Kiểm thử phần mềm

Chương 11: Kiểm thử giao diện và tiện dụng

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Nội dung

- **Kiểm thử giao diện**
- Kiểm thử tiện dụng
- Công cụ kiểm thử giao diện web

1. Giao diện người dùng

- GUI = Graphic User Interface
- Tại sao GUI phổ biến?
  - Cung cấp cách nhìn và cảm nhận chuẩn
  - Linh hoạt cho hầu hết các loại ứng dụng
  - Cung cấp kết hợp liên tục cho custom/package application
  - Sử dụng kết hợp bàn phím và chuột
  - Giao diện tự nhiên hơn cho người dùng
  - Màn hình được truy cập theo thứ tự mong muốn

1. Giao diện người dùng

- GUIs khác Forms-based Application?
  - Nhiều màn hình cùng hiển thị
  - Có thể điều hướng, truy cập các chức năng khác nhau
  - Không hạn chế thứ tự nhập liệu
  - Chủ động sử dụng chức năng theo cách riêng

2. Khó khăn trong kiểm thử giao diện

- Phần mềm hướng sự kiện (Event-driven software)
- Sự kiện không được yêu cầu (Unsolicited events)
- Hướng đối tượng (Object oriented)
- Đồng bộ và phụ thuộc ẩn (Hidden synchronization and dependencies)
- Không giới hạn miền đầu vào ('Infinite' input domain)
- Nhiều cách nhập xuất (Many ways in, many ways out)
- Quản lý cửa sổ (Window management)

3. Chiến lược kiểm thử giao diện

- Nguyên lý kiểm thử giao diện
- Qui trình kiểm thử giao diện
- Các loại lỗi giao diện
- Bốn giai đoạn kiểm thử giao diện

3. Chiến lược kiểm thử giao diện

- Nguyên lý kiểm thử giao diện
  - Tập trung vào lỗi giao diện để giảm phạm vi kiểm thử
  - Chia vấn đề lớn thành nhiều vấn đề nhỏ
  - Chọn kỹ thuật thiết kế thích hợp
  - Tổ chức kiểm thử thành các giai đoạn
  - Kiểm thử tự động khi có thể

3. Chiến lược kiểm thử giao diện

- Qui trình kiểm thử giao diện

3. Chiến lược kiểm thử giao diện

- Các loại lỗi giao diện
  - Data validation
  - Incorrect field default
  - Mishandling of server process failures
  - Mandatory fields, not mandatory
  - Wrong fields retrieved by queries
  - Incorrect search criteria
  - Field order
  - Multiple database rows returned, single row expected
  - Currency of data on screens
  - Window object/DB field correspondence

3. Chiến lược kiểm thử giao diện

- Các loại lỗi giao diện
  - Correct window modality?
  - Window system commands not available/don't work
  - Control state alignment with state of data in window?
  - Focus on objects needing it?
  - Menu options align with state of data or application mode?
  - Action of menu commands aligns with state of data in window
  - Synchronization of window object content
  - State of controls aligns with state of data in window?

3. Chiến lược kiểm thử giao diện

- Bốn giai đoạn kiểm thử giao diện

Stage	*Test Types*
Low Level	Checklist testing Navigation
Application	Equivalence Partitioning Boundary Values Decision Tables State Transition Testing
Integration	Desktop Integration C/S Communications Synchronisation
Non-Functional	Soak testing Compatibility testing Platform/environment

4. Kỹ thuật kiểm thử giao diện

- Checklist Testing
- Navigation Testing
- Application Testing
- Desktop Integration Testing
- Client/Server Communication Testing
- Synchronization Testing
- Non-Functional Testing

4. Kỹ thuật kiểm thử giao diện

- Checklist Testing
  - Chuẩn giao diện (GUIs standards)
  - Chuẩn ứng dụng (Application standards)

4. Kỹ thuật kiểm thử giao diện

- Navigation Testing
  - Xác định các điều hướng đến chức năng/màn hình khác
  - Xác định các điều hướng đảo ngược/không đảo ngược
  - Xác định các cách của 1 điều hướng
    - Menu, button, phím tắt, …

4. Kỹ thuật kiểm thử giao diện

- Application Testing
  - Tập trung vào ứng xử của các đối tượng bên trong màn hình
  - Áp dụng các kỹ thuật black box testing

Kỹ thuật	Dùng để kiểm thử
Equivalence Partitions and Boundary Value Analysis	-Input validation -Simple rule-based processing
Decision Tables	Complex logic or rule-based processing
State-transition testing	- Applications with modes or states where processing behavior is affected - Windows where there are dependencies between objects in the window.

4. Kỹ thuật kiểm thử giao diện

- Desktop Integration Testing
  - Giao tiếp trực tiếp hay thông qua các dialogue
  - Thông tin được truyền theo 1 hay 2 hướng
  - Giao tiếp đến các ngữ cảnh nhạy cảm
  - Có nhiều loại message khác nhau trong giao tiếp?

4. Kỹ thuật kiểm thử giao diện

- Client/Server Communication Testing
  - Tương tự Desktop Integration Testing ở mức cao hơn
  - Chú ý vào sự phản hồi từ phía server (lỗi, quá tải)

4. Kỹ thuật kiểm thử giao diện

- Synchronization Testing
  - Ứng dụng có nhiều chế độ
  - Dữ liệu trong CSDL thay đổi phải thay đổi dữ liệu trên cửa sổ liên quan
  - Dữ liệu trong 1 cửa sổ thay đổi phải thay đổi dữ liệu trong các cửa sổ liên quan

4. Kỹ thuật kiểm thử giao diện

- Non-Functional Testing
  - Soak testing
  - Compatibility testing
  - Platform/Environment testing

5. Kiểm thử giao diện tự động

- Một số lời khuyên
  - Luật Perato:
    - Chạy tự động 20% test case đạt 80% lợi ích
  - Hướng tiếp cận lai
    - Kết hợp tự động và thủ công
  - Coded scripts
    - Tốt cho navigation và checklist type script
    - Dễ duy trì kiểm tra hồi qui
  - Recorded scripts
    - Cần chỉnh sửa để sử dụng lại
    - Nhạy cảm với thay đổi giao diện
  - Non-Functional Tests
    - Scripts có thể sử dụng cho Soak/Performance testing

5. Kiểm thử giao diện tự động

Loại kiểm thử	Tự động hay thủ công
Checklist testing	Thủ công cho kiểm thử các application conventions Tự động cho kiểm thử object states, menus, standard features
Navigation	Tự động
Equivalence Partitioning, Boundary Values, Decision Tables, State Transition Testing	Thủ công cho kiểm thử phức tạp Tự động cho kiểm thử đơn giản nhiều trường hợp
Desktop Integration, C/S Communications	Thủ công cho tích hợp phức tạp Tự động cho những transactions đơn giản được lặp lại
Synchronization	Thủ công
Soak testing, Compatibility testing, Platform/environment	Tự động

6. Nâng cao khả năng kiểm thử của giao diện

- Thách thứ trong kiểm thử giao diện
  - Khó đặc tả kiểm thử
    - Không có mô tả chức năng bên dưới giao diện
    - Không có mô tả tương tác hướng sự kiện ẩn trong giao diện
  - Khó thiết kế kiểm thử
    - Số lượng test case để phủ ứng dụng rất lớn
    - Sử dụng menu, bàn phím, chuột làm tăng số lượng test case
  - Khó thực thi kiểm thử
    - Thao tác trên thiết vị ngoại vi khó lặp lại
    - Khó giả lập các tương tác ẩn
  - Khó phân tích kiểm thử
    - Màn hình và bên dưới màn hình thay đổi liên tục
    - Các màn hình đồng bộ khi có thay đổi
    - Kết quả mong đợi hiển thị trên màn hình ẩn
    - Các thuộc tính của đối tượng trên màn hình khó kiểm tra bằng mắt

6. Nâng cao khả năng kiểm thử của giao diện

- Lời khuyên thiết kế giao diện
  - 1. Ẩn hoặc vô hiệu hóa các menu dư thừa
  - 2. Tránh hiển thị dữ liệu phụ thuộc trên nhiều cửa sổ
  - 3. Điều hướng nên có thứ bậc
  - 4. Cửa sổ nên là modal
  - 5. Các thành phần trong cửa sổ trách phụ thuộc lẫn nhau, tách thành nhiều cửa sổ modal
  - 6. Giảm tối thiểu system commands (minimize, maximize, restore, close, …)
  - 7. Chức năng được gọi từ nhiều cách (chuột, bàn phím, menu) nên được gọi chung từ 1 phương thức

6. Nâng cao khả năng kiểm thử của giao diện

- Lời khuyên thiết kế giao diện
  - 8. Instrumentation nên được thực thi trong mã nguồn để cung cấp thông tin về giao tác với sản phẩm khác hay server
  - 9. Instrumentation nên được thực thi để cung cấp thông tin về sự kiện không được yêu cầu và giả lập nó

Nội dung

- Kiểm thử giao diện
- **Kiểm thử tiện dụng**
- Công cụ kiểm thử giao diện web

ISO 9126 & Usability

- ISO 9126
  - Tiêu chuẩn quốc tế đánh giá chất lượng của phần mềm.
- 6 đặc tính:
  - Functionality – đảm bảo yêu cầu
  - Reliability – độ tin cậy
  - Usability – độ tiện dụng
  - Efficiency – độ hiệu quả
  - Maintainability – khả năng bảo trì
  - Portability – khả năng tương thích

ISO 9126 & Usability

- Usability: mức độ tiện dụng của phần mềm
- Đánh giá qua các tiêu chí sau:
  - Khả năng hiểu được (understandability)
  - Khả năng học cách sử dụng (learnability)
  - Khả năng thực hiện được (operability)

ISO 9126 & Usability

- Khả năng hiểu được (understandability)
  - Nỗ lực của người dùng để hiểu được hoạt động của phần mềm
  - Ví dụ:
    - Ai là người sử dụng? Họ đã biết những gì và có thể học những gì?
    - Trình độ chung của người sử dụng?
    - Người sử dụng muốn hoặc cần điều gì?
    - Môi trường làm việc của người dùng?

ISO 9126 & Usability

- Khả năng học cách sử dụng (learnability)
  - Nỗ lực của người dùng để học cách sử dụng phần mềm
  - Ví dụ:
    - Người sử dụng có cần phải training nhiều không?
    - Tài liệu hướng dẫn, trợ giúp người dùng?

ISO 9126 & Usability

- Khả năng thực hiện được (operability)
  - Nỗ lực của người dùng để thực hiện chức năng của phần mềm
  - Ví dụ:
    - Những việc gì phần mềm sẽ phải giải quyết, việc gì cần người dùng thực hiện
    - Người sử dụng có thể hoàn tất tác vụ mong muốn trong khoảng thời gian định trước?
    - Có bao nhiêu sai phạm mà người dùng mắc phải khi sử dụng phần mềm?
    - Những thông báo lỗi có dễ hiểu, thân thiện không?
    - Phần mềm có những tính năng gì đặc biệt dành cho người khuyết tật?

ISO 9126 & Usability

- Phương pháp đánh giá
  - Tập trung vào người dùng khi phân tích yêu cầu
  - Xây dựng hồ sơ về người dùng
  - Kiểm tra tính tiện dụng

Usability Testing

- Nhóm đại diện người dùng thực hiện chức năng, nhóm phát triển theo dõi, lắng nghe và nghi nhận
- Đối tượng: prototype giai đoạn đầu, sản phẩm cuối
- Iterating: mô hình lặp

Usability Testing

- Các câu hỏi chung
  - Những người tham gia có thể hoàn tất các tác vụ thành công?
  - Đối với các tác vụ thành công, người tham gia mất bao nhiêu thời gian?
  - Đối với các tác vụ thành công, người dùng phải mở bao nhiêu trang web (bao nhiêu click chuột) để hoàn thành?
  - Người sử dụng có hài lòng về sản phẩm?
  - Cần phải thay đổi những gì để có thể cho phép nhiều người sử dụng thành công hơn nữa.

Usability Testing

- Các câu hỏi chi tiết
  - Người dùng sử dụng hệ thống navigation của trang web hay sử dụng chức năng search?
  - Những từ nào người sử dụng dùng để search nhiều nhất?
  - Ô tìm kiếm có ở vị trí thuận lợi và đủ lớn không?
  - Chức năng search có đáp ứng tốt yêu cầu tìm kiếm của người sử dụng?

Usability Testing

- Yêu cầu
  - Thu thập thông tin về cách sử dụng phần mềm
  - Xác định các vấn đề tính tiện dụng
  - Đánh giá mức độ hài lòng
  - Đưa ra các đề nghị

Phân loại kiểm thử tính tiện dụng

- Exploratory (thăm dò)
- Assessment (đánh giá)
- Validation (xác nhận)
- Comparison (so sánh)

Phân loại kiểm thử tính tiện dụng

- Exploratory (thăm dò)
  - Diễn ra đầu qui trình phát triển
  - Dựa trên các bản vẽ phác thảo, lược đồ về GUI
  - Đánh giá những ý tưởng thiết kế sơ khởi
  - Diễn ra tự nhiên, thoải mái, không hình thức
  - Thảo luận về các ý tưởng ở mức trừu tượng.

Phân loại kiểm thử tính tiện dụng

- Assessment (đánh giá)
  - Thực hiện sau khi các ý tưởng chính yếu đã được xác định
  - Đánh giá tính tiện dụng ở mức độ cụ thể hóa hơn
  - Người dùng thực hiện một số công việc được định nghĩa trước
  - Việc kiểm thử cần được giám sát, ghi nhận
  - Các thông tin về độ đo, số lượng được thu thập

Phân loại kiểm thử tính tiện dụng

- Validation (chứng thực)
  - Thực hiện ở cuối quy trình phát triển
  - Mục tiêu là chứng thực tính tiện dụng của sản phẩm
  - Thường được thực hiện khi đã có phiên bản tổng thể đầu tiên của sản phẩm.
  - Đánh giá dựa trên một số tiêu chuẩn về tiện dụng hoặc độ đo.
  - Các chuẩn này có được từ những lần test trước, hoặc từ các sản phẩm cạnh tranh khác.
  - Các bộ test phải có độ đo chi tiết
  - Có thể dựa vào đây, để phát triển tiếp cho các sản phẩm tương lai.
  - Có thể được thực hiện thông qua beta customers.

Phân loại kiểm thử tính tiện dụng

- Comparison (so sánh)
  - Có thể thực hiện tại bất kỳ thời điểm nào trong quy trình phát triển
  - Đánh giá những lựa chọn khác nhau một cách khách quan
  - Có thể diễn tra một cách hình thức, trang trọng hoặc thoải mái, tự nhiên tùy tình huống
  - Thông thường, những thiết kế, ý tưởng tốt nhất sẽ được kết hợp sử dụng

Qui trình kiểm thử tính tiện dụng

- Định dạng Test Plan
- Chọn lựa chức năng kiểm thử
- Chọn lựa người sử dụng đại diện
- Đưa ra độ đo và bảng đánh giá
- Tổng hợp các kết quả thực hiện và ý kiến người dùng
- Phân tích kết quả và xác định vấn đề
- Đưa ra góp ý

Qui trình kiểm thử tính tiện dụng

- Định dạng Test Plan
  - Mục đích
  - Phát biểu vấn đề
  - Mục tiêu
  - Hồ sơ người dùng
  - Phương pháp và thiết kế
  - Môi trường và thiết bị
  - Vai trò giám sát
  - Các độ đo
  - Nội dung báo cáo

Qui trình kiểm thử tính tiện dụng

- Chọn lựa chức năng kiểm thử
  - Chọn chức năng người dùng mong muốn
  - Chọn chức năng tiêu biểu, thường dùng

Qui trình kiểm thử tính tiện dụng

- Chọn lựa người sử dụng đại diện
  - Số lượng vừa đủ
  - Đại diện cho nhóm người dùng cuối

Qui trình kiểm thử tính tiện dụng

- Đưa ra độ đo và bảng đánh giá
  - Độ đo
    - Thời gian hoàn thành công việc
    - Số lượng, phần trăm công việc thành công
    - Thời gian truy xuất thông tin
    - Số lựa chọn sai
    - Số lỗi
    - Thời gian phản hồi của hệ thống

Qui trình kiểm thử tính tiện dụng

- Đưa ra độ đo và bảng đánh giá
  - Bảng đánh giá
    - Thang đáng giá
      - Rất không đồng ý, Không đồng ý, Không phản đối, Đồng ý, Rất đồng ý
      - Đơn giản 3 2 1 0 1 2 3 Phức tạp
    - Câu hỏi
      - Liệt kê
      - Rẽ nhánh

Qui trình kiểm thử tính tiện dụng

- Tổng hợp các kết quả thực hiện và ý kiến người dùng
  - Kết quả thực hiện
    - Hiệu năng (Performance)
      - Thời gian trung bình
      - Độ lệch thời gian trung bình
      - Thống kê thời gian phản hồi
    - Tỉ lệ
      - Hoàn tất trong thời gian qui định
      - Hoàn tất không quan tâm thời gian
      - Hoàn tất có sự trợ giúp
      - Lỗi xảy ra

Qui trình kiểm thử tính tiện dụng

- Tổng hợp các kết quả thực hiện và ý kiến người dùng
  - Ý kiến người dùng
    - Câu hỏi dạng lựa chọn
      - Tỉ lệ được chọn
      - Điểm trung bình
    - Câu hỏi dạng trả lời
      - Gom nhóm
    - Nhận xét góp ý
      - Gom nhóm và liệt kê

Qui trình kiểm thử tính tiện dụng

- Phân tích kết quả và xác định vấn đề
  - Chức năng thực hiện thất bại
  - Lỗi, khó khăn từ người dùng
  - Nguồn gốc lỗi
  - Độ ưu tiên lỗi
  - Sự khác nhau giữa các nhóm người dùng

  → Xác định vấn đề

Qui trình kiểm thử tính tiện dụng

- Đưa ra góp ý
  - Đối tượng là sản phẩm, không phải người dùng
  - Ưu tiên khả năng thực hiện (performance) hơn ý kiến chủ quan (preference)

Nội dung

- Kiểm thử giao diện
- Kiểm thử tiện dụng
- **Công cụ kiểm thử giao diện web**

Checklist testing

- Danh sách các câu hỏi để kiểm tra giao diện
- Lỗi giao diện
  - Data validation: kiểm tra giá trị nhập
  - Incorrect field default: dữ liệu mặc định không hợp lệ
  - Mishandling of server process failures: xử lý khi có lỗi ở server
  - Mandatory fields: kiểm tra trường bắt buộc nhập
  - …

Link checker tools

- Kiểm tra lỗi broken link của website
- Broken link là liên kết đã bị xóa hoặc không tồn tại
  → trả về trang 404 error page
- Ảnh hưởng
  - Người dùng truy cập
  - Cổ máy tìm kiếm

Link checker tools

- Công cụ trực tuyến
  - Chạy trực tuyến, không cần cài đặt
  - Sử dụng đơn giản, đôi khi chậm
- Công cụ cài đặt
  - Cài đặt, cấu hình trên máy cục bộ
  - Cài đặt cấu hình khó, chạy tương đối nhanh

Link checker tools

- Công cụ trực tuyến
  - W3C Link checker
    http://validator.w3.org/checklink
  - Google Webmaster Tools
    http://www.google.com/webmasters/tools/

Link checker tools

- Công cụ cài đặt
  - Linkchecker
    http://sourceforge.net/projects/linkchecker/
  - Xenu's Link Sleuth
    http://home.snafu.de/tilman/xenulink.html

W3C Link checker

Xenu's Link Sleuth

HTML, CSS validator tools

- Kiểm tra độ chuẩn của mã HTML, CSS
- Cần hiểu sự khác nhau giữa các chuẩn

Line 64, Column 84: **required attribute "alt" not specified**

```
…  src ="http://mycraft.apcs.vn/mc/mod/mycraft_theme/graphics/Site_title.png" />
```

The attribute given above is required for an element that you've used, but you have omitted it. For instance, in most HTML and XHTML document types the "type" attribute is required on the "script" element and the "alt" attribute is required for the "img" element.

Typical values for `type` are `type="text/css"` for `<style>` and `type="text/javascript"` for `<script>`.

HTML, CSS validator tools

- Công cụ trực tuyến
  - validator.w3.org → kiểm tra chuẩn html
    http://validator.w3.org/
  - jigsaw.w3.org → kiểm tra chuẩn css
    http://jigsaw.w3.org/css-validator/
  - validome.org → HTML, XML, DTD-Schema, RSS
    http://www.validome.org/
  - xhtml-css.com → HTML, CSS
    http://xhtml-css.com/

HTML, CSS validator tools

- Công cụ Firefox Plugins
  - HTML Validator
    http://users.skynet.be/mgueury/mozilla/
  - Total Validator
    http://www.totalvalidator.com/tools/extension.html
  - Validaty
    http://gemal.dk/mozilla/validaty.html

Cross browser testing tools

- Kiểm tra sự tương thích các trình duyệt
- Sự tương thích
  - Các trình duyệt khác nhau
  - Các phiên bản khác nhau
  - Các hệ điều hành khác nhau
- Nên cài đặt trình duyệt thực tế

Cross browser testing tools

- Công cụ trực tuyến → chậm, không tiện dụng
  - BrowserShots
    http://browsershots.org/
  - Adobe BrowserLab

Cross browser testing tools

- Công cụ cài đặt
  - Microsoft Expression Web 4
  - IETester
  - Lunascape Browser
    http://www.lunascape.tv/

Thảo luận