FPT Education

FPT POLYTECHNIC

QUẢN LÝ DỰ ÁN VỚI AGILE

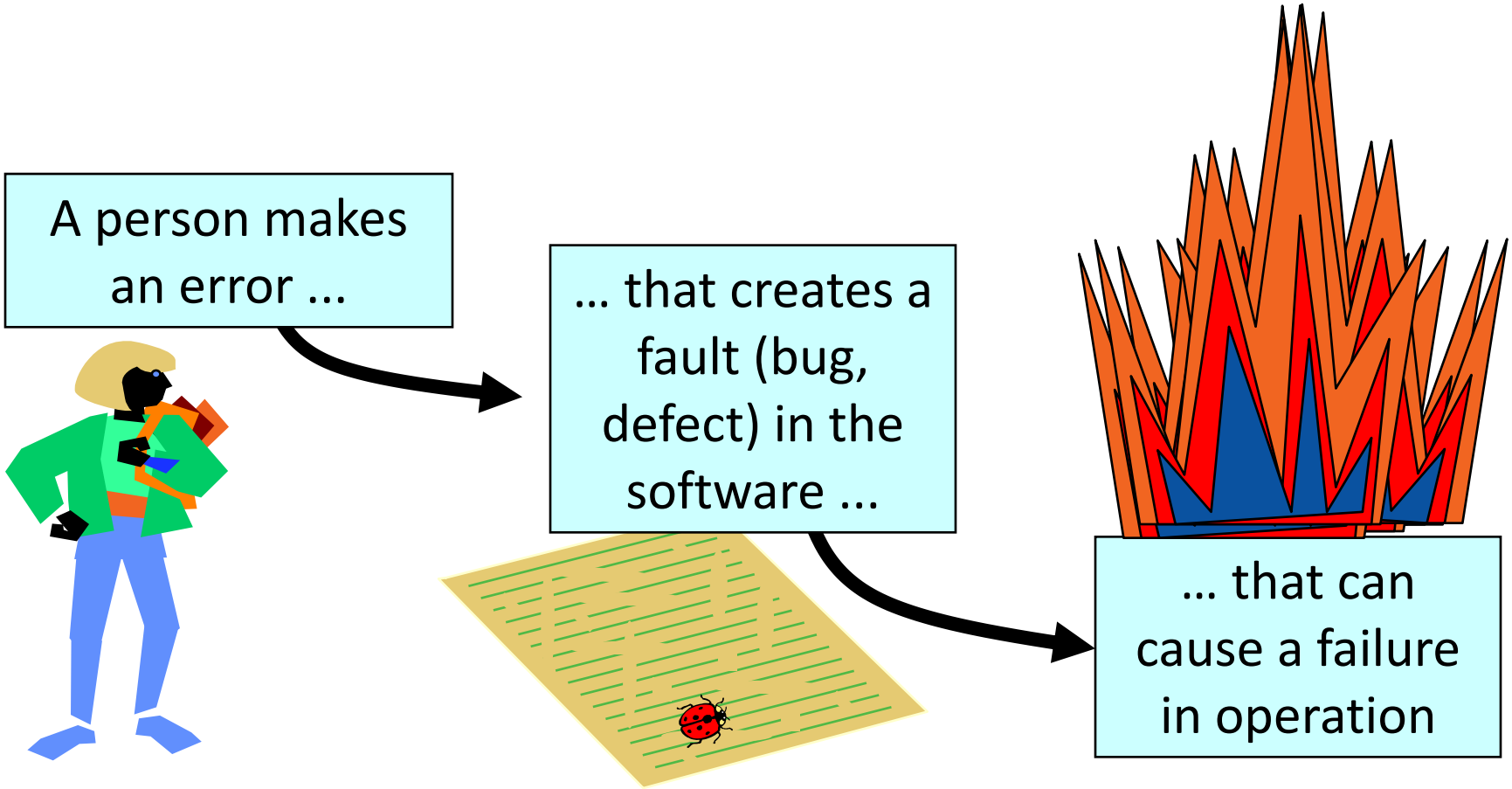
Bài 6: Kiểm thử

Nội dung bài học

- Khái niệm kiểm thử phần mềm
- Một số đặc điểm của kiểm thử phần mềm
- Tại sao kiểm thử lại cần thiết?
- Quy trình kiểm thử
- Các mức độ kiểm thử
- Vai trò của Tester
- Công việc Tester
- Tổ chức cơ sở hạ tầng kiểm thử
- Định nghĩa hoàn thành

Khái niệm kiểm thử phần mềm

- Kiểm thử là gì?



Khái niệm kiểm thử phần mềm

- Kiểm thử phần mềm là quá trình thực thi phần mềm với mục tiêu tìm ra lỗi (Glen Myers, 1979)
- Khẳng định được chất lượng của phần mềm đang xây dựng (Hetzel, 1988)

Một số đặc điểm kiểm thử phần mềm

- Kiểm thử phần mềm giúp tìm ra được sự hiện diện của lỗi nhưng không thể chỉ ra sự vắng mặt của lỗi (Dijkstra)
- Mọi phương pháp được dùng để ngăn ngừa hoặc tìm ra lỗi đều sót lại những lỗi khó phát hiện hơn (Beizer)
- Điều gì xảy ra nếu việc kiểm thử không tìm được lỗi trong phần mềm hoặc phát hiện quá ít lỗi:
  - Phần mềm có chất lượng quá tốt
  - Quy trình/Đội ngũ kiểm thử hoạt động không hiệu quả

Tại sao kiểm thử lại cần thiết?

- Thông thường thì phần mềm không hoạt động như mong muốn → lãng phí tiền bạc, thời gian, uy tín của doanh nghiệp, thậm chí có thể gây nên thương tích hay cái chết
- Ví dụ:
  - Một phần mềm tính toán lượng thuốc trừ sâu dùng cho cây trồng, vì lý do tính sai số lượng lên gấp 10 lần → Nông dân phải bỏ nhiều tiền mua, cây trồng hư hại, môi trường sống, nguồn nước bị ảnh hưởng,....
  - Website công ty có nhiều lỗi chính tả trong câu chữ -→Khách hàng có thể lảng tránh công ty với lý do công ty có vẻ không chuyên nghiệp

Tại sao kiểm thử lại cần thiết?

- Kiểm thử phần mềm → chất lượng phần mềm được nâng cao
- Chúng ta có thể đánh giá chất lượng phần mềm dựa vào số lượng lỗi tìm thấy và các đặc tính như: tính đúng đắn, tính dễ sử dụng, tính dễ bảo trì,...
- Kiểm thử có thể đem lại sự tin tưởng đối với chất lượng phần mềm nếu có ít lỗi hoặc không có lỗi nào được tìm thấy. Nếu lỗi tìm thấy và được sửa thì chất lượng phần mềm càng được tăng ➔ Giảm chi phí trong quá trình phát triển, nâng cấp, bảo trì phần mềm

Lỗi tăng lên khi nào?

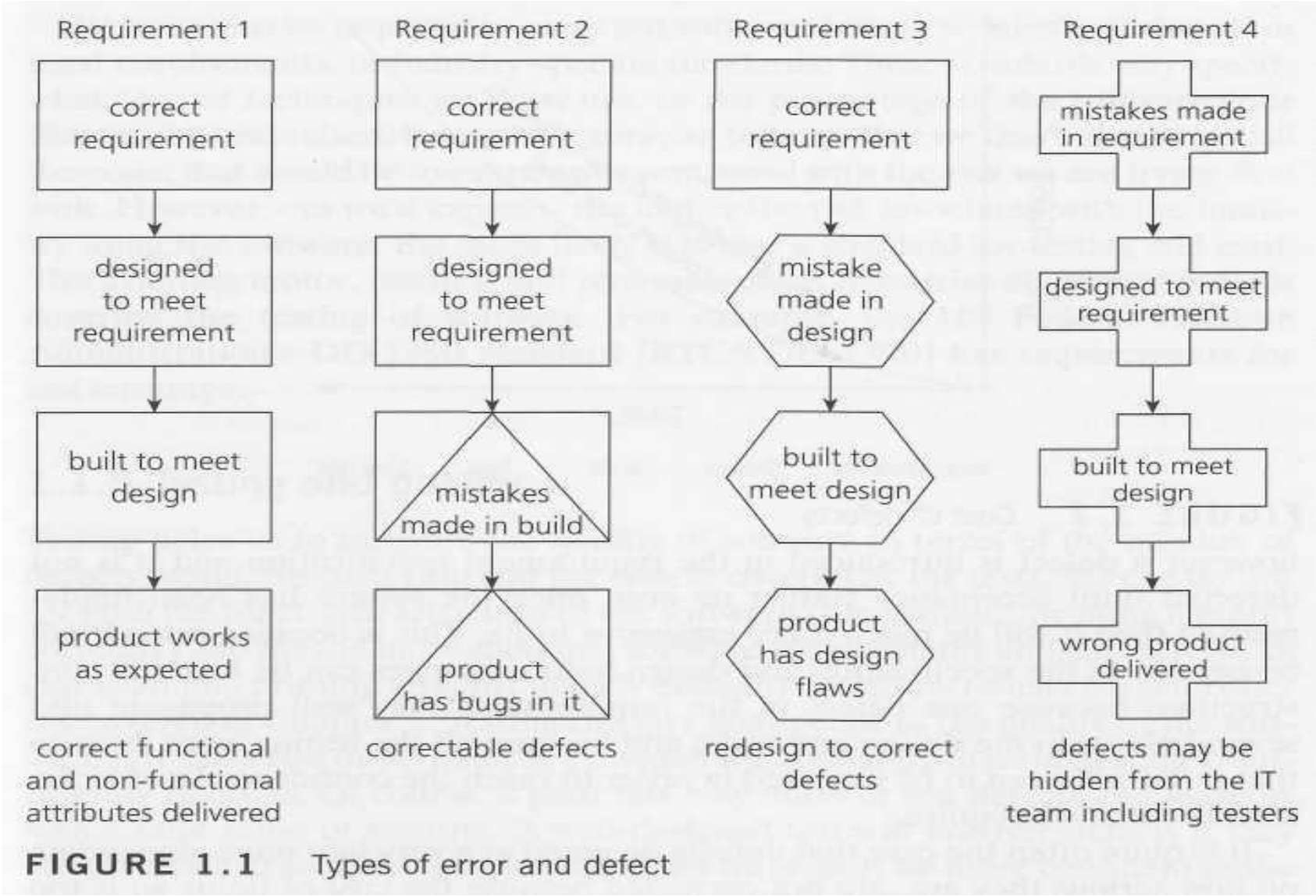


FIGURE 1.1 Types of error and defect

Lỗi tăng lên khi nào?

- Chi phí cho việc tìm thấy và sửa lỗi tăng dần trong suốt chu kỳ sống của phần mềm. Lỗi tìm thấy càng sớm thì chi phí để sửa càng thấp và ngược lại.

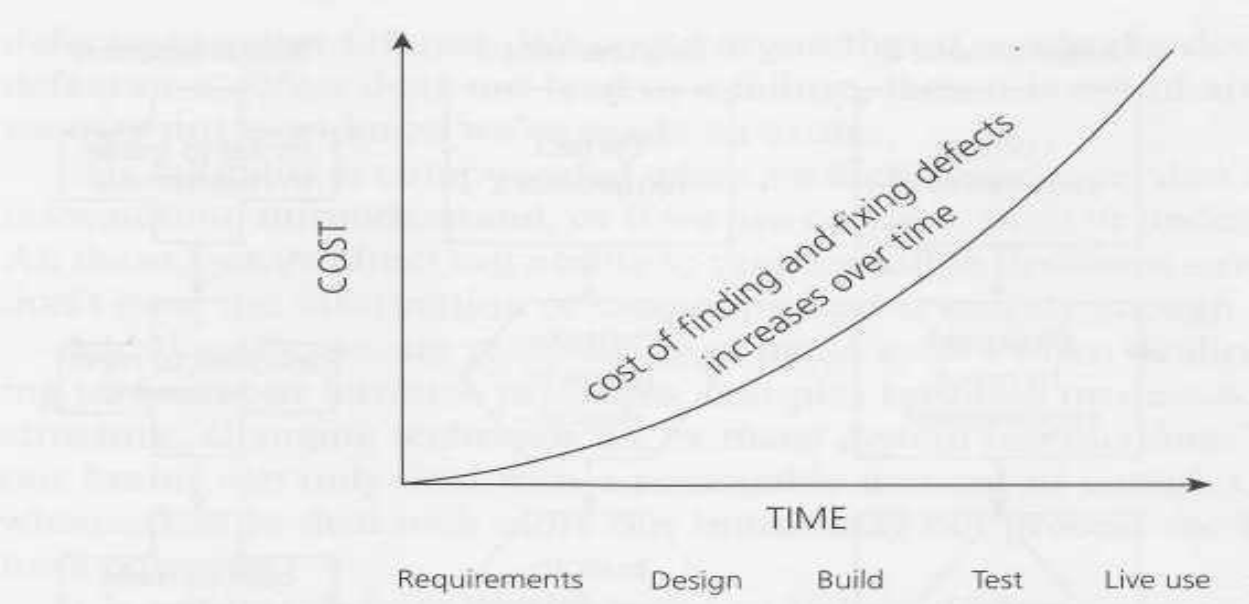


FIGURE 1.2 Cost of defects

Kiểm thử phần mềm trong Agile

- Kiểm thử không còn được thực thi ở cuối vòng đời mà được lồng vào suốt những phân đoạn hoặc Sprint khác nhau
- Kiểm thử là một trong những khác biệt lớn nhất trong Scrum
- Ngay cả với dự án Scrum nhỏ thì cũng khó nhận ra cách để nhóm có thể bàn giao một cách đều đặn nếu không có sẵn một vài cơ chế kiểm thử tự động đã hoạt động tốt

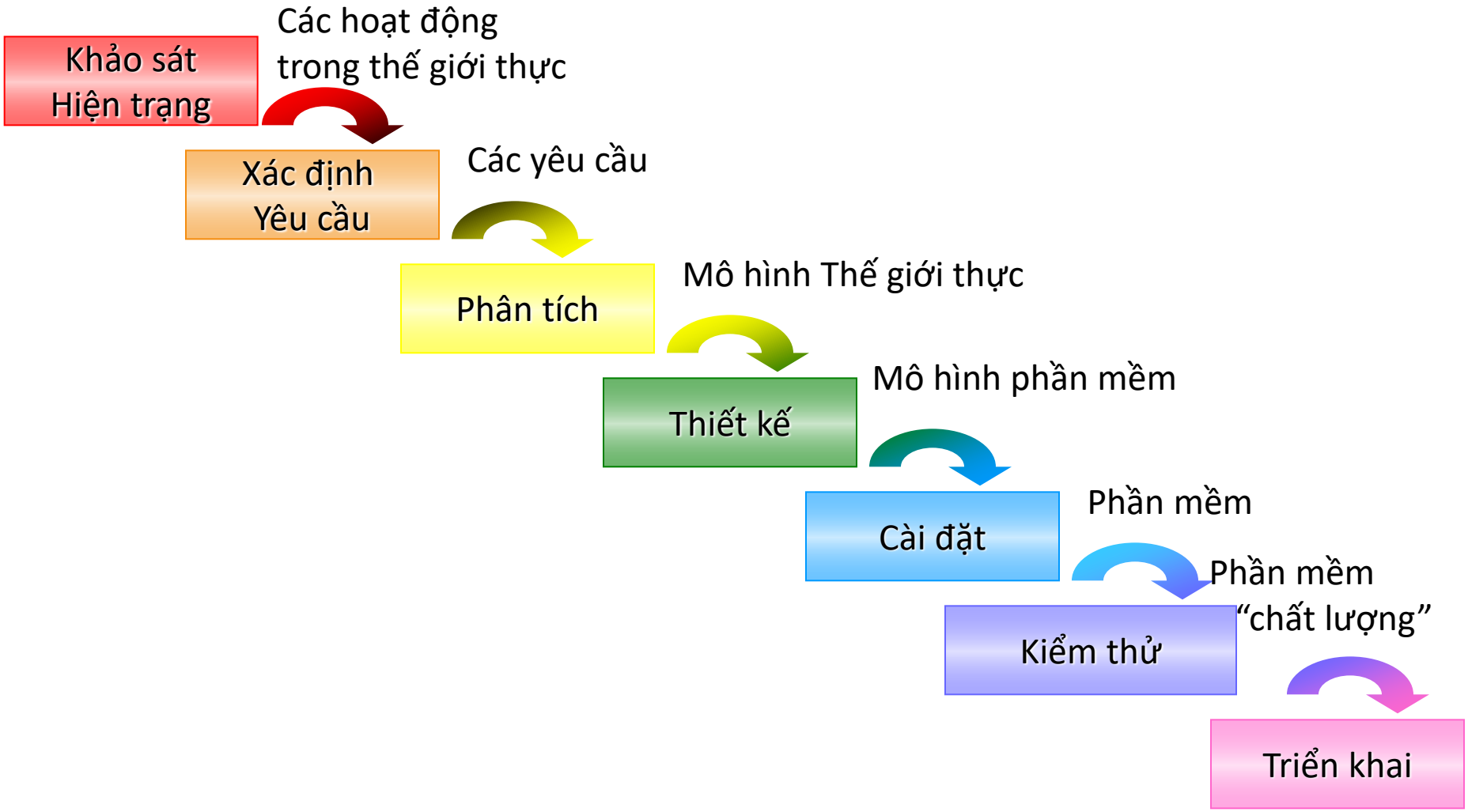
Các hoạt động của kiểm thử?

- Các hoạt động của kiểm thử tồn tại cả trước và sau khi thực thi phần mềm như:
  - Lập kế hoạch test (test plan)
  - Chọn các điều kiện test (test conditions)
  - Thiết kế các trường hợp test (test cases)
  - Kiểm tra kết quả, ước lượng khi nào thì dừng test.
  - Báo cáo kết quả test.

Vai trò của kiểm thử?

- Vai trò kiểm thử trong suốt quy trình sống của phần mềm
  - Kiểm thử không tồn tại độc lập.
  - Các hoạt động của kiểm thử luôn gắn liền với các hoạt động phát triển phần mềm.
  - Các mô hình phát triển phần mềm khác nhau cần các cách tiếp cận test khác nhau.

Mô hình thác nước



Mô hình phát triển Agile

- Chúng ta có thể chia nhỏ phần mềm ra làm nhiều giai đoạn thay vì làm một lần từ đầu đến cuối. Mô hình này cần các hoạt động test như: test chức năng mới, test lặp lại cho những chức năng cũ, và integration test cho cả phần cũ và phần mới.

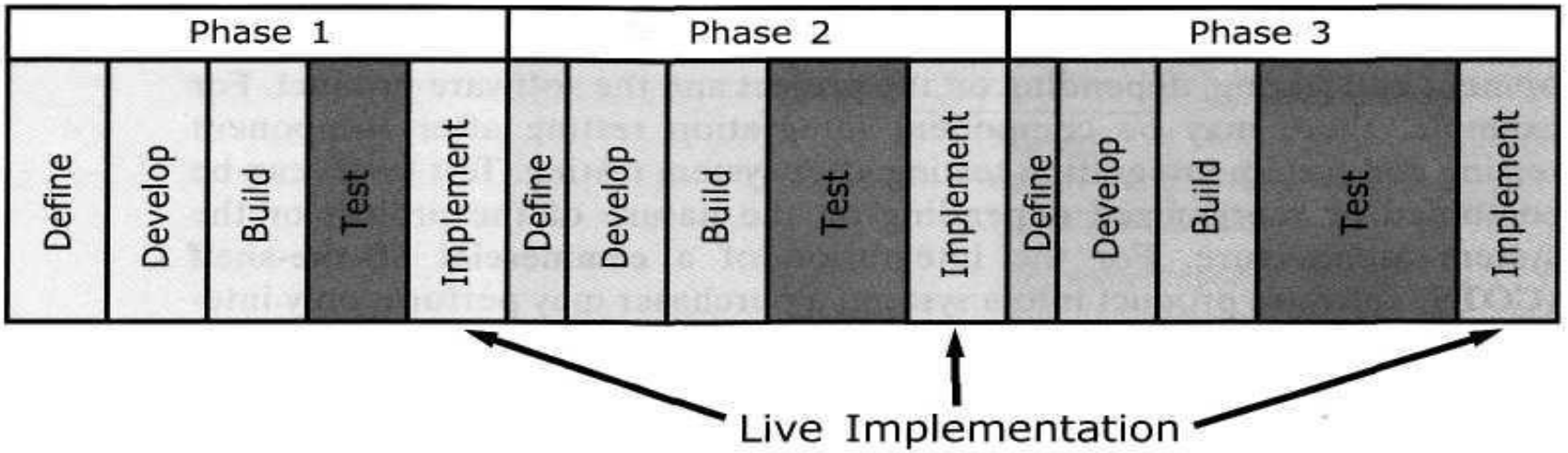
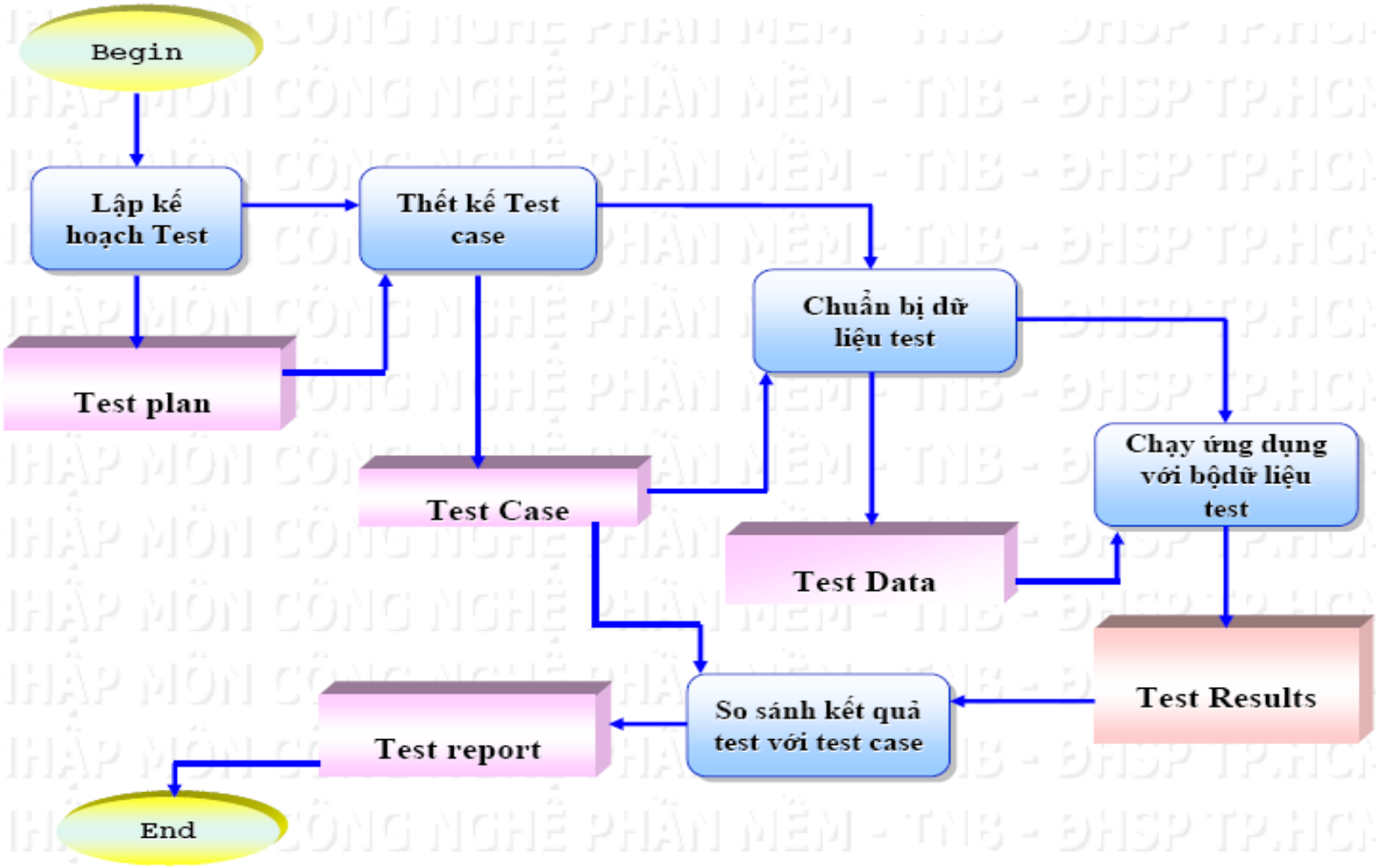


FIGURE 2.3 Iterative development model

Quy trình kiểm thử phần mềm



Các mức độ test (test levels)

- Component testing (unit testing):
  - Tìm lỗi trong các component của phần mềm như: modules, programs, objects, classes,…
  - Ai thực hiện?
- Integration testing:
  - Test sự kết hợp của các component, sự tác động của các phần khác nhau trong một hệ thống, sự kết hợp của các hệ thống với nhau,…

Các mức độ test (test levels)

- System testing: Test hệ thống.
  - Đảm bảo rằng hệ thống (sau khi tích hợp) thỏa mãn tất cả các yêu cầu của người sử dụng
  - Tập trung vào việc phát hiện các lỗi xảy ra trên toàn hệ thống
- Integration testing:Test phần mềm đứng theo góc độ người dùng để xác định phần mềm có được chấp nhận hay không

Thiết kế test case

Test Case	Description	Expected Outcome	New Tags Covered
1	Name: John Smith Acc no: 123456 Loan: 2500 Term: 3 years	Term: 3 years Repayment: 79.86 Interest rate: 10% Total paid: 2874.96	V1, V2, V3, V4, V5 .....
2	Name: AB Acc no: 100000 Loan: 500 Term: 1 year	Term: 1 year Repayment: 44.80 Interest rate: 7.5% Total paid: 537.60	B1, B3, B5, ...

Vai trò Tester

- Kiểm lỗi phần mềm
- Kiểm lỗi bản đóng gói
- Kiểm lỗi tài liệu:
  - Tài liệu cài đặt
  - Tài liệu hướng dẫn sử dụng
  - Tài liệu Release

Công việc của Tester

- Chuẩn bị môi trường Test:
  - Windows XP, 2000, 2003,2008
  - Linux
  - IE, FireFox, Netscape, Mozilla
  - Test Database, Test data
- Thiết kế Test case
- Thực hiện test các Test case trong từng môi trường khác nhau
- Mô tả Bug và chi tiết các bước để tạo ra bug
- Theo dõi quá trình Fix Bug
- Báo cáo kết quả test

Kiểm thử quan trong nhất

- Do Scrum chủ yếu là một khung quản lý dự án, nên nó không nhắc tới những phương pháp kỹ thuật liên quan tới lập trình và kiểm thử
- Bạn nên thực hiện những kiểm thử sau để duy trình những nỗ lực scrum
  - Kiểm thử chấp nhận
  - Kiểm thử tự động
  - Kiểm thử tích hợp liên tục

Kiểm thử chấp nhận (Acceptance Testing)

- Người sử dụng (hoặc khách hàng) chấp nhận kết quả làm việc của Nhóm Phát triển
- Các test được áp dụng đối với tất cả các logic nghiệp vụ quan trọng
- Có thể tự động hóa được thông qua các hệ thống trợ giúp
- Công cụ:
  - Framework for Integrated Test (http://FIT.c2.com)
  - FitNesse (http://www.fitnesse.org/)
  - Các hệ CI

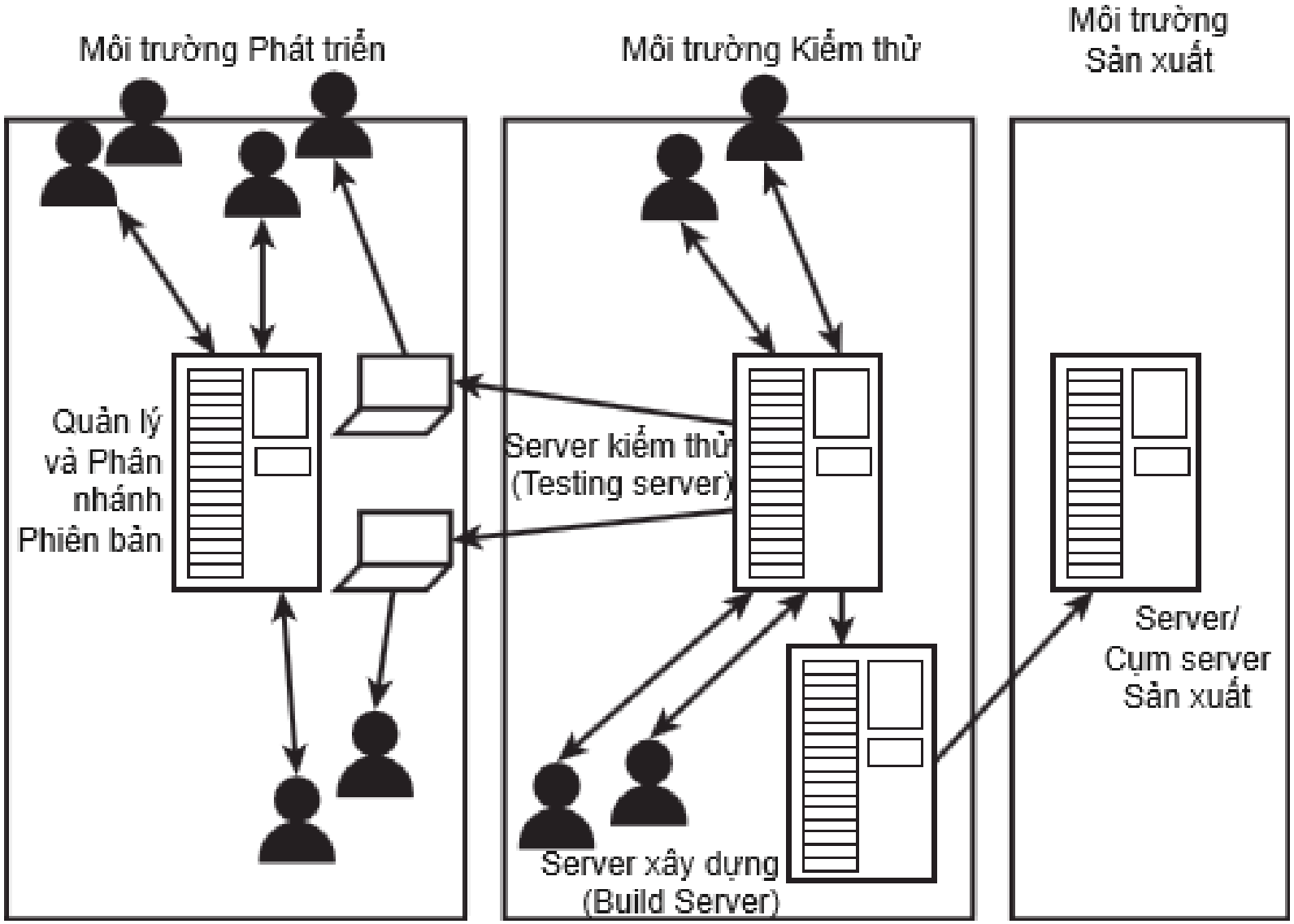
Kiểm thử tự động

- Ngay khi bạn bắt đầu thấy nhóm của mình tung ra những tăng trưởng phần mềm đều đặn, thì bạn sẽ nhận ra rằng mình không thể duy trì được nhịp trừ khi có một số loại tự động hóa
- Kiểm thử tự động sẽ giúp bạn tiết kiệm thời gian
- Một trong những lợi ích chính của kiểm thử tự động là một phần mềm sẽ chạy tất cả kiểm thử cho bạn
- Để có được kiểm thử tự động, bạn phải làm thêm một số việc và phải mua một công cụ và tạo mọi trường hợp kiểm thử.

Kiểm thử tích hợp liên tục

- Một loại kiểm thử khác mà cấp thiết với sự thành công của dự án Scrum là kiểm thử tích hợp liên tục
- Lý do quan trọng để thực hiện đều đặn loại kiểm thử này là vì bạn muốn đảm bảo rằng sản phẩm của bạn luôn luôn chuyển giao được

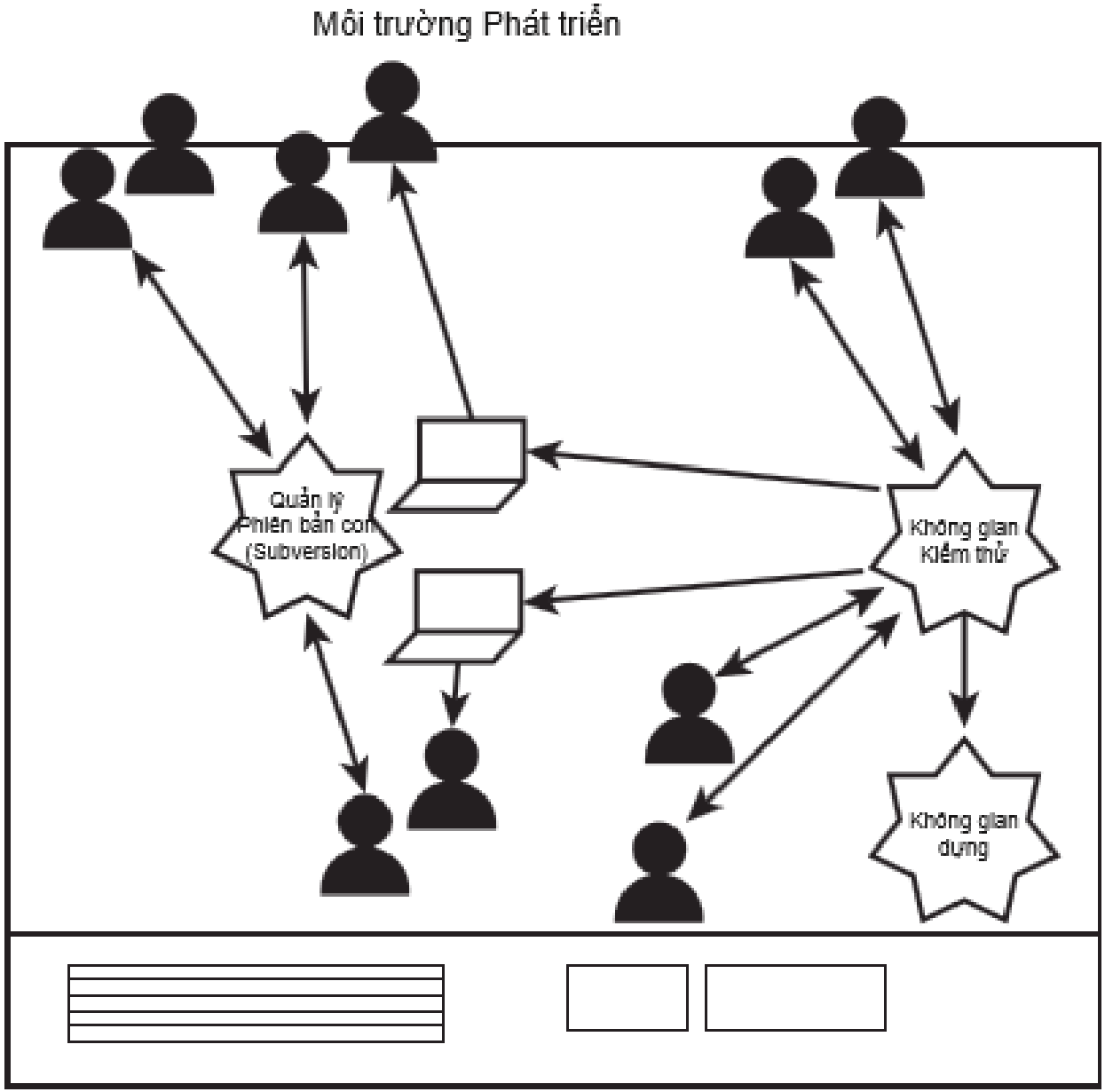
Tổ chức cơ sở hạ tầng kiểm thử



Tổ chức cơ sở hạ tầng kiểm thử

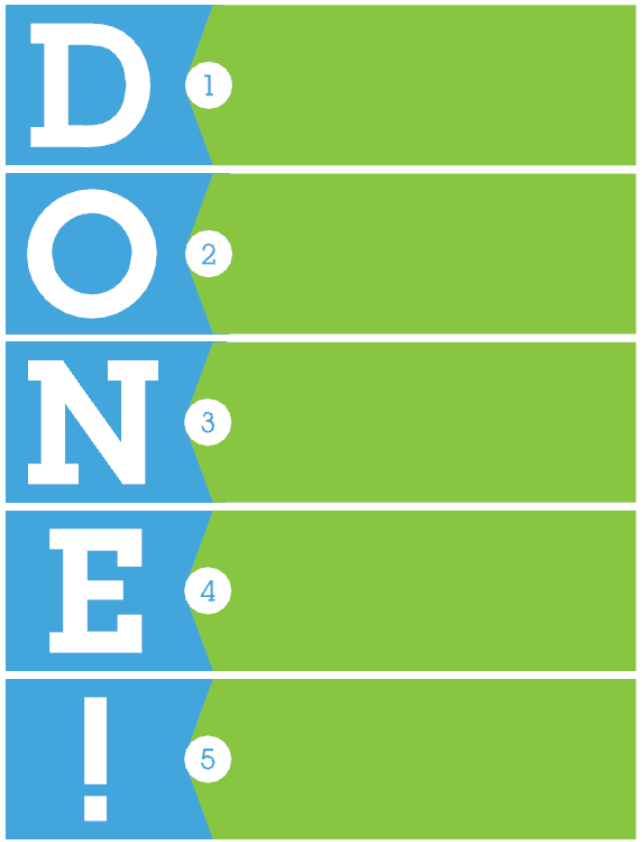
- Bạn nên gặp gỡ thành viên của nhóm kỹ thuật ngay từ đầu dự án, và xem cách họ cài đặt môi trường kiểm thử tạm thời trong môi trường phát triển dự án của bạn
- Nhiều nhóm không bắt đầu như thế, nhưng đó là một phần cốt lõi của mọi dự án Agile hoặc Scrum thành công
- Trừ khi bạn dành thời gian để tổ chức chiến lược kiểm thử của bản thân và tổ chức, còn không bạn sẽ không thể bàn giao chức năng đúng deadline

Tổ chức cơ sở hạ tầng kiểm thử

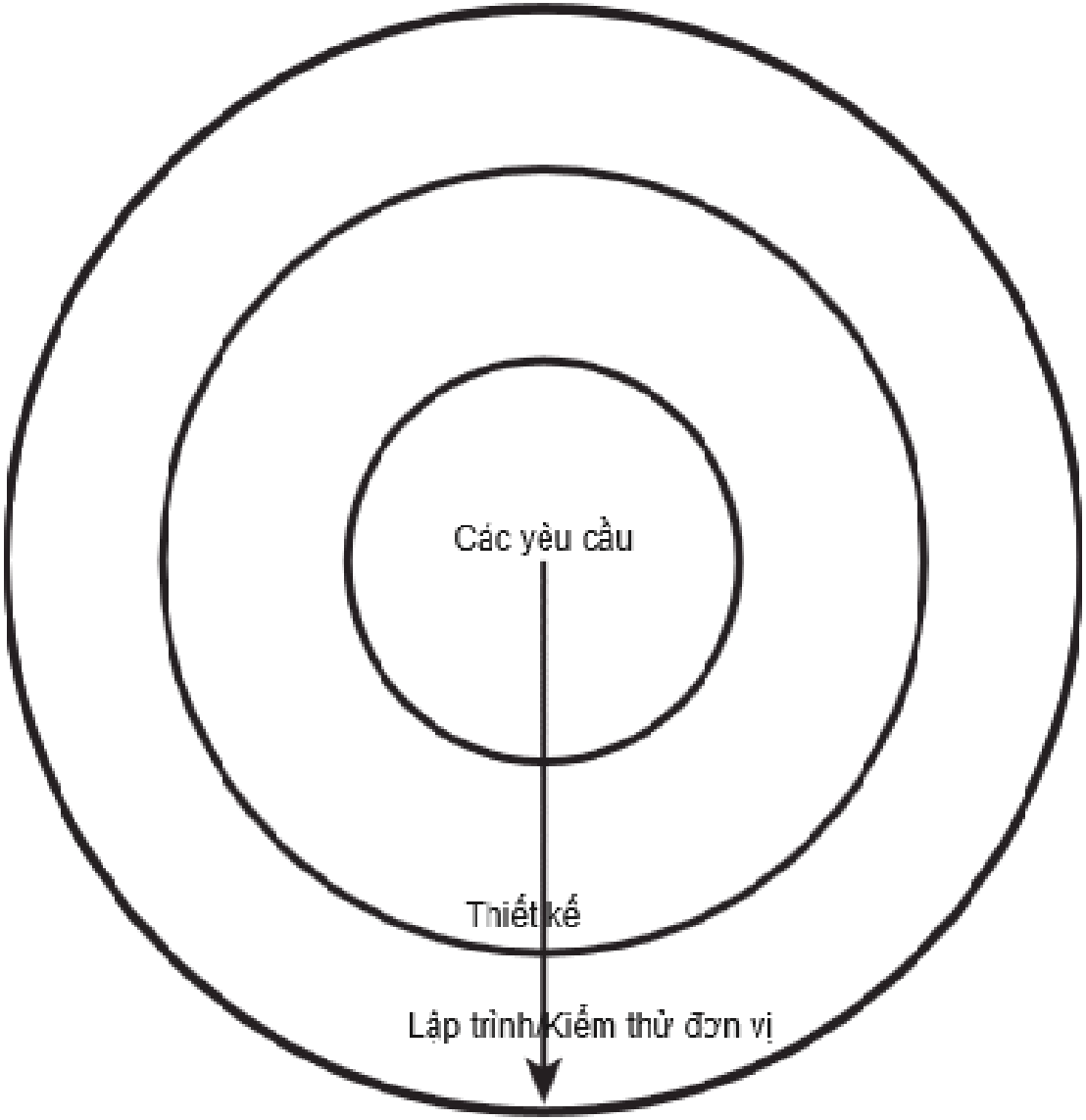


Tầm quan trọng của định nghĩa hoàn thành

- Định nghĩa hoàn thành thường khác nhau, tùy thuộc vào tình hình dự án
- Định nghĩa hoàn thành đầu tiên cho thấy rằng nhóm đã quyết định rằng công việc được coi là hoàn thành khi kết thúc việc lập trình và kiểm thử đơn vị

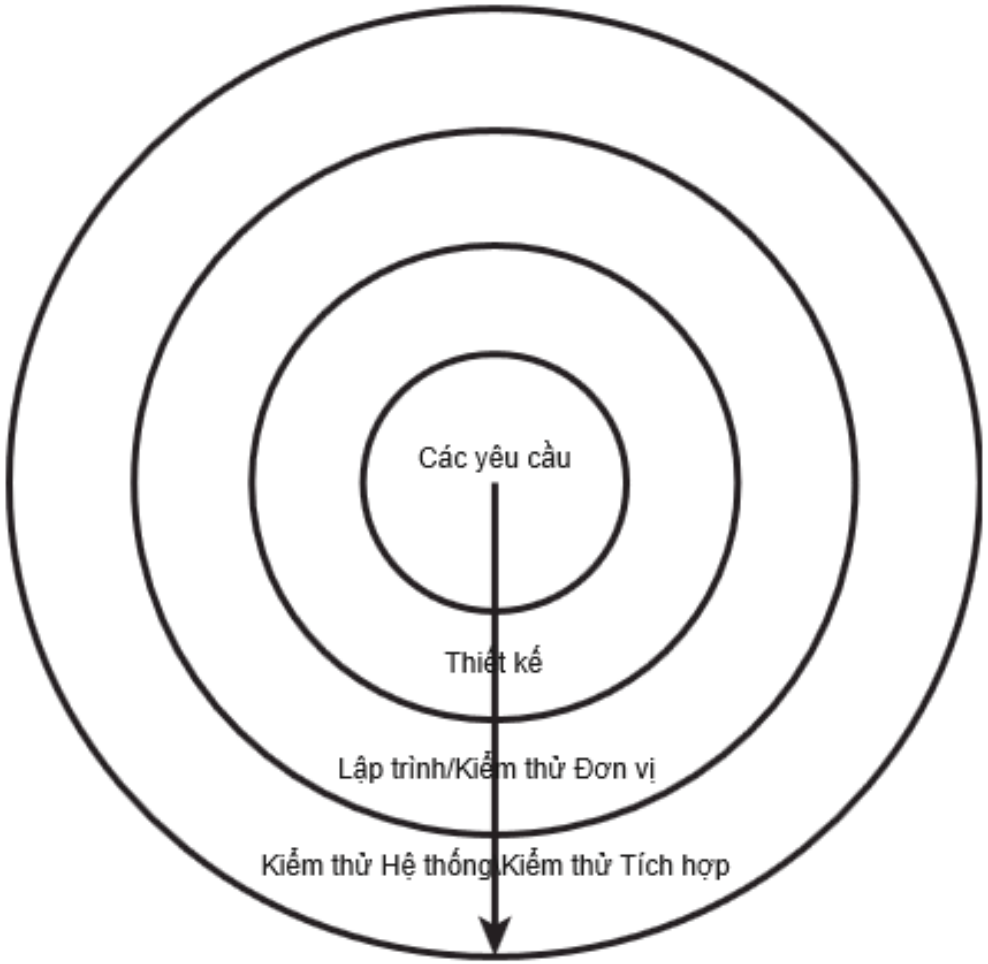


Một định nghĩa hoàn thành



Định nghĩa khác về hoàn thành

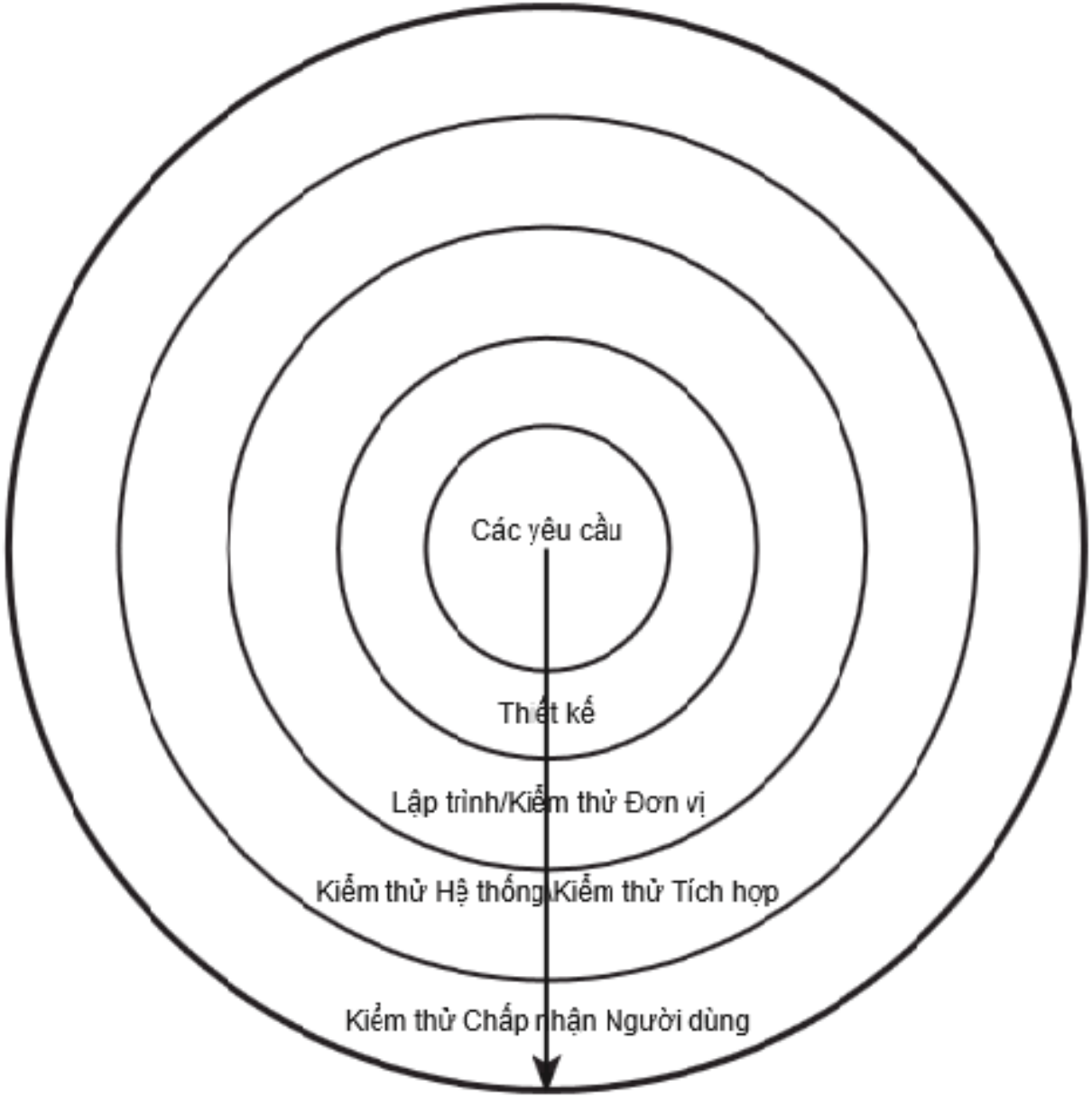
- Một kịch bản khác ở đó nhóm cho rằng họ chỉ hoàn thành khi mọi user story mới đã được tích hợp và kiểm thử trước buổi sơ kết Sprint



Tầm quan trọng của định nghĩa hoàn thành

- Kịch bản tốt nhất, trong đó người dùng nghiệp vụ là một phần của nhóm Scrum, và họ có trách nhiệm thực hiện các kiểm thử chấp nhận trước khi được cho là hoàn thành
- Dù dự án có hay không sử dụng Scrum, việc phát triển kiểm thử chấp nhận người dùng ngày càng trở thành một phương pháp phổ biến và nên được đưa vào các phân đoạn phát hành càng sớm càng tốt trong quy trình

Định nghĩa Hoàn thành



Thảo luận trao đổi tình huống

- Tester hỏi ScrumMaster: “Này, ScrumMaster, chẳng có gì để kiểm thử vào lúc này cả, vậy tôi nên làm gì đây?

Họp cải tiến Sprint

- Dừng và nhìn lại, tìm kiếm các cải tiến và xây dựng tổ chức học tập
- Khung thời gian: 3 giờ
- Thành phần: Scrum Master + Nhóm Phát triển (Product Owner có thể tham dự)
- Scrum Master trợ giúp nhóm tìm hiểu, không đưa ra câu trả lời

Họp cải tiến Sprint

- Câu hỏi:
  - Đã làm tốt những gì?
  - Phải cải thiện những gì?

Workshop 5

- Chuẩn bị trước buổi Workshop:
  - Các nhóm chuẩn bị giải pháp kiểm thử cho dự án
  - Nộp giải pháp kiểm thử đối với dự án lên LMS (kế hoạch kiểm thử, test case cho dự án)
  - Tiến hành thảo luận về buổi họp cải tiến Sprint. Nộp biên bản cuộc họp và nội dung buổi họp lên LMS
- Nội dung trong buổi Workshop
  - Các nhóm trình bày về giải pháp kiểm thử đối với dự án
  - Các nhóm trao đổi về cuộc họp cải tiến Sprint và trao đổi với giảng viên

Tổng kết nội dung bài học

- Khái niệm kiểm thử phần mềm
- Một số đặc điểm của kiểm thử phần mềm
- Tại sao kiểm thử lại cần thiết?
- Quy trình kiểm thử
- Các mức độ test
- Vai trò của Tester
- Công việc Tester
- Tổ chức cơ sở hạ tầng kiểm thử
- Định nghĩa hoàn thành


KẾT THÚC