Empirical Software Engineering Group

ESEG

Centro de Informática U·F·P·E

Universidade Federal de Pernambuco

AUTOMATED ACCEPTANCE TESTING

This briefing reports evidence on what is currently known about automated acceptance testing and also reports a case study from industry.

FINDINGS

- The general impression is that Fit is a good tool for specifying requirements.
-
- Fit seems an easy tool to learn. The average introduction is a 3-4 hour crash course supplemented with relevant readings. Computer science students found Fit easy to learn. No evidence about other kind of students, though.
- Teams mixed of business and computer graduates were no better at describing a good quality specification than pure computer graduate teams, even if the mixed teams should have more knowledge of the domain.
- Students that were asked if they would recommend using executable acceptance tests to colleagues, were on average positive to this (5 very likely, 6 somewhat likely out of 17).
- However, there was no correlation between the students' own perception of the value of the tool and the quality of the requirements specification they produced.
- While Fit is intended to make the customer help write the tests, the customers don't always feel the same. Writing acceptance tests isn't always the most prioritized thing to do for customers. However, while finding the time to write Fit tests might hard, there are some claims that doing so leads to a better specification.
- The use of tags in tests was reported as beneficial in order to keep test organized.
- There is no evidence to support that good quality acceptance test specifications made by a customer team resulted in better quality implementation by a developer team.
- Student groups that started a project using a test-first design philosophy ended up with fatter fixtures than the groups that started with Fit in the third iteration.
- The Fit document should be the main definition of the specification and should be kept free of duplication.
- When it comes to deciding to write tests in suites or keeping single tests, a student experiment showed no significant differences between groups focusing on either strategy during the ramp up phase.
- One team ended up with so many Fit documents that they created a special team who dealt only with this. This increased the Fit development, and enabled analysts to effectively write Fit documents.
- The large Fit test coverage give the impression that unanticipated changes that introduced errors would be picked up by the system.
- All tests must locatable by searching for names, functional areas, and metadata.
- According to a study, the average time to code a passing acceptance test was more than four hours.
- It was reported that a team felt using Fit helped them implement features in an economically feasible yet still correct way.
- For describing non-functional scores, Fit got the lowest score in a survey with several notation tools.

Keywords:

Software testing

Acceptance tests

Who is this briefing for?

Software engineers practitioners who want to make decisions about automated testing based on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the systematic review conducted by Haugset and Hanssen.

What is a systematic review?

cin.ufpe.br/eseg/slrs

What is included in this briefing?

The main findings of the original systematic review.

What is not included in this briefing?

Additional information not presented in the original systematic review.

Detailed descriptions about the studies analised in the original systematic review.

To access other evidence briefings on software engineering:

cin.ufpe.br/eseg/briefings

For additional information about ESEG:

cin.ufpe.br/eseg

ORIGINAL SYSTEMATIC REVIEW REFERENCE

Haugset, B.; Hanssen, G.K., "Automated Acceptance Testing: A Literature Review and an Industrial Case Study," in Agile, 2008. AGILE '08. Conference , vol., no., pp.2738, 48 Aug. 2008 doi: 10.1109/Agile.2008.82