

# Multi-modal Functional Test Execution

Shelly Park and Frank Maurer

University of Calgary  
Department of Computer Science  
2500 University Drive NW, Calgary, Alberta, Canada  
{parksh,maurer}@cpsc.ucalgary.ca

**Abstract.** Multi-modal test execution allows execution of the same test against various layers of a software system, e.g. the GUI layer, the web service layer and the business logic layer. Multi-modal test execution helps with identifying the location of software bugs during debugging and maintenance as well as in tracking the progress of the development effort. This paper presents a method that effectively encodes multi-modal functional tests without creating large test maintenance overhead. Our approach extends the Fit table specification structure by multi-modal fixtures and presents the results of test execution in a way to help with debugging and progress reporting.

**Keywords:** Executable Acceptance Testing, Test Driven Development, Functional Specifications, Functional Testing, Validation Testing, Architectural Design Decisions, Multi-modal Test Execution.

## 1 Introduction

In Executable Acceptance Test Driven Development (EATDD) – also called story-test driven development – software requirements are specified in form of executable acceptance tests or executable specifications. The purpose of acceptance testing is to give confidence to the customer that the functional specifications defined in the requirements are correctly implemented in the software product and to ensure high customer satisfaction through critical examination of the quality of the software through continuous regression testing.

In an agile environment, acceptance tests are developed by a customer in collaboration with BAs, QAs and/or developers. The tests can be seen as a contract between the developers and the customer and it defines criteria for a finished product. In an EATDD environment, the acceptance tests play an important role throughout the software development cycle due to the ubiquity of their presence in all aspects of the development cycle. Everyone in the development team and all stakeholders are at least partially involved in either specifying, testing, implementing or communicating through acceptance tests at some point in their work. Acceptance tests are not just for customers, but an important communication tool for all stakeholders – users, customers, analysts, developers and testers – in the development project.

Following the discussions at the Agile Alliance Functional Testing Tools Visioning Workshop as well as Andrea's paper on envisioning the next functional testing tools [1],

this paper argues that the next generation of functional testing tools needs to be multi-modal: tests need to be expressible in multiple formats to satisfy the requirements from different stakeholder groups and need to be executable against different layers of the software system. We call the first *multi-modal test definition* and the second *multi-modal test execution*.

## 2 Multi-modal Functional Test Execution

There are two problems in regards to multi-modal functional testing. The purpose of *multi-modal test execution* (MMTE) is to provide one-to-many mapping between the test definition and the test executions: a single test is executed against different layers and/or components of the software system.

A functional feature can appear in different layers of the software architecture or different components of the software. Rather than duplicating the acceptance test definition per appearance of the functionality or completely ignoring the multi-layer aspect of the software in the acceptance testing, the next generation of acceptance testing tools should acknowledge the need for MMTE and actively practice MMTE in all testing processes.

It is important that non-developers are also aware of the MMTE result based. One of the benefits of having architectural information embedded in the acceptance tests through MMTE is quick feedback about the impact of system-wide architectural or business requirement changes that may occur during the development project. Resource allocation or feature negotiation is easier when all stakeholders are informed with more concrete evidence to base their decisions.

MMTE can also play an important role in deriving requirements. Often customers may not be aware of exactly what they need, thus the requirements may be too vague. Having MMTE capability in the testing tool can help business analysts or testers to work with the customer to derive more detailed requirements in terms of multi-layered system components or platforms.

Based on the requirements and motivations provided above, we have implemented multi-modal test execution based on a popular executable acceptance testing tool called Fit.

## References

1. Agile Alliance Functional Testing Tools Visioning Workshop (October 2007), <http://www.agilealliance.org/show/1938>
2. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1994)