Test Gap Analysis
Risk-based Testing of ABAP Applications

Dr. Andreas Göb
Dr. Elmar Juergens
Anatomy of Test-Gap Analysis

Changes -> Test-Gap Analysis -> Untested Changes

Test-Coverage
Test-Gap Analysis

Changes

Test-Coverage

Untested Changes
Changes

Test-Gap Analysis

Untested Changes

Test-Coverage
- Executed during test
Test-Gap Analysis

Untested Changes

Changes

Test-Coverage
Application in Test Process

Changes → Test-Gap Analysis → Untested Changes → Test-Coverage → Test Cases → Test Manager
• = Modified & untested
• = Added & untested
• = Unchanged
Demo
Summary

Changes to code are a good error predictor

Test Gap Analysis reveals untested changes and allows to avoid unwanted shipment of untested changes to production.
As my colleague Fabian explained a few weeks ago, a combination of change detection and execution logging can substantially increase transparency regarding which recent changes of a software system have actually been covered by the testing process. I will not repeat all the details of the Test Gap Analysis approach here, but instead just summarize the core idea: Untested new or changed elements. Therefore it makes sense to use those changed but untested areas.

Several tools are available for this approach, including the Test Gap Analysis tool from the Pytest project.

Figure 1: Test Gaps in Manually Maintained ABAP code only

Take control of your testing process!

Posted on 10/08/2014 by Fabian Streitel

Testing is an integral part of a software product’s life-cycle. Some people prefer executing their tests continuously while they develop, others have a separate testing phase before each release. The goal, however, is always the same: finding bugs. The more, the better.

Unfortunately, the time available for testing and for writing new tests is limited. At some point, you have to get on with development, ship your software and be confident it contains no serious flaws. Often, more tests exist than can be run in the available time span. This is especially true if your tests are executed manually, as is the case for many of our customers. Then the question becomes: which tests do I select to find the most bugs before the release?

Research has shown that errors that are likely (“Did We Test?”) are more likely to have a high impact. This is especially true whether you have a small or a large test suite. 

https://www.cqse.eu/en/blog/take-control-of-your-testing-process/
Contact Information

We are looking forward to discussions!

Dr. Andreas Göb · goeb@cqse.eu · +49 176 10155225
Dr. Elmar Juergens · juergens@cqse.eu · +49 179 6753863

@a_goeb, @ElmarJuergens
www.cqse.eu/en/blog

CQSE GmbH
Lichtenbergstraße 8
85748 Garching bei München