elementary Testing

Introduction

The purpose of this document is to investigate all options for automated testing for elementary applications and libraries.

Possible Unit Test Frameworks

Vala

- GLib.Test (a.k.a. GTest)
    - The GNOME Documentation site has a very simple test setup: https://live.gnome.org/Vala/TestSample
    - Interesting article on implementing GLib.Test in Vala: http://esite.ch/2012/06/26/writing-tests-for-vala/
    - Using a [Test] attribute to automate test setup: http://asabil.wordpress.com/2009/01/23/vala-and-unit-testing/
    - Used by Libgee: https://github.com/GNOME/libgee/tree/master/tests
    - Mailing list item on GTest: http://www.mail-archive.com/gtk-devel-list@gnome.org/msg06994.html
- Gtk.Test (Gtk3)
    - Tests implemented in GTK: https://git.gnome.org/browse/gtk+/tree/gtk/tests
- Valadate (forked by Yorba)
    - Yorba wiki page on Automated Testing for Shotwell: http://redmine.yorba.org/projects/shotwell/wiki/AutomatedTesting
- vunit and VUnit (old and unmaintained)

Other languages and interesting libraries

- Dogtail; Gtk+ testing in Python (seems interesting)
- Pytest; Python Unit Testing framework
- JUnit; Java Unit Testing framework
- PHPUnit; PHP Unit Testing framework
- xUnit.net; an open source framework for C#, F#, VB.NET, etc. Might be a good starting point for rolling a Vala unit testing framework. (Also a good resource for general unit testing knowledge and practices)

BDD Testing Frameworks

Behavorial testing also seems interesting for us, as we are focussing a lot on user interface and experience. It would also enable us to “spec out” our applications in a more human readable way, which would be great for our designers.

Vala

None yet (?)

Other languages

- Cucumber (Ruby)
- Behat (PHP)
- pytest-bdd (Python)
- Lettuce (Python)

Language-independent

- https://launchpad.net/xpresser - framework to send X events to buttons and compare the window state with a predefined bihttp://esite.ch/2012/06/26/writing-tests-for-vala/tmap
- ATK-based GUI testing - LDTP, Mago
- Autopilot - GUI testing library developed by Canonical that works via plugin to the the toolkit, pretty much like GtkParasite does, and allows deep inspection of widget tree and widget properties. Ubuntu tests for GTK apps can be found here. The GTK plugin is somewhat buggy, but it's usable and we can expect cooperation from Ubuntu on improving it.

Integrating testing

We are currently using cmake as our build tool, and it already supports running tests, using CTest.

Testing also could be done on the packaging level, to ensure that the code works on the specific package versions used in the target distros (elementary OS in our case).
Ubuntu uses autopkgtests mechanism for that. Notably, they host hackfests to tutor people new to it. For more info see Martin Pitt's ubuntu dev week session on automated testing, skip to "== Packages =="

Developer documentation and website

If we are going through with this, it is probably a good idea add some testing documentation to the website. And if we choose a framework, it is also a good idea to feature our framework of choice on our website.

Example implementations

- Dane Henson already has a really basic implementation that uses GLib.Test: http://bazaar.launchpad.net/~thegreatdane/+junk/agenda-tests/files/head:/