

Linux distribution testing and quality assurance: current developments of the EDOS project and open questions

François Déchelle (EDGE-IT, fdechelle@mandriva.com)

Stéphane Laurière (EDGE-IT, slauriere@mandriva.com)

Paulo Trezentos (Caixa Mágica, ADETTI, ISCTE, Paulo.Trezentos@iscte.pt)

Mário Morgado (Caixa Mágica, ADETTI, ISCTE, mario.morgado@caixamagica.pt)

Laurent Godard (NUXEO, lgodard@indesko.com)

February 12, 2007

1 Introduction

This presentation will describe the developments done within the EDOS project regarding Linux distribution testing and quality assurance process. It will also point out remaining open questions that could be subjects for further research and which could bring significant advances for Linux distribution developers.

Linux distribution quality assurance process is a difficult task, that is either delegated to an open community who is in charge of installing and trying a candidate distribution (Debian testing, Mandriva Cooker...) or performed by dedicated in-house and expensive staff doing repetitive testing. As the number of testing units (the distribution packages) is usually large (more than 10,000 packages for most of the distributions), the complexity of the testing task is huge and leads to either reduce system quality by accepting non-tested packages or delay final releases and increase overall system development cost.

2 Description of the EDOS testing framework

One of the goals of the EDOS project is to research and experiment solutions which will allow to reduce the costs and delays of quality assurance in the process of building a GNU/Linux distribution.

The currently experimented solution is a modular architecture that provides tools for testing a Linux distribution, namely testing that the distribution can be installed or upgraded and that the distribution passes tests after being installed or upgraded.

The EDOS testing framework aims to integrate automatic and manual testing tools and to centralize access to test results and testing data, allowing distribution developers, testers, managers... to have an accurate and real-time view of the testing state of the distribution.

The components of the EDOS testing framework are:

- the test and QA web portal, a web portal that centralizes test results, test suites, test metrics and provide a central view of the QA process
- the test runner, running automatic or manual tests and reporting results to QA portal

- TULIP, a framework for testing upgrades of Linux distributions, to ensure that the distribution can be upgraded and that the upgraded distribution is still functional

2.1 Test runner

The EDOS test runner is a software tool that runs tests and provides reporting in a generic way w.r.t. the test suite. Using the EDOS test runner, a Linux distribution tester can run automatic tests suites such as LTP (Linux Test Project[1], a collection of tests that harnesses more than 2900 tests addressing kernel, system calls, networking, inter-process communication...) and report results to the EDOS QA portal without any need to modify the test suite itself.

To achieve this goal, an XML specification format has been designed, inspired by the STAF/STAX test description format. An XML specification of a test wraps the informations that are necessary to run a test and report its results, namely test name, test requirements (necessary packages, user privileges...), harness used to run the test (shell, python...).

A brief example of an XML description file for one test of the LTP test suite is given in Listing 1.

Listing 1: The XML specification of the LTP “crashme” test

```
<?xml version='1.0' encoding='UTF-8'?>
<test>
  <description
    name="crashme"
    suite="LTP">This test runs the LTP crashme test.</description>
  <package name="edos-ltp"/>
  <condition>
    <user name="root"/>
  </condition>
  <test-runner>
    <shell script="yes | /usr/lib/ltp/runltp" shortLog="true">
      <arg value="-q"/>
      <arg value="-f /usr/lib/ltp/runtest/crashme"/>
    </shell>
  </test-runner>
</test>
```

2.2 Test reporting and QA portal

The QA portal is a web portal that provides two levels of functionalities:

- web services for test reporting, accessed by the test runner, that store the test results in the portal database using HTTP POST requests
- user interfaces to browse tests, test reports... and more generally to provide a central view to the data relevant for the QA process

Figure 1 shows one page of the QA portal that display test reports, namely a series of test using the LTP test suite.

The current implementation of the QA portal is based on XWiki[3], a Java based wiki engine that offers several interesting functionalities for the development of complex portals, in particular the Groovy scripting engine that has been used to implement both the web services offered by the portal and the display of test reports.

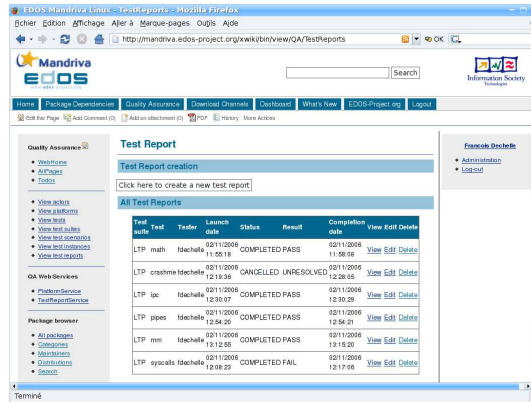


Figure 1: EDOS QA portal displaying test reports

The current state of the QA portal provides a small set of *metrics*, i.e. numeric indicators that quantify the testing state of the distribution, for instance ratio of tests and test suites that have passed. However, this development is still in its early stage.

2.3 TULIP

Unlike proprietary software, Linux distributions are more live processes than frozen products and are upgraded very often, for instance to correct security holes, using the upgrade tools that are provided with almost every distribution (apt-get, urpmi...). However, upgrading a distribution is a risky task and provides no a-priori information about the fact that the upgraded distribution will still be functional.

The TULIP framework purpose is to drive upgrade tests of various Linux distributions to ensure both that the distribution under test can be upgraded and that the result of the upgrade is still functional.

TULIP uses Linux installations on virtual machines (currently Qemu[2]) and drives periodic upgrades of the distribution using the distribution native upgrade tools (apt-get, urpmi...) to detect packages installation failures. Once upgraded, the images are tested by running a collection of automatic tests. Results of the upgrade process and of post-upgrade tests are reported to the QA portal.

2.4 Test assistant and manual testing

As a significant part of the tests cannot be completely automated (for instance USB device manual plug), a test assistant helps the human testers to run check-lists and enter test results. The check-lists are described using an XML specification and the test assistant reports test results to the QA portal. This tool is currently under development at the time of the writing of this paper.

3 Open questions about Linux distributions quality assurance

The EDOS project will likely contribute useful tools for Linux distributions testing, but many points still remain uncovered yet and further research could help improving testing processes and provide new tools.

3.1 Quality metrics

Software quality assurance makes use of metrics, for instance code coverage. Similar approaches should be established for Linux distribution in order to build metrics that could help in managing the complexity inherent to Linux distributions given the large number of testing units. For instance, similar to code coverage, a measurement of packages coverage could be elaborated, that would describe the degree to which the packages of a Linux distributions have been tested.

3.2 Automatic testing

With a number of testing units that is generally above 10,000, manual testing rapidly reaches its limits. Furthermore, testing must be repeated often and re-run after each distribution build. However, automatic testing of some components of a Linux distribution, namely the graphical user interfaces, is still in its infancy and generic GUI testing tools have not reached yet a reasonable level of usability.

4 Conclusion

Quality assurance of Linux distributions is a complex process that has mainly followed an empiric development and that leaves room for improvements. The EDOS project addresses several issues of the quality assurance process, namely running functional and upgrade tests of a Linux distribution in an integrated framework and providing an accurate view of the testing process via the QA portal.

Future developments of the EDOS framework will address :

- integration of testing tools in 'public' distributions, so that a larger community can contribute to the test process
- larger sets of metrics to provide testers an accurate and real-time view of the testing state of the distribution.

Open source has changed the way we think now software development and validation. For instance, the development of Linux distribution does not really follow the now classical scheme of test-driven development nor does it give testers good conditions of certainty and stability. However, despite these non-standard development processes, Linux distributions are stable and reliable products. Exploring and better understanding the distribution building processes would probably open new ways to think about reliable software development and propose new approaches to testing and quality assurance.

References

- [1] Linux test project. <http://ltp.sourceforge.net/>.
- [2] Qemu. <http://fabrice.bellard.free.fr/qemu/>.
- [3] Xwiki. <http://www.xwiki.org/>.