Topics in Website Testing

How to test a website

- Easiest way to start is by treating the web site as a black box.
  - Look at a sample website such as www.apple.com to get a sense of the scale of such an endeavor.
  - Treat each page as a state with hyperlinks as state transitions.

Website text

- Web page text should be treated like documentation and tested using the techniques we described previously.
- Check for …
  - correctness of contact information e.g., phone numbers, addresses
  - correctness of dates and copyright notices
  - title bar text, bookmark text on browser's favorites
  - correctness of the ALT text (i.e., mouse over text)
  - layout issues when browser window is resized
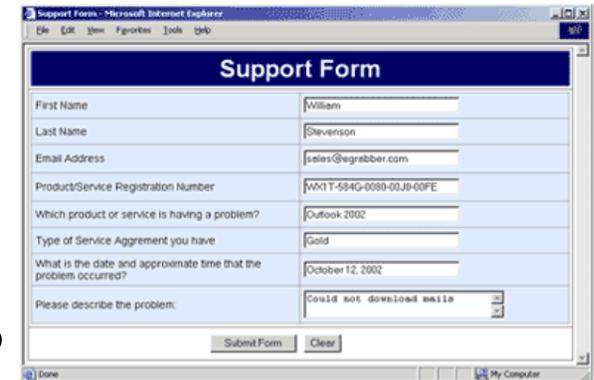
Website hyperlinks

- Each link should be checked to make sure it jumps to the correct destination or website.
- Ensure that hyperlinks are obvious:
  - E.g., underlined text, mouse pointer changes
- If the link opens an e-mail message, test it
  - Send an e-mail and verify that you get a response.
- Check for *orphan pages* that are part of the website but cannot be accessed through a hyperlink.
  - Someone forgot to create the link
  - Might be intentional … Google will find it, though

Website graphics

- Do all graphics load and display properly?
- Is a graphic missing or incorrectly named?
- Does the website intermix text and graphics?
  - Does the text wrap around the graphics?
  - What happens when the browser window is re-sized?
- Does the page load fast enough?
  - Are there too many graphics?
  - Did you try to test the website on a dialup connection instead of a high-speed LAN?

Website forms

- Forms are the text boxes, list boxes, and other fields for entering and selecting information on the web page.
- Are the form fields positioned properly?
- Are the fields the correct size?
- Do they accept correct data?
- Do they reject bad data?
- Are optional fields really optional?
- A favorite entry point for buffer overflow attacks.

"Grey-box" testing

- A mixture of white-box and black-box testing.
- You stick to black-box testing primarily and supplement it by taking a peek at the HTML to figure out how the website works.
- For website testing it is worth looking at the HTML code.
  - It's there, easy to look at, why not?
- Not looking at the HTML code is wasteful, especially since HTML is such a simple language.
  - HTML is a tagging language for text and graphics.
  - To create dynamic web content requires that HTML be supplemented by programming code (e.g., Java applets, ActiveX, VBScript, CGI, Perl).

White-box testing

- To make sure you find the important bugs you should have some knowledge of the website's programming:

  – Dynamic content

  – Database-driven web pages

  – Programmatically created web pages

  – Server performance and loading and security

Dynamic content

- Dynamic content is graphics and text that changes based on certain conditions.
  - E.g., time of day, weather, stock tickers
- Client-side programming involve embedding scripting code (e.g., JavaScript) into HTML.
  - Less efficient if the computations are expensive
  - Code is local, easy to access for testing
- Server-side programming is code located on the server.
  - More efficient for intensive calculations
  - Requires access to the web server to view the code (might be a problem for testing).

Database-driven web pages

- Most E-commerce that show catalogs or inventories are database driven.
  - E.g., Amazon
- Data is pulled from the database, formatted into HTML and sent to the web browser for viewing.A three tier architecture is used:
  - Tier 1: web browsers (presentation)
  - Tier 2: web page formatter (converts data from Tier 3 to HTML)
  - Tier 3: database (queried by Tier 2)

Programmatically- created web pages

- HTML or code is generated by a program.
  - E.g., a web designer may drag and drop elements in a layout program, press a button, and get an HTML page.
- Compilers do this all the time, except they translate high-level language code (Java source code) into low-level code that can be executed natively or on a VM (e.g., byte code).
- Testing these systems is like testing a compiler …

Discussion …

- How would you test a compiler?
  - Black-box?
  - White-box? What are the coverage criteria?
  - Performance?
  - Correctness?

Server performance loading and security

- Popular websites can receive millions of hits per day.
    - www.youtube.com hits 100 Million Videos per day
    - Check out http://www.youtube.com/watch?v=jP26ZYhQktY for a software testing video!
- Each hit requires a download of data from the website's server to the browser's computer.
- You need to simulate millions of connections and downloads to test a system for performance and loading.
- Security issues include:
    - Denial of service attacks
    - Buffer overflow attacks

Configuration and compatibility testing

- Hardware platform
  - Mac, PC, smartphone?
- Browser software version
  - Firefox, IE, Chrome, Safari
- Browser plug-ins
  - To play specific types of audio or video files
- Browser options
  - Security options, ALT text, plug-in, pop ups
- Video resolution and color depth
  - 640x480, 800x600, 1024x768, 1280x1024, 256 colors, 16 colors
- Text size
  - Small fonts, medium fonts, large fonts
- Connection speed
  - DSL, modems of varying speed.

Usability testing:
Jacob Nielsen's top 10 mistakes in web design

- Gratuitous use of bleeding-edge technology
- Scrolling text, marquees, and constantly running animations
- Long scrolling pages
- Non-standard link colors
- Outdated information (need website maintenance)
- Overly long download times (less than 10 sec to load)
- Lack of navigation support
- Orphan pages
- Complex website addresses (URLs)
- Using Frames (just open another window …)

Website testing tools

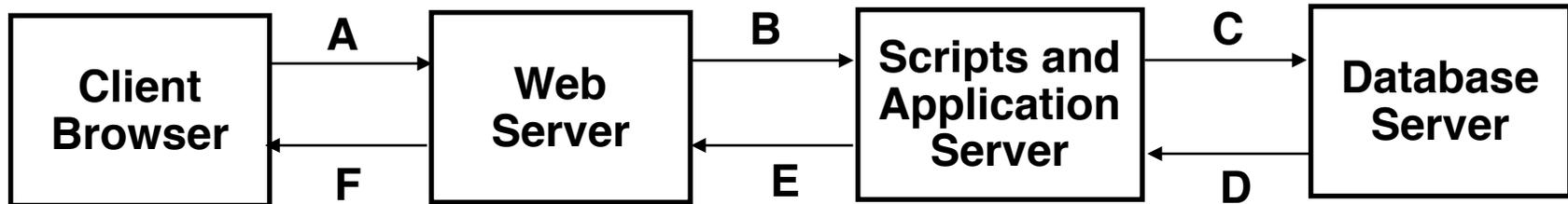- Check out www.netmechanic.com
- Website testing is very labor-intensive.
- Tools that automatically check websites for:
  - Browser compatibility
  - Performance problems
  - Broken hyperlinks
  - HTML standard adherence
  - Spelling on text

Web applications

- Website - requests are navigational requests.

- Web Application - requests can affect the state of the business logic on the server.

A web application

```
┌──────────┐   A    ┌──────────┐   B    ┌──────────────┐   C    ┌──────────┐
│  Client  │ ─────→ │   Web    │ ─────→ │ Scripts and  │ ─────→ │ Database │
│ Browser  │        │  Server  │        │ Application  │        │  Server  │
│          │ ←───── │          │ ←───── │   Server     │ ←───── │          │
└──────────┘   F    └──────────┘   E    └──────────────┘   D    └──────────┘
```

A: Request

B: Request for page generation

C: SQL command

D: Result set

E: Dynamically generated page

F: Response (dynamic I static page)

Web application failures

A study by the Business Internet Group of San Francisco found that:

- 72% (29/40) leading e-commerce sites, and
- 68% (28/41) government sites contained Web application failures
  - 25 "technical errors"
    - E.g., page not found, multiple attempts to subscribe to a service
  - 3 data errors
    - E.g., page without text, wrong page returned

What type of bugs?

Study by Kallepalli and Tian

- Analyzed the logs of the Web pages of the School of Engineering and Applied Science at Southern Methodist University for usage and failure information.
- Looked at data from 26 consecutive days.
- Total number of "hits" 762,971 (~30,000 hits/day)
  - A "hit" is registered for each Web page if one of the following happens:
    - The HTML file corresponding to a page is requested.
    - Any graphics within the HTML page is requested.

Bug types

Permission denied

No such file or directory

Stale NFS handle

Client denied by server configuration

File does not exist

Invalid method in request

Invalid URL in request connection

Mod_mime_magic

Request failed

Script not found or unable to start

Connection reset by peer

Distribution of bugs

Permission denied	2079
No such file or directory	14
Stale NFS handle	4
Client denied by server configuration	2
File does not exist	28631
Invalid method in request	0
Invalid URL in request connection	1
Mod_mime_magic	1
Request failed	1
Script not found or unable to start	27
Connection reset by peer	0