

Position Paper:

Test-Driven Development for GUI-Applications: Is it possible?

With 20 years of software development experience within BREDEX, we have some clear statements on testing of event driven applications, like any GUI- or component based applications. We would like to share our experiences and discuss our position on these topics with the other symposia participants. We have a strong opinion against programmed tests and would like to explain or discuss one of the following statements:

- Programmers can't write test code.
- Programmers should not write test code.
- Programmatic tests are not feasible for TDD of GUIs.
- GUI tests are (should be) black box tests, while programmatic tests tend to be more white box tests.
- Why should we use programmatic tests at all?

TTD needs techniques to specify automatic tests without the hassle of writing program code. Tests have to be reusable and maintainable.

We would like to explain the evolution of our test strategies within the GUIDancer project. We have started with writing JUNIT tests first. While JUNIT tests are good for testing modules, APIs or interfaces to other systems, this technique does not really help on projects, which use complex graphical user interfaces. We use an agile development process with time boxed development, with various cruise control builds. The existing JUNIT tests did not help very much. Most bugs have been found by manual testing. For regression testing, we could have then programmed a test for checking this bug in future. Most often the effort in writing code for this test was too high.

While GUIDancer has shown in Swing based projects that it supports TTD, we could not use the tool for its own development because there was no support for SWT/RCP up to version 1.2 of GUIDancer.

We are now in the development of version 2.0, which will support SWT/RCP. This allows us to start the transition back to TDD. Transition, because tests will have to catch up with development. There is a big backlog – more than 2 years of development compared to only 2 months of automatic test development. This transition to TTD unveils a lot of conflicts with the current development process.

When an automatic test fails, we have to decide, whether to fix the bug immediately or temporarily remove this test from the nightly build. During the

transition phase, our policy is to fix the bug immediately if it was caused by the latest software change. Because the bug was not there the night before, it must be caused by the last software change. Most of the time it is a bug in the software, but sometimes we have to adapt the test because of changed requirements.

On the other hands the automatic tests unveil bugs, that have been in the code for quite a while, some of which are newly discovered, while others are known bugs. Fixing these bugs will be scheduled in the normal time box planning and the automatic test for this scenario will be removed from the automatic test until the bug has been fixed.

After release of version 2.0, we are confident that we are able to switch to full TDD. Tests will be specified in parallel with the programming and can executed in the nightly builds, as soon as the software is checked in.

BIO

- Hans-Joachim Brede is the founder of BREDEX GmbH and is one of its managing directors. He has implemented and managed multiple, highly complex customer projects and is responsible for the concepts in the development of the automated test tool GUIDancer. Hans holds a Master in Computer Science and was working with the University of Braunschweig and Digital Equipment as a technical consultant.