M2.16 – Localised unit testing mechanism

Drupal SimpleTest

Scratchpads are based on the Drupal[1] Content Management System (CMS), which has enabled us to take advantage of the tools that have been developed specifically for it. One such tool is the SimpleTest[2] module, which is now included as a part of Drupal 7. The SimpleTest module allows a programmer to write unit tests for their code. For Scratchpads, this means that we can run tests on specific modules, or we can run tests together with other modules. This gives us a facility for testing all new code, whether it be code submitted by an external project (e.g. Antkey), or code added to our Git repository[3] by a ViBRANT partner.

Tests can be run manually by any Scratchpads developer, or they can be automated. As part of D2.4 – Unit testing[4], we will be developing an automated test environment. The automated test environment will test each piece of code that is committed and pushed to our Git repository. This could be done at regular intervals (e.g. once every four hours), or after x number of commits. Drupal.org already has its own test environment, which in turn has a public facing website "Drupal Quality Assurance"[5]. It is likely that the environment we develop will be based on Drupal's, but with features developed specifically for Scratchpads code and tight integration with Aegir, our Drupal site management tool.

Two very basic unit tests have been written for components of the Scratchpads, specifically the ITIS Term module, and the Scratchpads Tweaks module. These tests have been written to ensure that the basic principles of what we are attempting are correct.

Selenium

Using the SimpleTest module, as described above, allows us to test the server-side components of Scratchpads, but it does not allow us to test the client-side components. This basically means we still need to test all of the Scratchpads features, manually, in multiple browsers. It may be possible to integrate Selenium[6] into our unit tests work flow. Selenium will make it possible to write client-side tests for Scratchpads, which in turn could be automatically tested using our test environment. By doing this work, we not only meet D2.4, but we will also go some way to reduce the workload involved in sustaining Scratchpads, which helps us to meet D2.3 – Financial sustainability[7].

---

1 http://drupal.org/
2 http://drupal.org/project/simpletest
3 http://git.scratchpads.eu/
4 http://vbrant.eu/content/d24-unit-testing
5 http://qa.drupal.org/
6 http://seleniumhq.org/
7 http://vbrant.eu/content/d23-financial-sustainability