

Application for Summer of Code 2008: Jimmy Berry – Usability Testing Suite¹

Purpose

My project will provide a method of measuring the user experience² which has been stated as a very important goal of Drupal. In Dries' keynote he detailed the results of the usability testing^{3 4} and stressed that we should “Continue to make Drupal easier to use.”⁵ My project will provide a suite of usability testing tools that will allow data to be recorded and analyzed in order to make improvements to Drupal's usability.

Project

My project is composed of two parts that will combine to provide a powerful usability testing suite. The first is a usability test module that provides an API for plug-ins of various kinds. The second part of my project will be to develop a full set of plug-ins that will allow the usability engineers to take full advantage of the the usability suite.

The usability testing suite will allow a test designer to create “usability unit tests.” A usability unit test will consist of eligibility requirements, scenario, tasks, feedback questionnaires, and data capture requirements.

A test can be conducted from any location that has been setup with the necessary client side data capturing software as specified by the test. Data capture requirements can range from screen capture with voice to the bare minimum drupal log. The test will confirm that a user has the necessary skills or knowledge, or eligibility, to participate in the test and will provide them with the scenario and a list of tasks to complete. The user will then proceed to complete the tasks while the usability testing suite captures the requested data and feedback is captured.

1 <http://groups.drupal.org/node/9661>

2 <http://groups.drupal.org/node/9252>

3 <http://buytaert.net/first-results-from-usability-testing>

4 <http://groups.drupal.org/node/9339>

5 <http://www.lullabot.com/blog/drupalcon-state-dries-buytaert>

An evaluator can then review collected data through the usability testing module itself or external tools. The data will be stored in such a way that it can be easily correlated to a specific task, evaluator, and step. The module may need to perform post-processing operations in order to clean up collected data and present it in a form that makes sense to the usability testing engineer. Possible post-processing might include merging a screen or video recording with audio or text feedback, like subtitles in a movie. Another important evaluation tool that will be available are click heatmaps that will give an evaluator a basic idea of how the user interacted with the layout.

Plug-ins will use the API provided by the usability testing module in order to collect data and export it for use in external tools. The plug-ins will provide mouse tracking and click heat maps, user paths, form submissions, audio (or text fallback), screencast, videocam, and database snapshots. This data will then be merged into a test record that will give a usability engineer a clear picture of the user's experience even when in a remote location. The extensible API will allow for the data collection, exportation, post-processing, and dynamic event system to be enhanced.

The data collected during the tests will be stored in various manners depending on factors like bandwidth. Screencasts, videocams, and audio data may be stored on the tester's machine and uploaded to the testing server once the test has been completed. The data collected/submitted from the tests could be stored on the testing server, uploaded to a data server after collection, or possibly e-mailed to an evaluator. The system for data exportation and storage will use a system of plug-ins that will allow it to expand as needed.

Plug-in API

The usability suite will provide an API for data collection, data exportation, and dynamic events. The API will allow different tools and techniques to be added as needed.

The data API will provide a set of hooks that will inform a data collection plug-in of test events such as test beginning, test ending, and task or objective completion. Relational information such as a common timestamp, test id, and user id will be provided to the

plug-in. The API will require hooks to be implemented that will request data from a completed test in order to reference it with other data. The hooks system will provide hooks for requesting data from specific steps, after certain events, or other triggers that will be based on timestamps stored by the usability suite.

An API for exporting data based on filters and formats will also be provided. The API will provide hooks for setting filters and requesting data that is managed by the usability suite. The API will allow for data to be exported in formats such as CSV, XML, compressed or any other useful format.

Benefits

1. "Measure the UX of Drupal"⁶
2. Provide much needed feedback on the usability of Drupal in an easy and consistent manor that can be conducted repeatedly with little overhead.
3. Facilitate usability changes based on evidence and testing instead of speculation and developer preferences.
4. Allow user experience professionals to easily and cheaply test user interfaces and allow prototypes and changes to be tested before going live.

Beyond these direct benefits Drupal will benefit as a whole since more users and fewer questions follow increased usability.

6 <http://groups.drupal.org/node/9252>

Deliverables

- Usability Testing Suite
 - Interface for setting up unit tests and managing test results
 - User test environment that provides triggers and data hooks
 - Pluggable API for input, export, post-processing, and analysis of test data
- Initial set of input plug-ins
 - Data collection
 - Click Heatmap⁷
 - Live text feedback through testing environment
 - Drupal log during test
 - Data exportation
 - XML for unit test description and setup, or test results with references to accompanying data files
 - CSV for click data

Roadmap

1. Design and implement usability unit test creation and review interface, and provide database structure for usability unit tests. (June 14th)
2. Create user testing environment and event capture interface. (June 28th)
3. Design and implement pluggable API for collection of data. (July 12th)
4. Create input plug-ins for necessary data. (July 26th)
 1. Drupal log
 2. Click Heatmap
 3. Live text
5. Design and implement pluggable API for export and post-processing of data. (August 9th)
6. Provide documentation on how to use usability testing suite. (August 18th)

⁷ http://drupal.org/project/click_heatmap

About

I am senior in high school in Omaha, Nebraska. I rank in the top 5% of my class and plan to attend the prestigious Peter Kiewit Institute at the University of Nebraska at Omaha with a full tuition scholarship. I have competed in and won several programming competitions.

My father introduced me to programming and during my early teen years I taught myself how to use HTML, CSS, JavaScript, and PHP. I created several sites in my spare time and eventually created a site for hire. I went on to improve my skills and, with my dad, later completed a large retirement benefits management site in PHP that we later ported to C# .NET.

My friend and I designed, wrote, and tested a game engine written in Java. We learned an enormous amount about coding, working together, and game development. We wrote and refactored over 20,000 lines of code before deciding that Java would be too slow for our purposes.

Qualifications

I was introduced to Drupal through the GHOP (Google Highly Open Participation) Contest. The contest provided a way to gain credibility while completing "small" tasks that had a clear finishing point. Since then I have gone on to create server modules, one of which, Click Heatmap⁸, will become the first input plug-in for the usability suite. I have also become heavily involved in the push to get full testing coverage for Drupal 7 and recently became a co-maintainer of the SimpleTest⁹ module.

You can see my activity by visiting the tracker: <http://drupal.org/user/214218>.

⁸ http://drupal.org/project/click_heatmap

⁹ <http://drupal.org/project/simpletest>