

# Usability Testing Suite

Summer of Code 2009

Jimmy Berry

## Purpose

The Usability Testing Suite<sup>1</sup> that I created during SoC 2008<sup>2</sup> needs additional work in order to make it ready for use<sup>3</sup>. The existing module provides a powerful API that makes writing data collection plug-ins simple, but the user interface needs refinement and a screen recording plug-in would make the module much more helpful. Through this project I intend to finalize the Usability Testing Suite and make it ready for widespread use.

## Project

The project consists of four main parts: 1) enhance the study and analysis user interfaces, 2) create a basic screen recording plug-in, 3) design and implement a multivariate task type, 4) provide access to back-end tools and features through the user interface so the Usability Testing Suite can be used to setup Drupal for testing with external systems. In addition to these main goals any bugs and minor changes that are deemed useful will be made.

## *Study interface*

In order to make the Usability Testing Suite easier to use a number of interface and workflow changes will be implemented as described in the issue queue<sup>4</sup>. The changes include the addition of contextual help in a few key areas<sup>5</sup> and the creation of wizards to help guide users through the initial setup.

---

1 <http://drupal.org/project/uts>

2 <http://groups.drupal.org/node/11011>

3 <http://groups.drupal.org/node/20416>

4 <http://drupal.org/project/issues/uts>

5 [http://drupal.org/files/issues/UTS\\_create\\_study.jpg](http://drupal.org/files/issues/UTS_create_study.jpg)

## ***Analysis interface***

The ability of the Usability Testing Suite to summarize information and help the person analyzing the data to come to a logical conclusion is key to the project. In order to make analysis easier a number of dashboards and summary screens will be created<sup>6</sup> in addition to the main analysis dashboard<sup>7</sup>.

## ***Screen recording plug-in***

A screen recording plug-in is undoubtedly the most useful tool that provides a very detailed view of what the participant did during the study. There are a number of tools available at relatively low or no cost. An example of such a tool is userfly<sup>8</sup> which provides a JavaScript library to record user actions and generate a *video* that can be replayed at any time.

Research will be conducted to decide if any of the available tools provide the necessary features and fit into the Usability Testing Suite. If a suitable tool is found then an integration plug-in will be written. Otherwise the feasibility of writing such a tool will need to be discussed and a decision reached.

## ***Multivariate testing***

Being able to test variable interfaces on a live site is a very powerful research tool. An API will be created that makes it possible to design a number of interfaces and have the system switch them out at random and record user interactions with the interface using the standard data collection plug-ins. The API will be built on the existing Drupal theme layer to provide a consistent experience and require minimal code. The analysis and dashboard screens will need to be enhanced to allow viewing of data from either testing method.

---

6 <http://drupal.org/node/308140>

7 <http://drupal.org/node/307946>

8 <http://userfly.com/>

## **Tools**

The Usability Testing Suite has a number of useful tools that are *hidden* inside the code base and are not accessible as standalone features. One of the most useful is the tool that creates Drupal environments for testing. The tool can be extended to allow for batch creation of environments through standalone interface which can then be used for studies conducted with the Usability Testing Suite or external. Creating Drupal test environments at the usability studies that have been conducted at universities has been noted as something that could use to be automated.

## **Benefits**

The benefits of finishing the Usability Testing Suite are quite compelling.

1. "Measure the UX of Drupal"<sup>9</sup>
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.

---

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

## Deliverables

By the end of the project the Usability Testing Suite will be ready for prime-time use and provide a powerful tool for testing the usability of Drupal.

- Enhanced study and analysis interfaces
  - Wizard to guide users through initial setup
  - Additional analysis summary and dashboard screens
- Basic screen recording plug-in (based on research and decision)
- Multivariate task type and API
  - Test variables interfaces on a live site
  - Provide a simple API that builds on the theme system
- Expose interface for creating Drupal environments

## Roadmap

1. Discuss and enhance study and analysis interfaces. (June 13)
2. Complete research of screen recording tools and come to a conclusion. (June 20)
3. Implement screen recording solution found during research. (July 11)
4. Collect use-case information and design API for multivariate testing. (July 18)
5. Implement multivariate testing API and analysis/dashboard improvements. (August 8)
6. Expose interface for creating Drupal environments. (August 15)

## About

I currently attend the University of Nebraska at Omaha where I am working towards of BS in computer science. When I am not in school, among other things, I work on Drupal. During the year and a half since GHOP (Google Hightly Open Participation) 2007-2008 I have become heavily involved in the Drupal community.

Most notable is my work enhancing the Drupal testing framework and automated testing system. After successfully writing a number of tests through GHOP I went on to become a co-maintainer of SimpleTest<sup>10</sup>. I made heavy modifications and enhancements to the module itself while filling out the core testing suite and managing the process<sup>11</sup>.

After years of development and discussion surrounding testing and more specifically SimpleTest being added to core I was a part of making it happen. After a coding sprint in Paris SimpleTest was committed to core<sup>12</sup>. In addition to maintaining the SimpleTest API in core I took over the effort to create and integrate an automated testing system into the drupal.org project issue queue.

The system has been running successfully for for some time<sup>13</sup> now and has greatly enhanced the core development cycle and Drupal 7 core itself<sup>14</sup>. I am currently finishing up a second generation of the system that will provide a much enhanced feature set.

In addition to testing, both usability and code, I have created a number of other modules and work on projects to develop sites using Drupal. A detailed look at my contributions can bee seen by looking at my use page and tracker, <http://drupal.org/user/214218>.

---

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

11 <http://groups.drupal.org/node/9408>

12 <http://blog.boombatower.com/success-at-paris-coding-sprint>

13 <http://testing.drupal.org/node/4>

14 <http://boombatower.com/news/automated-testing-changing-landscape-drupal>