

Chapter 20

■ Testing Web Applications

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 7/e

by Roger S. Pressman

Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman

For non-profit educational use only

May be reproduced ONLY for student use at the university level when used in conjunction with *Software Engineering: A Practitioner's Approach, 7/e*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information MUST appear if these slides are posted on a website for student use.

Kiểm thử chất lượng một chiều

- **Nội dung** được đánh giá về cả ngữ nghĩa lẫn cú pháp.
 - Mức cú pháp: Lỗi chính tả, ngữ pháp, dấu câu đều được đánh giá đối với các văn bản dựa vào nội dung.
 - Mức ngữ nghĩa: tính đúng đắn (của thông tin được trình bày), tính nhất quán và tính trong sáng đều được đánh giá.
- **Chức năng** được kiểm tra về tính đúng đắn, sự không ổn định và các chuẩn đối với các thư viện thực thi (Ví dụ Java, XML...).
- **Cấu trúc** được đánh giá để đảm bảo về:
 - Trình bày nội dung và chức năng của WebApp.
 - Tính mở rộng.
 - Được hỗ trợ như nội dung hay chức năng mới được thêm vào.

Kiểm thử chất lượng hai chiều

- *Tính sử dụng* được kiểm tra để đảm bảo mỗi mục của người dùng
 - Được hỗ trợ bởi những thứ chung .
 - Nhận biết và áp dụng tất cả cú pháp chuyển hướng yêu cầu và ngữ nghĩa .
- *Khả năng điều hướng* được kiểm tra để đảm bảo rằng
 - Tất cả cú pháp và ngữ nghĩa đều được thực thi để phát hiện bất kì lỗi nào như sai và không đúng liên kết...
- *Việc thực thi* được kiểm tra theo những điều kiện, cấu hình để đảm bảo rằng
 - Hệ thống đáp ứng các tương tác của người dùng.
 - Hệ thống xử lí loading mà không làm giảm những thao tác không chấp nhận.

Kiểm thử chất lượng ba chiều

- *Tính tương hợp* được kiểm tra bằng việc thực thi WebApp trên đa dạng host khác nhau trên cả client và server.
 - Mục đích là tìm lỗi để mà là đặc trưng đối với việc cấu hình host duy nhất.
- *Khả năng tương thích* được kiểm tra để đảm bảo tương tác của WebApp với những ứng dụng hay cơ sở dữ liệu khác.
- *Tính bảo mật* được kiểm tra bằng cách đánh giá các lỗ hổng và tập trung vào khai thác các lỗ hổng này.
 - Bất kì quá trình xâm nhập thành công sẽ là thất bại của hệ thống an ninh.

Lỗi trong WebApp

- Do rất nhiều kiểu kiểm thử WebApp không phát hiện hết các lỗi nên bên phía người dùng khi chạy thử phần mềm sẽ thường gặp lỗi.
- Do WebApp được thực thi trên các môi trường khác nhau nên khó có thể tạo ra các lỗi bên ngoài môi trường mà lỗi ban đầu gặp phải.
- Một số lỗi có thể là kết quả của việc thiết kế không chính xác hoặc HTML không đúng , ngoài ra nhiều lỗi còn do cấu hình WebApp.
- Bởi vì WebApp thuộc về kiến trúc client/server, lỗi có thể khó theo vết trong 3 mức :client,server và mạng .
- Một số lỗi do thao tác tĩnh của môi trường, trong khi một số lỗi khác do thao tác động của môi trường.

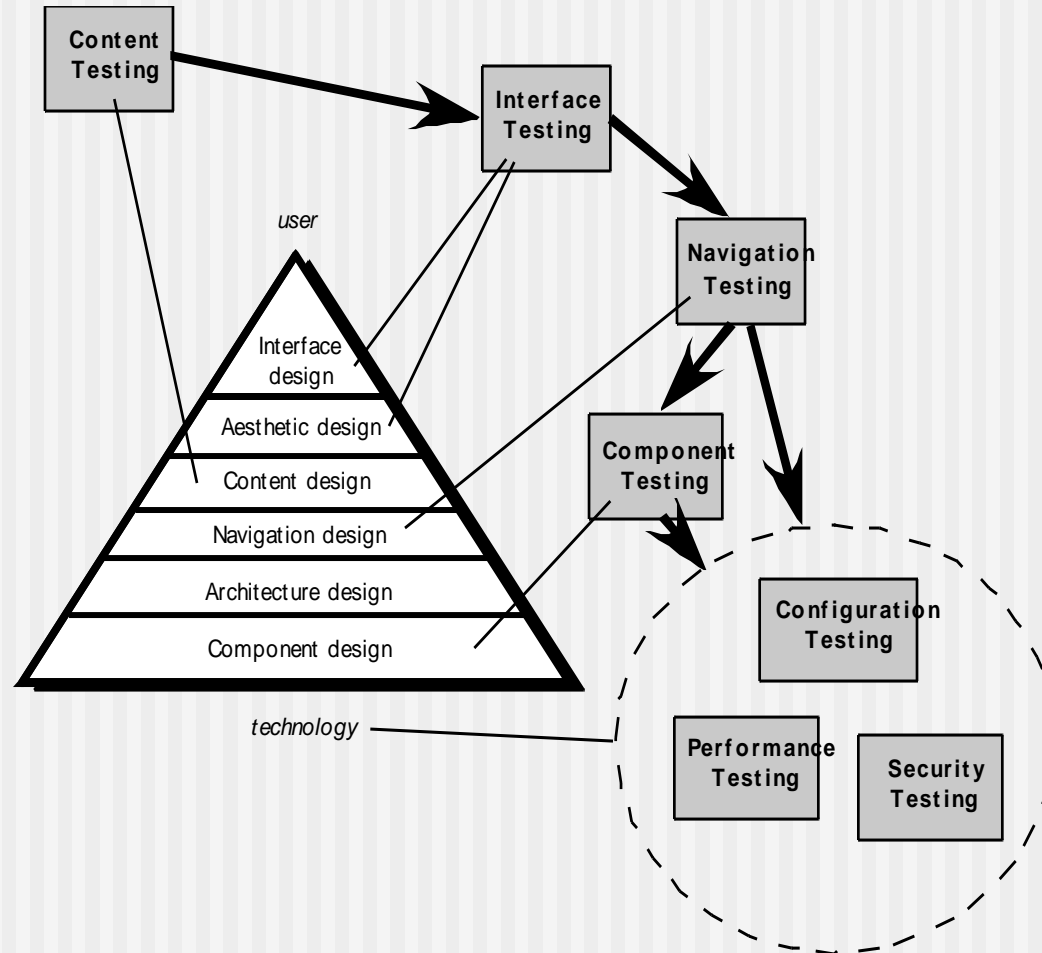
Chiến lược kiểm thử WebApp I

- Mô hình kiểm thử này cho WebApp sẽ được xem xét để phát hiện ra các lỗi.
- Giao diện của mô hình sẽ được xem xét để đảm bảo tất cả use-case đều được điều tiết hợp lí.
- Mô hình thiết kế của WebApp cũng được xem xét để phát hiện ra các lỗi.
- Giao diện người dùng được kiểm tra để phát hiện những lỗi trong quá trình thực thi.
- Thành phần chức năng được lựa chọn là những đơn vị được kiểm tra.

Chiến lược kiểm thử WebApp II

- Chuyển tiếp giữa các phần được kiểm tra.
- WebApp sẽ được thực thi trên nhiều môi trường khác nhau và được kiểm tra trên mỗi môi trường đó.
- Kiểm tra tính bảo mật được kiểm tra để tập trung phát hiện các lỗi của WebApp trên các môi trường cài đặt.
- Kiểm tra hiệu năng cũng được thực hiện.
- WebApp được kiểm tra bởi người điều phối và giám sát cuối cùng
 - Kết quả của những tương tác đó với hệ thống được đánh giá về nội dung, các lỗi và liên quan đến tính tương thích, tính ổn định và hiệu năng của WebApp.

Tiến trình kiểm thử



These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

Nội dung kiểm thử

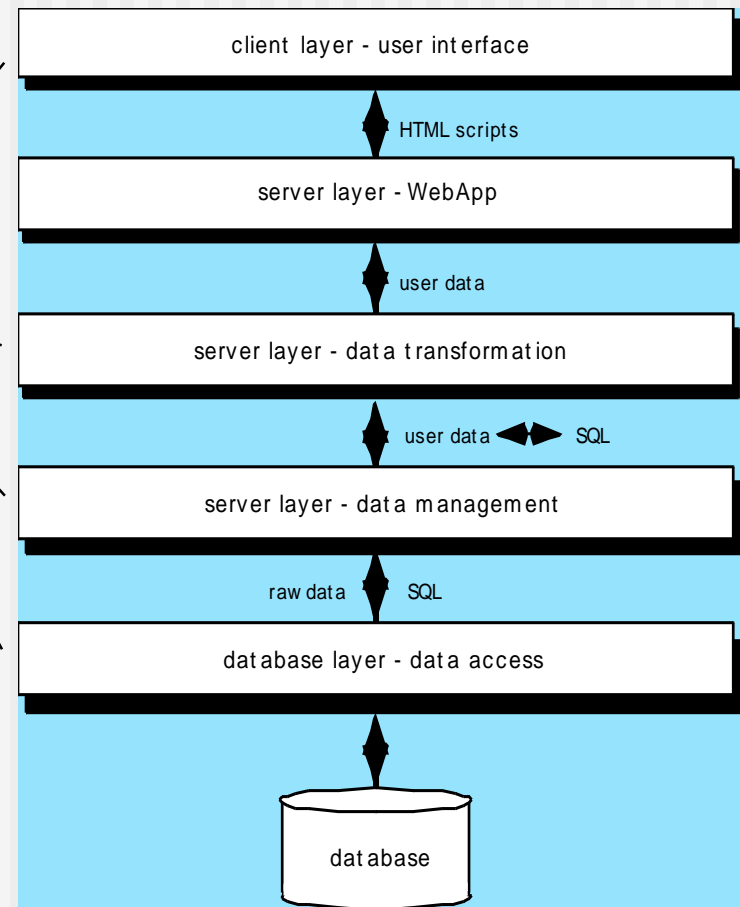
- Nội dung kiểm thử gồm 3 mục tiêu quan trọng :
 - Phát hiện lỗi cú pháp (lỗi ngữ pháp ...) trong một số văn bản , sơ đồ hay một số văn bản khác.
 - Phát hiện lỗi ngữ nghĩa(lỗi về tính chính xác,đúng đắn của thông tin ...) trong nội dung của các đối tượng được trình bày.
 - Tìm lỗi trong các nội dung cấu trúc được trình bày tới người dùng cuối cùng.

Đánh giá nội dung ngữ nghĩa

- Thông tin thật sự chính xác chưa ?
- Thông tin đã súc tích và trình bày được vấn đề chưa ?
- Bố cục của các đối tượng đã làm cho người dùng dễ hiểu chưa ?
- Thông tin đưa vào trong các đối tượng có thể được tìm thấy dễ dàng không ?
- Tài liệu tham khảo đã cung cấp đầy đủ thông tin tham khảo từ các nguồn khác ?
- Thông tin đã được trình bày nội bộ và phù hợp với các thành phần khác chưa?
- Nội dung có sai lệch hay không đúng không ?
- Nội dung của phần mềm có vi phạm bản quyền hay không ?
- Có chứa các liên kết nội bộ không ? Các liên kết đó có chính xác không ?
- Giao diện có đẹp và bắt mắt không ?

Kiểm thử cơ sở dữ liệu

**Kiểm tra được định nghĩa
cho mỗi tầng**



Kiểm thử giao diện người dùng

- Thuộc tính của giao diện được kiểm tra để đảm bảo các quy tắc thiết kế, thẩm mỹ và nội dung liên quan hợp lệ với người dùng và không có lỗi.
- Từng giao diện riêng biệt được kiểm tra giống nhau theo từng đơn vị kiểm tra.
- Từng giao diện được kiểm tra trong phạm vi của các use-case hoặc NSU đối với từng người dùng cụ thể .
- Giao diện hoàn chỉnh được kiểm tra đối với mỗi test-case và để phát hiện lỗi ngữ nghĩa đối với giao diện .
- Giao diện cũng sẽ được kiểm tra trên nhiều môi trường khác nhau để đảm bảo giao diện tương thích trên các môi trường đó .

Cơ chế kiểm thử giao diện I

- **Liên kết(links):** Cơ chế điều hướng để liên kết người dùng với các đối tượng hay chức năng khác.
- **Hình thái(Forms):** Cấu trúc chứa các trường trống được điền vào bởi người dùng. Dữ liệu sẽ được làm đầu vào cho các chức năng của WebApp.
- **Scripting phía client:** Danh sách các tập lệnh trong ngôn ngữ scrip (ví dụ Javascript) để xử lý dữ liệu đầu vào thông qua các form hay dữ liệu người dùng khác.
- **HTML động :** Các đối tượng được thao tác bên phía client sử dụng script hay như kiểu CSS.
- **Cửa sổ pop-up phía client:** Các cửa sổ nhỏ xuất hiện đột ngột mà không tương tác với người dùng.

Cơ chế kiểm thử giao diện II

- ***CGI scripts***: Một giao diện script thực thi các phương pháp chuẩn cho phép Web server tương tác với người dùng.
- ***Streaming content***: Thay vì chờ dữ liệu từ phía client, các đối tượng sẽ download tự động từ phía server. Cơ chế này gọi là “đẩy”(push) server đẩy dữ liệu cho client.
- ***Cookies***: Khối dữ liệu được gửi từ máy chủ và lưu trữ bởi trình duyệt như kết quả của các tương tác với người dùng. Nội dung của dữ liệu là các thành phần đặc trưng của WebApp.
- ***Application specific interface mechanisms***: Gồm một hay nhiều macro , cơ chế tương tác như giỏ hàng, thẻ tín dụng và tính toán chi phí vận chuyển,

Kiểm thử khả năng sử dụng

- Được thiết kế bởi Web team và thực thi bởi người dùng cuối cùng.
- Kết quả kiểm thử
 - Xác định khả năng sử dụng của từng phần và mục đích của mỗi phần.
 - Kiểm tra thiết kế cho phép mỗi mục tiêu được đánh giá .
 - Các thành phần lựa chọn sẽ được kiểm tra .
 - Tương tác với WebApp khi kiểm tra được thực thi.
 - Phát triển cơ chế đánh giá khả năng sử dụng của WebApp.
- Các mức trừu tượng khác nhau :
 - Khả năng sử dụng của giao diện có thể được đánh giá.
 - Khả năng sử dụng của trang web được đánh giá.
 - Khả năng sử dụng của WebApp được xem xét .

Kiểm thử tính tương thích

- Kiểm thử tính tương thích là xác định tập các cấu hình thương gập và các biến thể của nó .
- Tạo cấu trúc cây xác định
 - Mỗi nền tảng tính toán .
 - Cơ chế hiển thị đặc trưng .
 - Hệ điều hành hỗ trợ nền tảng nào.
 - Trình duyệt sử dụng.
 - Tốc độ kết nối internet.
 - Các thông tin tương tự.
- Nhận chuỗi kiểm tra tương thích
 - Nhận từ kiểm thử giao diện, hiệu năng và bảo mật .
 - Mục đích của kiểm thử này là để phát hiện các lỗi và các vấn đề thực thi.

Kiểm thử mức thành phần

- Tập trung vào kiểm thử để phát hiện ra các lỗi của WebApp.
- Phương pháp kiểm thử hộp đen và hộp trắng có thể được sử dụng.
- Kiểm thử cơ sở dữ liệu là phần không thể thiếu của kiểm thử mức thành phần.

Kiểm thử điều hướng (Navigation Testing)

- Các cơ chế điều hướng sau nên được kiểm tra :
 - *Navigation links* : Cơ chế này bao gồm các liên kết nội bộ và liên kết bên ngoài đến WebApp.
 - *Redirects*: Các liên kết yêu cầu một URL không tồn tại hay lựa chọn một trang đã bị bỏ.
 - *Bookmarks*: Mặc dù bookmark là một chức năng của WebApp nhưng vẫn cần kiểm tra để đảm bảo tiêu đề trang có ý nghĩa như bookmark được tạo ra.
 - *Frames and framesets*: Kiểm thử tính chính xác, bố cục ,kích thước,hiệu suất và khả năng tương thích của trình duyệt.
 - *Site maps*: Các ánh xạ của các trang sẽ được kiểm tra để đảm bảo liên kết đưa người dùng đến nội dung hợp lệ.
 - *Internal search engines*: Kiểm tra tính hợp lệ, chính xác ,đầy đủ của cơ chế search,quá trình xử lí lỗi cũng như các chức năng tìm kiếm nâng cao khác.

Kiểm thử ngữ nghĩa điều hướng I

- NSU có đạt được tính toàn vẹn mà không có lỗi?
- Mọi nút điều hướng (định nghĩa cho 1 NSU) có đạt được với ngữ cảnh đường điều hướng định nghĩa bởi NSU?
- Nếu NSU có thể đạt được sử dụng nhiều hơn một đường điều hướng, mọi đường thích hợp có được kiểm tra?
- Nếu sự hướng dẫn được cung cấp bởi giao diện người dùng để trợ giúp trong điều hướng, phương hướng có đúng và có thể hiểu được như tiếp tục điều hướng?
- Có kỹ thuật (khác với mũi tên duyệt “đen”) cho điều hướng lại nút điều hướng tiếp tục và bắt đầu của đường điều hướng không?
- Kỹ thuật cho điều hướng với nút điều hướng lớn có làm việc đúng không?
- Nếu một hàm để được thực hiện tại một nút và người dùng chọn không cung cấp đầu vào, phần còn lại của NSU có thể được hoàn thành không?

Kiểm thử ngữ nghĩa điều hướng

-II

- Nếu một hàm được thực hiện tại một nút và một lỗi trong xử lý hàm xảy ra, NSU có thể hoàn thành không?
- Có một cách để ngừng điều hướng trước khi tới tất cả các nút, nhưng sau đó quay lại đến nơi mà điều hướng đã ngừng và tiếp tục từ đó không?
- Mọi nút có tới được từ sơ đồ vị trí không? Tên các nút có ý nghĩa với người sử dụng cuối không?
- Nếu một nút với một NSU được đi tới từ nguồn bên ngoài, có thể xử lý nút kế tiếp trên đường điều hướng không. Có thể quay lại nút trước đó trên đường điều hướng không?
- Người dùng có hiểu được vị trí của họ với kiến trúc nội dung như NSU được thực hiện?

Kiểm thử cấu hình

■ Phía server

- WebApp có hoàn toàn tương thích với hệ điều hành máy chủ?
- File, thư mục hệ thống và dữ liệu hệ thống liên quan có được tạo đúng khi WebApp vận hành?
- Biện pháp bảo mật hệ thống (ví dụ tường lửa hoặc mã hóa) có cho phép WebApp thực thi và phục vụ người dùng mà không có trở ngại hoặc giảm hiệu năng?
- WebApp có được kiểm thử với cấu hình máy chủ phân tán (nếu tồn tại) được chọn?
- WebApp có được tích hợp đúng với phần mềm cơ sở dữ liệu? WebApp có chính xác với các phiên bản khác nhau của phần mềm cơ sở dữ liệu?
- Kịch bản WebApp phía máy chủ có thực thi đúng?
- Lỗi quản trị hệ thống có được xem xét tác động của nó tới vận hành WebApp?
- Nếu máy chủ proxy được sử dụng, sự khác biệt trong cấu hình có được gửi với kiểm tra on-site?

Kiểm thử cấu hình

- Phía client
 - *Phần cứng*—CPU, bộ nhớ, lưu trữ và thiết bị in
 - *Hệ điều hành*—Linux, Macintosh OS, Microsoft Windows, a mobile-based OS
 - *Trình duyệt*—Internet Explorer, Mozilla/Netscape, Opera, Safari,...
 - *Thành phần giao diện người dùng*—Active X, Java applets, ...
 - *Plug-ins*—QuickTime, RealPlayer, ...
 - *Kết nối*—cable, DSL, regular modem, T1
- Số lượng các biến cấu hình phải được giảm đến một số có thể quản lý

Kiểm thử bảo mật

- Được thiết kế để thăm dò tính có thể tấn công của môi trường phía client, truyền thông mạng mà có thể xảy ra như dữ liệu được chuyển từ client tới server và quay lại, và môi trường phía server
- Tại phía client, tính có thể tấn công thường xuyên được theo dõi lỗi tồn tại trước trong trình duyệt, chương trình e-mail hoặc phần mềm truyền thông.
- Tại phía server, tính có thể tấn công bao gồm tấn công từ chối dịch vụ và mã độc có thể truyền tới phía client hoặc sử dụng để vô hiệu sự hoạt động của server.

Kiểm thử hiệu năng

- Thời gian phản hồi máy chủ có giảm tới điểm đáng chú ý hoặc không thể chấp nhận?
- Tại điểm nào (theo người dùng, giao dịch hoặc tải dữ liệu) mà hiệu suất trở nên không thể chấp nhận?
- Thành phần hệ thống có trách nhiệm gì cho việc giảm hiệu năng?
- Thời gian phản hồi cho người dùng là bao nhiêu dưới sự thay đổi của điều kiện tải?
- Sự giảm hiệu năng có ảnh hưởng đến bảo mật hệ thống?
- Sự tin cậy và độ chính xác của WebApp có bị ảnh hưởng khi tải trên hệ thống tăng lên?
- Chuyện gì xảy ra khi tải lớn hơn khả năng cực đại của máy chủ?

Kiểm thử tải

- Ý định là để xác định WebApp và môi trường phía server của nó sẽ phản hồi như thế nào với điều kiện tải khác nhau
 - N , số lượng người dùng đồng thời
 - T , số lượng giao dịch trực tuyến trên đơn vị thời gian
 - D , tải dữ liệu được xử lý bởi máy chủ trên giao dịch
- Lưu lượng tổng cộng, P , được tính toán theo cách sau:
 - $P = N \times T \times D$

Kiểm thử ứng suất

- Hệ thống có giảm “nhẹ” hoặc máy chủ có tắt khi khả năng vượt quá ?
- Phần mềm máy chủ có phát sinh thông điệp “server not available”? Chung hơn, người dung có nhận thức rằng họ không thể tới được máy chủ?
- Hàng đợi máy chủ có yêu cầu tài nguyên và làm rỗng hàng đợi khi mà yêu cầu khả năng giảm xuống?
- Tính toàn vẹn dữ liệu có bị ảnh hưởng khi khả năng bị vượt quá?
- Giá trị nào của N, T, và D buộc môi trường máy chủ phải lỗi? Lỗi tự nó biểu lộ thế nào? Thông báo có tự động gửi tới nhân viên hỗ trợ kỹ thuật tại phía server?
- Nếu hệ thống lỗi, mất bao lâu để trở lại trực tuyến?
- Chức năng WebApp chắc chắn (ví dụ tính toán cường độ, dung lượng dòng dữ liệu) có bị ngừng khi khả năng đạt tới mức 80% hoặc 90 %?