SOURCE CODE TO REGRESSION CAMPAIGNS

This briefing reports scientific evidence based on experiments about the use of information extracted from source code as input to the selection process of creating regression test campaigns.

FINDINGS

The findings presented in this briefing consider results obtained from three step-by-step experiments. This study was developed incrementally and the final result is a tool to automatize the process of test cases selection based on recent source code changes.

The manual tests selection process is very costly and not always effective. Industry demands automatic tools to perform such processes in a daily basis.

Our hypothesis was that it is possible to identify a relationship between source code and text-based test cases. If this is indeed valid, then we can use the information extracted from source code to select the most related test cases to create a regression test plan.

The information obtained from source code can reflect a more precise view of source code changes then that presented in release-notes.

To use information retrieval techniques on source code and text-based test cases, we need first to extract relevant keywords from source code.

Not all keywords from source code are relevant or relationship to existing test cases. So, a dictionary of stop words needs to be defined such than only the relevant keywords remain.

A metric to validate the results of a technique should be adopted. In scenarios where code are used, code coverage is a representative metric.

Code coverage can express the degree to which some region of the source code of an application is executed after a test suite campaign. With this metric, we can determine how much a changed region is exercised.

The metric can also be used to compare the resulting campaign (automated) with the software architect result (manual) to identify the most appropriate approach (that with the greatest coverage) to the organization.

In a first experiment, we note that keywords together used in a query on the test repository does not exhibit a good result when compared with the test architect campaign.

In the first experiment, test architect selected 120 test case and obtained 51,74% of code coverage, while our selection based on source code change and using all keywords selected 60 test cases and obtained 39,53% of code coverage.

In a second experiment, beyond considering source code, we also used the component associated with the part of code changed.

The combination between components and keywords showed a significant improvement on code coverage. Doing multiple searches on test repository the code coverage increased from 39,53% to 44,18%.

Our third and last experiment was based an extension of the second by avoiding some restrictions used by our industrial partner. That is, we experimented without using its Master Plan.

When we disregard the Master Plan in our proposed process we obtained a better result than test architects selection. The code covarage was 58,18%.

Thus, we have some evidence that the Master Plan is not the best choice as a filter of further selections.

The process to use source code in regression campaigns based on text-based test cases can be summarized in figure below:
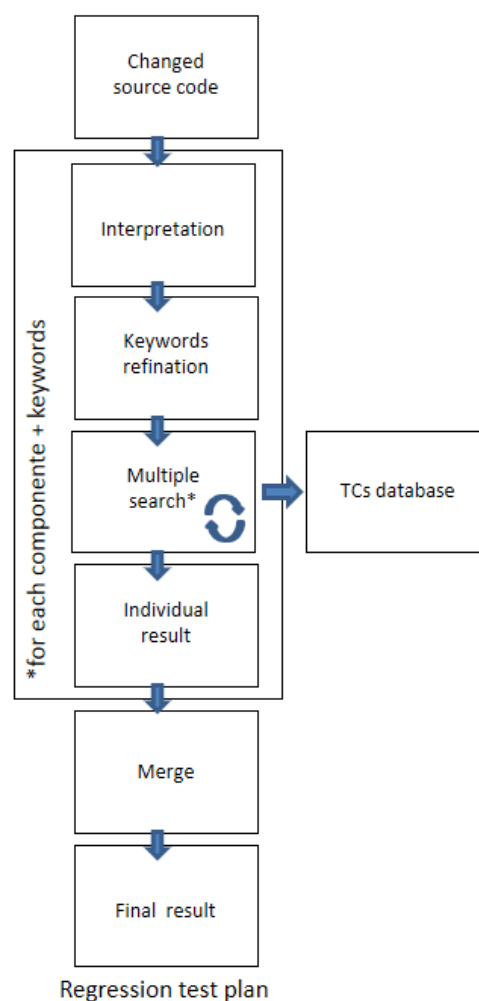
Regression test plan

Who is this briefing for?

Software engineering practitioners who want to make decisions about the use information extracted from source code to generate regression tests campaigns based on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the experiment conducted by Araújo et al.

What is included in this briefing?

The main findings of the original study.

Evidence characteristics through a brief description about the experiment conducted.

What is not included in this briefing?

Additional information about the experiments.

Detailed descriptions about the tools and other internal proprietary information.

To access other evidence briefings on software engineering:

http://www.lia.ufc.br/~cbsoft2017/ii-sast/