# SystemTestPortal – A Web-Application for managing Manual System Tests

Daniel Kulesz, Ivan Bogicevic

Institute of Software Technology, University of Stuttgart
(daniel.kulesz | ivan.bogicevic)@informatik.uni-stuttgart.de

November 23, 2017

**Abstract**

*While test automation is desirable in every software project, not all tests can be automated well, e.g. testing software that requires specific hardware. Furthermore, manual system tests often unveil unexpected faults simply because humans are more sensitive in this regard. Actually, almost every developer does this kind of manual tests "somehow". However, due to the lack of a proper tool support, these tests happen in an unplanned, unsystematic way and without documentation. As a result, it is hard for users to find out which features of a product were tested in which environment, on which hardware, and with which result.*

*We are working on a novel web application named SystemTestPortal for creating, running and analyzing manual system tests. The tool aims at being useful for developers, testers, test managers, and end-users. It is lightweight, written in Golang, available under the GPLv3 license, and developed in a student project. Many of its planned features are working already, and by the time of FOSDEM 2018 a mostly complete 1.0 version will be available.*

## I. Introduction

One important part of software quality assurance is the software test. By testing, the software is executed under controlled conditions, the results and the behavior of the program are recorded and compared with the expected results. The goal is to find as many issues as possible.

Test cases that can be executed automatically are very helpful. Once they have been developed, their execution basically costs nothing as there is no need for a human tester who executes the tests step by step. Especially for regression testing, automated test cases are an important part of the continuous integration process as they find issues very early on.

## II. The need for manual testing

While automated tests greatly enhance software quality, nearly every software system still needs manual testing in addition. Especially for tests on the system level, tests that check the user interface, and tests that need manual interaction because a human has to decide whether the behavior of the program or its results are acceptable. Although these manual tests are very important, in practice all data of the test cases is not managed very well. In many free and commercial software projects today test data and records of executed test runs are managed using text processors or spreadsheets. Even worse, in many cases these manual tests contain specific needs for hardware or the environment of the software, but these are neither specified exactly nor are they correctly recorded in the test runs.

Due to these issues, it is hard for users to find out which features of a product were tested in which environment, on which hardware and with which result. This is especially painful for users of free software projects that are hardware-related, e.g. projects which provide free firmware distributions for smartphones or routers/firewalls. Often, users waste pre-

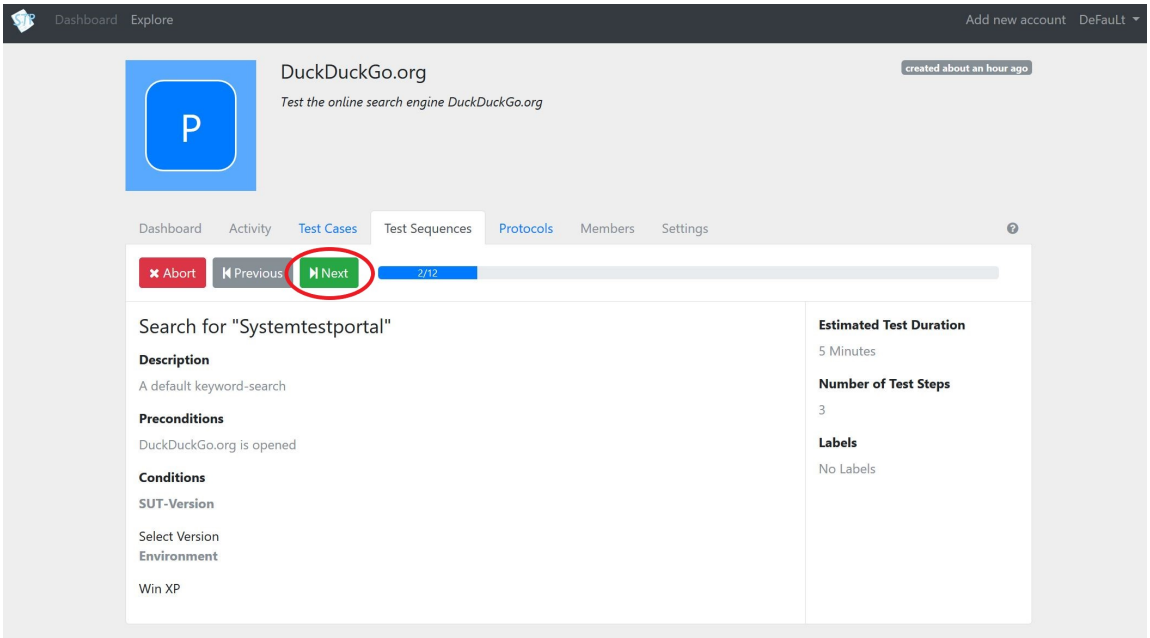


**Figure 1:** *Screenshot of the application executing a test case*

cious time digging through forums or mailing lists, trying to find out which functions actually might work on which devices. They often find outdated information and in many cases it is not clear if a device that was once supported has ever been actually tested with the current version of the software.

## III. Proper tools are missing

We believe that a major cause of the current situation is not documentation laziness but the lack of proper tool support for manual system testing. There are only few (and not very good) commercial tools available and, even worse, there is no established free tool available. This has been a dissatisfying situation for years and we decided to make an attempt to change it for the better. To do so, we initiated an agile study project at the University of Stuttgart and found seven highly motivated software engineering students who took up the challenge.

## IV. System Test Portal

The resulting project started in Spring 2017 and, since then, has steadily developed a web-application named "SystemTestPortal"[1]. SystemTestPortal allows users to create, run and analyze manual system tests. It aims at being useful for developers, testers and end-users and provides social channels to foster interactions between these groups:

- test designers can think of good test cases, specify them and group them in sensible orders, so-called test sequences
- testers can execute the test cases and judge their results (pass, fail, partial fail, undecided, ...) - and have everything recorded automatically
- test managers can plan and schedule tests
- developers can see how certain failures can be reproduced (issue trackers can link to protocols)
- end-users can see which tests pass/fail in which configurations, helping to decide which device to buy or which version of the software to install

SystemTestPortal is lightweight, written in Golang, and available under the GPLv3 license. Many of its planned features are working already in the currently available version 0.6. By the time of FOSDEM 2018, a mostly complete 1.0 version will be available.

[1] http://www.systemtestportal.org