case đều thích hợp để thực thi tác vụ?	
11	Mỗi tiến trình được định nghĩa trong use case đều khả thi?	
12	Mỗi tiến trình được định nghĩa trong use case đều có thể kiểm tra?	
Hình 14-5. Checklist thanh tra tài liệu use-case		

Requirements Review Challenges

- ❑ Những thách thức mà một tổ chức phải đương đầu khi duyệt các tài liệu về yêu cầu.
 - Các tài liệu yêu cầu lớn (Large requirements documents)
 - Các nhóm thanh tra lớn (Large inspection teams)
 - Sự xa cách về địa lý giữa các thanh tra viên (Geographic separation of inspectors)

Kiểm Thử Yêu Cầu (Testing Requirements)

- ❑ Khó hình dung hệ thống sẽ vận hành thế nào nếu chỉ đọc SRS.
- ❑ Test cases dựa vào yêu cầu chức năng hay suy diễn từ yêu cầu người dùng để tạo ra các hành vi hệ thống.
- ❑ Các test case thiết kế đơn giản cũng giúp phát hiện nhiều vấn đề về yêu cầu ngay cả khi không thực hiện các test trên hệ điều hành.
- ❑ Nếu bắt đầu xây dựng các test case sớm ngay khi 1 số yêu cầu đã ổn định, có thể tìm ra các vấn đề giúp sửa sai mà không quá tốn kém



Trường hợp kiểm thử ý niệm (Conceptual test cases)

- ❑ Tạo ra các test cases từ use cases hay từ các đại diện yêu cầu người dùng ngay từ đầu .
- ❑ Các test case được dùng để đánh giá các yêu cầu, mô hình phân tích và prototype - nên bao phủ ngữ cảnh thông thường và các ngoại lệ trong kịch bản use case.
- ❑ Các test cases độc lập với thực thi.

Ví dụ conceptual test cases

- ❑ Khảo sát use case "View Order" của hệ thống Chemical Tracking.

Ví dụ conceptual test cases

❑ Các test cases như sau:

- User enters order number to view, **order exists**, user placed the order. Expected result: show order details.
- User enters order number to view, **order doesn't exist**. Expected result: Display message "Sorry, I can't find that order."
- User enters order number to view, **order exists**, **user didn't place the order**. Expected result: Display message "Sorry, that's not your order."

Tạo test case cho use case

- ❑ Test case để kiểm thử use case: vì mỗi use case có 1 kịch bản chính và các kịch bản phụ, các ngoại lệ (exception) nên có nhiều execution path → tạo các test case khác nhau để kiểm thử
- ❑ Các test case này đều trừu tượng, độc lập với thực thi, chỉ đến khi chuyển sang thiết kế giao diện các **test case** trừu tượng này sẽ được định nghĩa thành các **thủ tục test (test procedure)** cụ thể.

Ví dụ về requirement testing

❑ Xét **Business requirement** sau:

CTS sẽ giảm bớt chi phí mua sắm hóa chất bằng cách thúc đẩy việc sử dụng lại các công-ten-nơ hóa chất đã có sẵn trong công ty.

❑ **Use case.** Một use case yêu cầu kinh doanh trên là “**Đề xuất một hóa chất**” (Request a chemical), use case bao hàm một chuỗi hành động cho phép người dùng đề xuất một công-ten-nơ hóa chất đã có sẵn trong kho hóa chất

Yêu cầu chức năng của use case

"Request a Chemical"

- ❑ Requester đặc tả hóa chất đề xuất, hoặc bằng cách nhập chemical ID number hoặc bằng cách import cấu trúc của nó từ một công cụ vẽ cấu trúc hóa chất.
- ❑ Hệ thống có thể đáp ứng request này theo 2 lựa chọn:
 - Đưa ra cho Requester một công-ten-nơ mới
 - Hoặc công-ten-nơ đã sử dụng rồi chứa hóa chất đó từ chemical stockroom hoặc bằng cách cho phép Requester đặt hàng từ một nhà cung cấp bên ngoài.

Yêu cầu chức năng của use case "Request a Chemical"

- ❑ **Functional requirement:** cho phép người dùng lựa chọn một hóa chất đã có thay vì đặt hàng một hóa chất đã có từ bên ngoài.
- ❑ *Nếu stockroom có các công-ten-nơ hóa chất đang được yêu cầu, hệ thống phải hiển thị một danh sách các công-ten-nơ sẵn sàng đó.*
- ❑ *Người dùng hoặc là lựa chọn một công-ten-nơ, hoặc đề nghị đặt hàng mua một công-ten-nơ mới.*

Use case “Request a Chemical”

Use case ID: USE CASES-5	
Tên use case: Request a Chemical	
Người tạo: Tim	Người cập nhật lần cuối: Janice
Ngày tạo: 4/10	Ngày cập nhật lần cuối: 27/10
Actor: Requester	
<p>Mô tả: Requester đặc tả hóa chất đề nghị, hoặc nhập chemical ID number của hóa chất đó, hoặc nhập cấu trúc của hóa chất từ một công cụ vẽ cấu trúc hoá chất. Hệ thống có thể đáp ứng đề nghị hoặc bằng cách đưa ra cho Requester một hộp hoá chất mới, hoặc một hộp chứa hoá chất dùng rồi mà trong kho đã có sẵn hoặc một thông báo Requester tự đặt hàng từ một nhà cung cấp bên ngoài.</p>	
<p>Tiền tình huống:</p> <ol style="list-style-type: none"> 3. User ID được cấp quyền. 4. CSDL kho hóa chất được trực tuyến. 	
<p>Hậu tình huống:</p> <ol style="list-style-type: none"> 3. Các đề nghị được lưu trữ đầy đủ trong Chemical Tracking System 4. Đề nghị được gửi qua email tới Chemical Stockroom hoặc Bộ phận mua sắm để thực hiện. 	
Ưu tiên: Cao	
<p>Tần suất sử dụng: Xấp xỉ 5 lần 1 tuần trên mỗi nhà hoá học, 100 lần mỗi tuần trên mỗi nhân viên quản lý kho hóa chất.</p>	

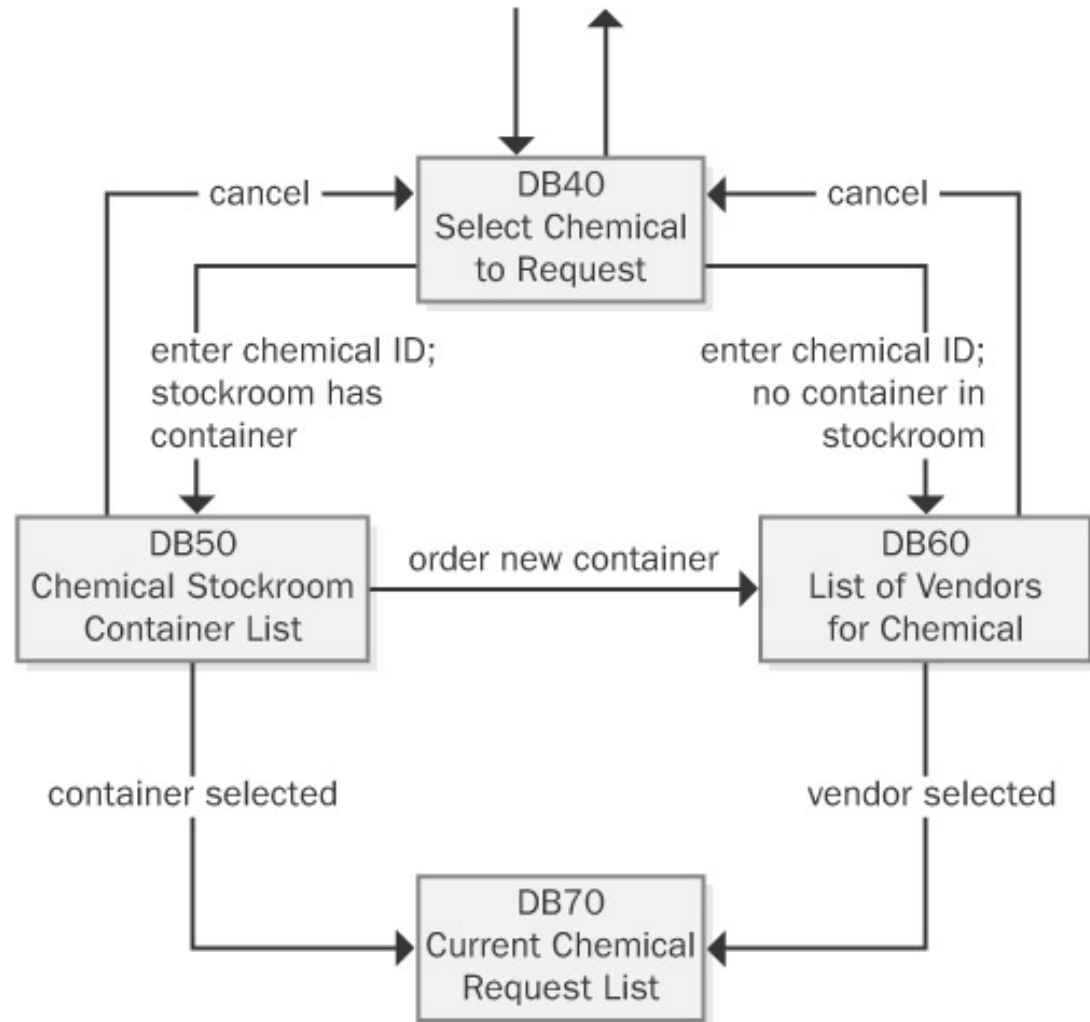
Use case

“Request a Chemical”

Tiến trình chuẩn: 5.0 Request a Chemical from a Vendor	
Hành động của Actor	Đáp ứng của hệ thống
1. Nhập Chemical ID number hoặc tên của file chứa cấu trúc hoá học	2. Kiểm tra tính hợp lệ của Chemical ID
4. Xác định nhà cung cấp (tiếp tục) hoặc kho hoá chất	3. Hỏi Requester là muốn đặt hàng từ một nhà cung cấp mới hoặc muốn một hộp hoá chất từ kho
	5. <tiếp tục đối thoại cho đến khi đề nghị được thực hiện>
Tiến trình thay thế: 5.1 Request a Chemical from the Chemical Stockroom (rẽ nhánh kể từ 5.0.4)	
Hành động của Actor	Đáp ứng của hệ thống
2. Tùy chọn (optionally), yêu cầu cho xem tình hình (history) của bất kỳ hộp hoá chất nào	1. Hiện thị một danh sách các hộp chứa hóa chất mong muốn còn ở trong kho.
3. Chọn một hộp cụ thể hoặc yêu cầu đặt hàng từ một nhà cung cấp	
Các loại trừ: 5.E.1 Chemical Is Not Commercially Available	
Hành động của Actor	Đáp ứng của hệ thống
3. Đề nghị một hoá chất khác.	1. Hiện thị message: Không Nhà cung cấp. 2. Hỏi Requester liệu có đề nghị một hoá chất khác hoặc thoát khỏi chương trình. 4. Bắt đầu lại Tiến trình Chuẩn.
Includes: UC-12 Enter Change Number	
Yêu cầu đặc biệt: Hệ thống có thể nhập một cấu trúc hoá học theo một khuôn dạng đã được mã hoá chuẩn từ bất cứ thiết bị vẽ cấu trúc hoá học nào.	
Giả định: Cấu trúc hoá học được nhập vào hệ thống được giả định hợp lệ	
Ghi chú: Tim sẽ tìm hiểu xem liệu sự chấp thuận của cấp quản lý có cần thiết khi một hoá chất trên mức nguy hiểm cấp 1 được đề nghị hay không. Hạn ngày 4/11 (Due date).	

Dialog map for the "Request a Chemical" use case

❑ Có nhiều
Execution path
trong use case
→ Có nhiều
Execution path
cần kiểm thử



Tạo các Test case

- ❑ Vì Use case này có nhiều cách thực thi (execution paths) tương ứng với kịch bản chính và ngoại lệ của use case, nên có nhiều test case
- ❑ Ví dụ: tạo 1 test case cho path hiển thị cho người dùng các bình hóa chất hiện có trong kho.

Test case mẫu 1

- ❑ Test case này được suy ra từ các use-case description của tác vụ và từ dialog map trong Hình 14-6.
- ❑ *Tại dialog box DB40, nhập một chemical ID hợp lệ; chemical stockroom có hai công-ten-nơ của hóa chất này. Dialog box DB50 xuất hiện thể hiện số lượng 2 công-ten-nơ. Lựa chọn công-ten-nơ thứ hai. DB50 đóng lại và côngten- nơ2 được thêm vào cuối của Current Chemical Request List trong DB70*

Thực hiện Kiểm thử yêu cầu chức năng

Gồm 2 bước:

- Kiểm tra test case có đáp ứng yêu cầu chức năng hay không?
- Kiểm tra test case có bao phủ các path trong dialog map không?

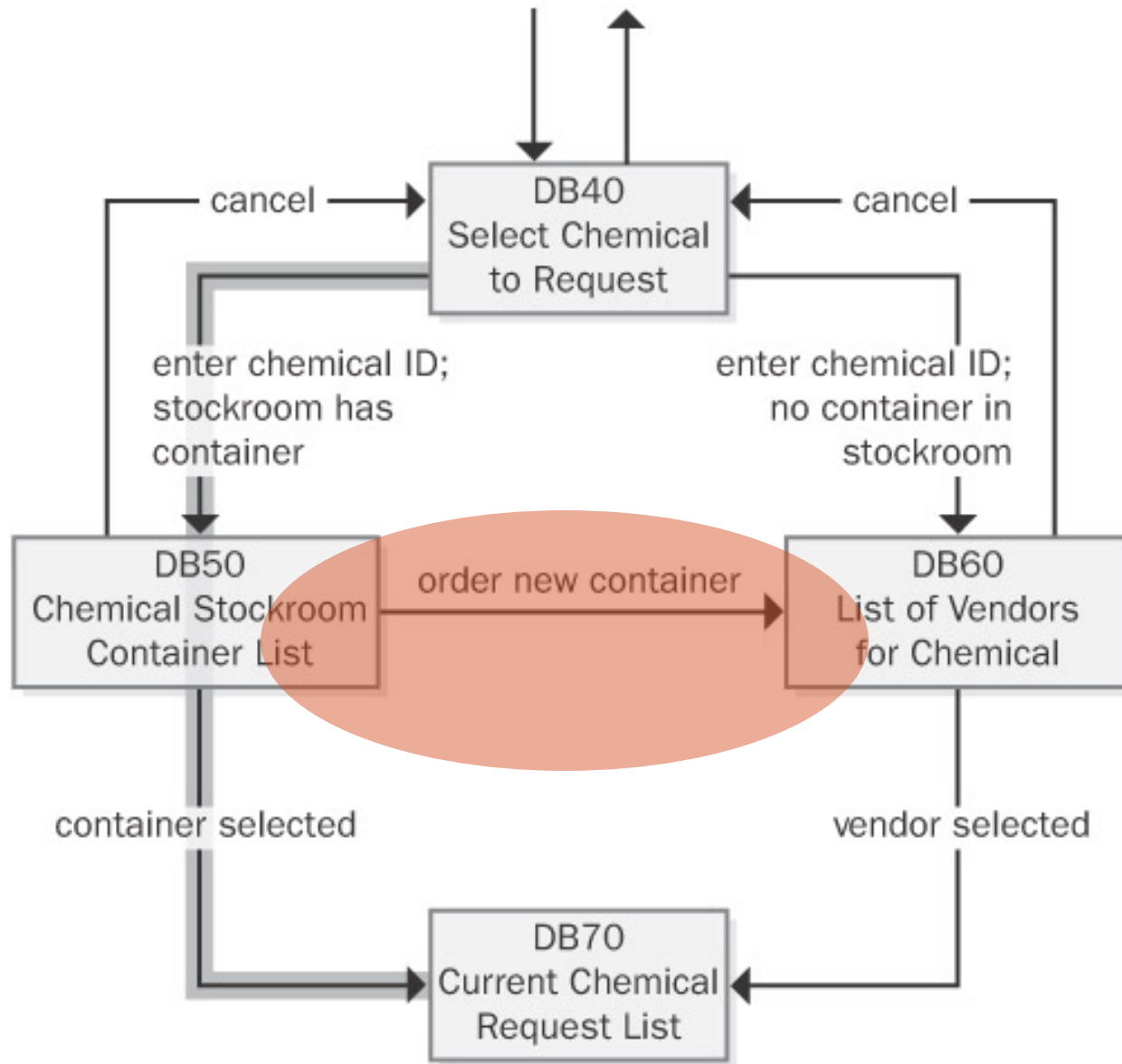
Kiểm thử yêu cầu chức năng

- ❑ Ánh xạ mỗi test case với các yêu cầu chức năng để bảo đảm mỗi test case đều có thể được thực thi bởi 1 tập các yêu cầu chức năng.
- ❑ Bảo đảm là mỗi yêu cầu chức năng phải có ít nhất 1 test case

Kiểm thử yêu cầu chức năng

- ❑ Dùng bút đánh dấu các execution path tương ứng với mỗi test case trong dialog map.
- ➔ Nhờ đó có thể phát hiện được các yêu cầu bị thiếu hay không đúng ➔ sửa lỗi và tinh chỉnh lại test case cho phù hợp.

Execution path



Ví dụ test case không đúng/bị thiếu

- ❑ Giả sử sau khi thực thi tất cả test cases và đã dùng bút đánh dấu vào dialog map thì phát hiện thấy đường có nhãn "order new container" đi từ DB50 đến DB60 không được đánh dấu.
- ➔ Có thể hiểu theo 2 cách:

Ví dụ test case không đúng/bị thiếu

1. Đường nối từ DB50 đến DB60 không phải là hành vi hệ thống (system behavior)

→ Analyst cần xóa nó khỏi dialog map. Nếu SRS có chứa yêu cầu liên quan đến thì yêu cầu này cũng phải loại bỏ

2. Đường nối này là 1 hành vi hệ thống (system behavior) hợp lệ

→ Thiếu test case mô tả hành vi này, cần bổ sung thêm.

Test case mẫu 2

- ❑ Thử tạo test case cho trường hợp “the user can take some action to move directly from dialog box DB40 to DB70”
- ❑ Trong dialog map không có execution path nào nối từ DB40 đến DB70 vì vậy test case này không thể thực thi được.

Có thể hiểu theo 2 cách

Lý do test case bị sai

- ❑ Việc chuyển từ hộp thoại DB40 đến DB70 **không phải là hành vi hệ thống (system behavior)** vì vậy test case bị sai.
- ❑ Việc chuyển từ DB40 đến DB70 là chức năng hợp lệ nhưng dialog map và có thể cả SRS thiếu 1 yêu cầu nào đó cho phép thực thi test case này.

Bài tập: Viết Test Case từ Use Case Login

Luồng cơ sở Login.

- Người dùng nhập Username và Password
- Nhấn nút Đăng Nhập
- Hệ thống kiểm tra Username và Password
- Hệ thống thông báo đăng nhập thành công.
- Kết thúc use case, có thể sử dụng các chức năng của chương trình

Luồng ngoại cơ sở

- A1 - Người dùng nhập sai Username và password: Nếu người dùng nhập sai Username hoặc Password thì hệ thống thông báo lỗi.
- A2 - Người dùng không nhập Username và Password: Nếu người dùng không nhập Username hoặc Password thì hệ thống hiển thị nhắc nhở cho người dùng.
- A3 - Người dùng thoát trong quá trình đăng nhập: Nếu người dùng thoát trong quá trình đăng nhập thì quá trình đăng nhập bị hủy bỏ và kết thúc Use case.
- A4 - Cơ sở dữ liệu trục trặc: Nếu cơ sở dữ liệu trục trặc hệ thống hiển thị lỗi và kết thúc use case.

Bài tập: Viết Test Case từ Use Case Login

❑ Các kịch bản

KB1	Luồng cơ sở			
KB2	Luồng cơ sở	A1		
KB3	Luồng cơ sở	A1	A3	
KB4	Luồng cơ sở	A1	A4	
KB5	Luồng cơ sở	A1	A4	A3
KB6	Luồng cơ sở	A2		
KB7	Luồng cơ sở	A2	A3	
KB8	Luồng cơ sở	A3		
KB9	Luồng cơ sở	A4		
KB10	Luồng cơ sở	A4	A3	

Bài tập: Viết Test Case từ Use Case Login

❑ Các kịch bản

B1 : Sinh đầy đủ các kịch bản của use-case

TÊN KỊCH BẢN	BẮT ĐẦU LUỒNG	NGOÀI LUỒNG
Đăng nhập thành công	LUỒNG CƠ SỞ	
Sai Username và password:	LUỒNG CƠ SỞ	A1
Không nhập Username và Password	LUỒNG CƠ SỞ	A2
Thoát	LUỒNG CƠ SỞ	A3
CSDL trực trặc	LUỒNG CƠ SỞ	A4

Bài tập: Viết Test Case từ Use Case Login

B2 : Xác định test-case

Test case ID	Scenario Condition	Username	Password	Expected result
RC 1	Scenario 1 - successful login	V	V	Confirm and can use functions
RC 2	Scenario 2 -Invalid ID	I	V	Error message: back to login screen
RC 3	Scenario 3 - Invalid Password	V	I	Error message: back to login screen
RC 4	Scenario 4 - Invalid user	I	I	Error message: back to login screen
RC 5	Scenario 5 - unidentified ID	Empty	N/A	Error message: back to login screen
RC 6	Scenario 6 - unidentified password	V	empty	Error message: back to login screen
RC 7	Scenario 7 - unidentified user	Empty	empty	Error message: back to login screen
RC 8	Scenario 8 - user quits	N/A	N/A	Error message: back to previous screen
RC9	Scenario 9- system unavailable	N/A	N/A	Error message: "Lỗi CSDL không truy cập được"

Bài tập: Viết Test Case từ Use Case Login

B3 : Xác định giá trị dữ liệu

Test case ID	Scenario Condition	Username	Password	Expected result
RC 1	Scenario 1 - successful login	Admin	admin	Confirm and can use functions
RC 2	Scenario 2 - Invalid ID	Hsagdah	admin	Error message: back to login screen
RC 3	Scenario 3 - Invalid Password	Admin	Dsadasd	Error message: back to login screen
RC 4	Scenario 4 - Invalid user	Jhdsfj	Fdsjfhsk	Error message: back to login screen
				screen
RC 5	Scenario 5 - unidentified ID		N/A	Error message: back to login screen
RC 6	Scenario 6 - unidentified password	Admin		Error message: back to login screen
RC 7	Scenario 7 - unidentified user			Error message: back to login screen
RC 8	Scenario 8 - user quits	N/A	N/A	Error message: back to previous screen
RC 9	Scenario 9- system unavailable	N/A	N/A	Error message: "Lỗi CSDL không truy cập được"

Bài tập: Viết Test Case từ Use Case Login

Viết Test Case

<u>TC</u>	<u>Các bước</u>	<u>Dữ liệu vào</u>	<u>Kết quả mong đợi</u>	<u>Kết quả thực tế</u>	<u>P/F</u>
1.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username và nhấn nút đăng nhập.	User Name= admin, không nhập Password	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
2.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập password và nhấn nút đăng nhập.	Password= admin, , không nhập username	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
3.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = admin AND Password = XYZ	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
4.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = XYX AND Password = admin	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
5.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = XYZ AND Password = XYZ	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
6.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn	USER = " " AND Password = " "	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật		

Bài tập: Viết Test Case từ Use Case Login

	nút đăng nhập.		khẩu!		
7.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = " " AND Password = admin	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
8.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = admin AND Password = " "	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
9.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = xyz AND Password = " "	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
10.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = " " AND Password = xyz	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
11.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = Admin AND Password = ADMIN	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		
12.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = Admin AND Password = Admin	Phải hiển thị thông báo Bạn phải cung cấp chính xác mã người dùng và mật khẩu!		

Bài tập: Viết Test Case từ Use Case Login

13.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = Admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
14.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = aDmin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
15.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = adMin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
16.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn	USER = admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		

Bài tập: Viết Test Case từ Use Case Login

	nút đăng nhập.				
17.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
18.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = Admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
19.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = Admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
20.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = Admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
21.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = Admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
22.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = ADMIN AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
23.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập	USER = admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		

Bài tập: Viết Test Case từ Use Case Login

24.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập.	USER = admin AND Password = admin	Phải đăng nhập vào được và sử dụng được các chức năng		
25.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập Nhập username, password và nhấn nút đăng nhập với csdl trực trực	Username và password bất kỳ	Phải hiển thị thông báo Lỗi CSDL không truy cập được		
26.	Từ main form chọn chức năng đăng nhập để hiển thị form đăng nhập		Trở về màn hình trước đó		
	Nhấn nút thoát				

Kiểm thử yêu cầu và nghiệm thu từ khách hàng

- ❑ **Kiểm thử yêu cầu (Conceptual testing)** là 1 kỹ thuật rất hiệu quả để kiểm soát chi phí và thời gian của dự án bằng cách phát hiện những yêu cầu mơ hồ hay bị lỗi ngay từ giai đoạn đầu
- ❑ **Nghiệm thu từ khách hàng (acceptance testing)** là để xác định xem hệ thống có thỏa mãn các tiêu chuẩn có thể chấp nhận được của khách hàng (*acceptance criteria*) hay không?

Acceptance Testing

- ❑ Acceptance testing chỉ được thực thi khi hệ thống đã hoàn tất.
- ❑ Nên thực hiện các test này trong môi trường càng giống với môi trường thực của người dùng sau này
- ❑ Kết quả của test chỉ có 2 giá trị pass or fail, không nên đánh giá theo kiểu success or failure.
- ❑ Người dùng của hệ thống sẽ thực hiện các test này.
- ❑ Các developers sẽ suy dẫn ra các test này từ SRS đã phê duyệt của hệ thống.

Acceptance Testing

Các mục tiêu kiểm tra:

- ❑ Kiểm tra các quy trình xử lý công việc trên hệ thống đã được tích hợp đầy đủ nhất.
- ❑ Các quy trình hoạt động và bảo trì
- ❑ Các thủ tục người dùng (ví dụ: phân quyền dựa trên user login)
- ❑ Các form (ví dụ, các màn hình nhập liệu)
- ❑ Các report (ví dụ các bản báo cáo như phiếu thu để in, báo cáo doanh thu...)

Acceptance Testing

Kiểm thử dựa vào:

- ❑ Các yêu cầu người dùng (User requirements)
- ❑ Các yêu cầu hệ thống (System requirements)
- ❑ Các trường hợp sử dụng (Use cases)
- ❑ Các quy trình xử lý công việc (Business processes)
- ❑ Các báo cáo phân tích rủi ro (Risk analysis reports)

Lý do cần kiểm thử chấp nhận

- ❑ Nhà phát triển phần mềm luôn tin vào sản phẩm mình xây dựng là hoàn chỉnh, nhưng khách hàng mới là người đánh giá cuối cùng
- ❑ Khách hàng thực hiện các kiểm thử chấp nhận (acceptance testing) để xác định xem hệ thống có thỏa mãn các acceptance criteria của họ hay không?
 - Nếu không thỏa mãn, khách hàng có thể chuyển sang một hệ thống khác

Acceptance Criteria

- ❑ Acceptance criteria dùng để đánh giá sản phẩm có thỏa mãn các yêu cầu đã đề ra hay không và nó có phù hợp để sử dụng trong hoàn cảnh thực của tổ chức hay không.
- ❑ Để người dùng đề ra các acceptance test là 1 chiến lược phát triển yêu cầu có hiệu quả. Người dùng viết ra các acceptance tests càng sớm thì họ càng sớm phát hiện được các khiếm khuyết (defect) trong yêu cầu.

Acceptance Criteria

- ❑ Nên tập trung vào viễn cảnh sẽ sử dụng phần mềm như thế nào
- ❑ Người dùng chính nên xem xét use case nào là sử dụng nhiều nhất và quan trọng nhất khi quyết định xem nên đánh giá khả năng chấp nhận phần mềm.
- ❑ Acceptance tests tập trung vào các trường hợp thông thường của use case, không nên để ý quá đến các trường hợp ít thông dụng hay vào việc hệ thống quản lý mọi điều kiện ngoại lệ có đúng không.

Acceptance Criteria

- ❑ Nên để acceptance test tự động khi có thể.
- ❑ Acceptance tests cũng nên xét cả các yêu cầu phi chức năng, bảo đảm mục tiêu có thể thực thi trên mọi platform, hệ thống phù hợp với các tiêu chuẩn và tất cả mọi yêu cầu đã cam kết đều được thực thi.

Acceptance Criteria

- ❑ Viết ra yêu cầu không chưa đủ, mà cần phải bảo đảm là yêu cầu phải đúng và đủ chi tiết để dùng cho thiết kế, xây dựng, kiểm thử và quản lý dự án.
- ❑ Lập kế hoạch cho acceptance test, rà soát lại SRS, và các kỹ thuật kiểm thử yêu cầu sẽ giúp xây dựng hệ thống chất lượng, nhanh hơn và ít đắt hơn những dự án trước đó.

Assignment

- ❑ Chọn ngẫu nhiên một trang mô tả yêu cầu chức năng từ SRS của bạn. Triệu tập mọi người thành một nhóm nhỏ đại diện cho các stakeholders khác nhau và kiểm tra cẩn thận yêu cầu để tìm bất cứ một độ lệch nào so với các đặc tính của lời phát biểu yêu cầu tuyệt vời (excellent requirement statements).
- ❑ Nếu bạn tìm đủ các vấn đề bằng cách soát xét ngẫu nhiên khiến cho những người tham gia soát xét thấy lo lắng về chất lượng tổng thể của yêu cầu thì hãy thúc đẩy người dùng và những người đại diện phát triển thanh tra toàn bộ SRS. Đào tạo các thành viên của nhóm về quy trình thanh tra để công việc đạt hiệu quả cao nhất