Planning for front-end User Testing

1. Why plan testing now?
   1. You get additional clarity on scope
      1. Knowing which browsers and devices you’ll test for helps you focus and saves time
   2. Less explaining (and argument) later with the client
      1. Setting expectations now for what you’ll test reduces misunderstanding later
   3. Less stress before, during and after lunch
      1. If you’ve thoroughly tested everything, u can be confident everything displays and performs as expected.
2. You can’t test everything
   1. Realistic about what matters most
      1. Budgets are finite.
         1. If you’re working for a client, you’re working with a budget
      2. Schedules are finite
         1. If testing isn’t on the schedule, it’ll be rushed or left out
      3. Not every scenario matters
         1. Your website may not need to work on hundreds of devices
3. What are we testing for?
   1. Identify who you’re testing for
      1. Focus on scenarios most relevant to your audience:
         1. Most popular devices they’ll use?
            1. Iphone5, ipad2+, desktop (1024 pixels and up)
         2. What OS/browser combinations are most popular among your audience?
            1. Windows 8 internet explorer, OS X safari, IOS X Firefox.
         3. What connection speeds do they have
            1. 3g, 4g, broadband?
         4. How tech-savy is your demographic?
            1. Women between 18-35 fairly tech-savy, smartphone owner, social media user
      2. Sites for statistical information
         1. W3 schools
         2. Net market share
         3. Stat counter
   2. Prioritize browser & device support
      1. Fully supported browsers & devices
         1. All content must be readable
         2. All functionality must work properly
         3. Minimal deviation from approved UI design
            1. Consistency with presentation
      2. Partially supported browsers & devices
         1. All content must be readable
         2. Navigation must work
         3. Login functionality must degrade gracefully
         4. Degradation in the UI cannot obscure content
      3. Unsupported browsers & devices
         1. No support or testing will be performed
   3. Decide what to test
      1. Functionality testing
         1. Includes all the links in web pages, forms used in the web pages for submitting or getting information from users, cookie testing and database connections
         2. Test all your links:
            1. Test all internal and outgoing links
            2. Test links that jump within single pages
            3. Test links used to send email from web pages
            4. Test to check for orphan pages (no inbound links)
         3. Test forms on all pages:
            1. Check validations on each field
            2. Check default values of fields
            3. Test correct input into form fields
            4. Options to create, delete, view or modify forms
         4. Cookies
            1. Small files that are stored on your device

Sessions

Log in sessions

* + - * 1. Test functionality and security by enabling/disabling cookies in your browser
        2. Ensure cookies are encrypted before writing to the user’s device
        3. For session-based cookies, check login sessions and user stats
      1. Test CSS/HTML validation
         1. Validate the site for HTML syntax errors

WC3 markup Validation service

* + - * 1. Check that the site is crawl able to search engines

Google webmaster tools

* + - 1. Test database connections and consistency:
         1. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality
         2. Ensure all database queries are executing correctly
         3. Ensure data is retrieved correctly
         4. Ensure data is updated correctly
    1. Usability testing
       1. Covers navigation, content, presentation and task success
       2. Several methods:
          1. Survey Monkey
          2. Videotaped, moderated user sessions
          3. Online services such as:

User testing

Loop11

Optimizely

* + - 1. Ease and effectiveness of navigation:
         1. Do users find what they need easily
         2. How long does it take them to find what they’re looking for?
         3. Does the structure and organization of the navigation match the user’s expectation?
         4. Are links labeled with terms that make sense to users?
      2. Usefulness of content:
         1. Does content match what your users want and need?
         2. Can users find & read the content they need?
         3. Do they understand the content?
         4. Can they act on the content?
      3. Test the effectiveness of visual presentation (UI)
         1. Do UI elements clearly separate navigation from content?
         2. Do UI elements distract or create a barrier between the user and the content?
         3. Do UI elements clearly signal what can be acted upon or interacted with and how to do so?
         4. Do all UI elements work together to guide the user through a particular process or task flow?
      4. Task success:
         1. Were users able to accomplish the key task they came to the site to perform?
         2. Satisfied? Angry or frustrated?
         3. What prevented them from succeeding?
         4. What’s the consequence of each task failure – lost customers, increased support costs, inaccurate data captured?
    1. Errors and Exception testing
       1. Covers what happens when things go wrong
          1. A user enters bad info, stops a process, the system crashes etc.
       2. Test that interactions between servers are executed properly
       3. Test that errors are handled properly
       4. If the database or web server returns an, error, make sure error messages are displayed appropriately to users.
       5. Make sure error messages are understandable and instructive
       6. Test for transaction interruption or lost connection to a web server
    2. Compatibility testing
       1. What happens when the site is used across different devices, browsers, platforms, along with specific user scenarios like mobile browsing or printing.
          1. Browser compatibility

Does the website display consistently in different browsers?

Does the website display consistently in the same browser across different OS platforms?

Does all functionality (links, forms, etc. ) work properly across different browsers?

List of online tools that simplify testing:

http://Mashable.com/2014/02/06/browser-testing-tools/

* + - * 1. OS & device compatibility

Does all functionality (links, forms, etc. ) work properly across different browsers?

Web service calls or API functions work properly across all OS platforms and devices?

Do fonts or CSS styles display properly across all OS platforms and devices?

* + 1. Performance testing
       1. How well the website can accommodate heavy user activity. Performance testing should include:
          1. Web load testing

What happens when a high volume of users are accessing or requesting the same page? The site should be able to handle simultaneous user requests, large input data from users, simultaneous database connections, etc.

* + - * 1. Web stress testing

What happens when the site is stretched beyond its specification limits? How does the site and the server reacts to stress and recovers form crashes? Stress is generally tested on input fields, login processes and sign up areas.

Input fields, login processes, signup

* + 1. Security testing
       1. Measures how well the website and the user, product or transactional data it captures and transmits is protected against hackers.
          1. Paste an internal url (https) directly into the browser’s address bar without logging in. internal pages should not open.
          2. Try changing url options directly while logged in with a specific User ID. Eg. If you’re checking site statistics with publisher site ID=123. Try directly changing that ID parameter to a different site ID. Access should be denied
          3. Try some invalid entries in input fields like login username, password, or input text boxes. Check the system reaction
          4. Web directories or files should not be directly accessible unless a specific download option exists in the UI
          5. Test whether SSl is used for security measures. If used, you should see a warning when switching form non-secure (http) pages to secure (https) pages and vice versa
          6. Check for logs of all transactions, error messages and security breach attempts on your web server (usually via CP)

1. Takeaways:
   1. You need to know three things up front:
      1. What is worth doing?
      2. What are we creating?
      3. What value does it provide?
   2. If you ever hear the words “we have no competition.”
      1. That’s your cue to exit
   3. The most important principle of good UI/UX is progressive disclosure.
      1. Information presented to someone who isn’t interested in it or isn’t ready to process it is noise
   4. Three kinds of requirements things people:
      1. Say they need
      2. Actually need
      3. Don’t know they need
   5. When you need to get requirements quickly
      1. Use scenarios
      2. Are your best friend in the world
   6. Documentation eliminates
      1. Excuses and surprises
   7. When it comes to interacting with clients, partners or fellow team members:
      1. Silence equals agreement
   8. If you’re doing it without the list,
      1. You’re guessing
   9. Be realistic about what matters most:
      1. You can’t test everything