

# The suitability of FitNesse as Acceptance test tool in the Agile environment

Mukund G Kagatkar  
Sabre Travel Technologies Pvt Ltd  
Navigator Building, Level 2  
ITP, Bangalore - 66  
Mukund.kagatkar@sabre-holdings.com

## **Abstract:**

The Objective of this paper is to provide insight to FitNesse for agile development teams. It will first cite the reason for testing tool “FitNesse” which will comprehend the need of acceptance test, help in adopting the TDD. Next it will suggest that FitNesse as a viable testing tool for use in agile environment with the necessity of rapid release schedule. Then it will provide a number of benefits to demonstrate the simplicity and power of FitNesse.

## **Keywords:**

Acceptance Testing, Agile, FIT

## **1 Introduction**

There has been many tools used for acceptance testing in the software world and each of these has their own features and drawbacks. One must diligently choose the tools with their need, business model adopted and financial goals. FitNesse is one of the acceptance test tool which suites to .Net and Java environment, easy to use by anyone involved in the project and most suitable to agile environment. FitNesse has been in use for long time now but its strength and capabilities yet to be understood by the wide software community. We always face the challenge of translating the acceptance test provided by the customer to be understood and verified by testing team. The best way for such challenges will be that a user be able to write the tests in the automated environment. This will satisfy both the customer on the functionality implemented and the testing team on achieving the regression. The attempt is made here to provide a brief about the tool in terms of its usage area and how applicable to agile rather than its technical specification and customization of the tool.

## **2 What FitNesse can provide as Acceptance test tool?**

“Developers write unit tests to determine if their code is doing things right. Customers write acceptance tests to determine if the system is doing the right things.”

The term acceptance testing itself is strongly related to the agile software development method, the customer requirements are gathered in the form of user stories. Acceptance tests, also called customer tests, are tests owned and defined by the customer to verify that a story's implementation is complete and correct.

Acceptance tests represent the customer's interests. The acceptance tests give the customer confidence that the application has the required features and that they behave correctly. In theory when all the acceptance tests pass the project is done.

Advantage of FIT and FitNesse for acceptance testing is that they use an approach that enables users and user representatives to write acceptance tests via a spreadsheet-like approach. This feature decouples the creation of acceptance test suites from dependence on developers. It also enables a mixture of documentation and testing that can result in testing documents that also provide considerable insight into how systems should work. Such test suites can really help when the time comes to produce formal documentation and training materials.

### **3 How FitNesse fits into Agile?**

User stories are basically "3Cs – written on a card, serve as promise of conversation and serve a confirmation" for developers and customers to work on a problem together. The outputs of that conversation are detailed understanding of the story, estimates of the amount of effort each task will take, intermediate candidate solutions, and ultimately acceptance tests.

General Acceptance testing proceeds is:

1. The customer writes stories.
2. The development team and the customer have conversations about the story to flesh out the details and make sure there is mutual understanding.
3. If it is not clear how an acceptance test could be written because there is not enough to test it against yet, the developer does some exploration to understand the story better.
4. When the exploration is done, the developer writes a "first cut" at one or more acceptance tests for the story and validates it with the customer.
5. Once the customer and the developer have agreed on the "first cut" acceptance test(s), he hands them over to business people (QA people on our project) to write more tests to explore all boundary conditions, etc.

The above is most general situation prevails in the agile environment. The involvement of customer at each stage is good part of application development but how much real time he/she can be able to spend with Dev team after the stories are handed over? Will it be easy for him to write his acceptance test rather than reviewing of the tests written by development team? To answer these questions FitNesse comes as handy where user can write the acceptance tests in excel based template and save time on effective communication on business related queries rather than test itself.

Some of the reasons that agile web development teams would want to adopt a rapid release schedule

- Super-short releases allow the marketing department to obtain rapid customer feedback and to respond to that feedback quickly. They can test modifications to existing products and release new products sooner and more often. The agile concept of rapid feedback works well for the marketing team, too!
- Frequently updated web sites perform better in organic search engine results. This enables the business to attract customers to its web site without paying for sponsored links.
- Businesses that support diverse customer niches demand more agility from their products.

Framework for Integrated Test (FIT) is an open source framework for user acceptance testing, and a tool for enhancing the communication and collaboration between analysts and developers. FIT lets analysts write acceptance tests using simple HTML tables or in spreadsheet. Developers write fixtures to link the test cases with the actual system itself. FIT compares these test cases; written using HTML tables or in spreadsheet, with actual values, returned by the system using fixtures, and highlights the results with colors and annotations. These tests can be stored and run as regression suite to achieve the rapid release schedule keeping all the acceptance and functional tests in one tool.

#### **4 All about FitNesse**

The translation of idea to concept in software development is an area plagued with tension, miscommunication and failure. Every software development process has its own take on how to solve the issue and streamline the translation of requirements into tested code. In the real world, applications keep growing in size and complexity, and change frequently; thus, the necessity for continuous testing constantly increases. But although the idea seems simple, in practicality it requires discipline and application.

FitNesse is a tool/approach to this age old problem that looks at it from a new and surprisingly effective point of view. By treating easy-to-create tests as the actual requirements, FitNesse allows non-technical Business Analysts to specify their needs in the form of executable requirements. These requirements then become a living part of the software ecosystem instead of a set of stale documents that grow further out of sync with the code base as time goes by.

FitNesse is a tool created by Robert Martin, Micah Martin, and Michael Feathers. It is a wiki-based test execution engine that allows users to write, edit, and run FIT tests. Together FIT & FitNesse enable developers and users to express acceptance tests in a tabular format and execute them from a simple web interface.

Agile programming prescribes automated acceptance testing so that tests can be run often, while facilitating regression testing at a low cost.

Just two steps are required to automate user acceptance tests using FIT:

1. Express a test case in the form of a FIT table
2. Write the code in Java, called a fixture, that bridges the test case and system under test

Tables are the crux of FitNesse. Acceptance tests in FitNesse are defined in the tabular format. This deterministic tabular format is understood by the FIT library. These tests or tables are defined in terms of input and expected output data. In order to read those tables and test your application under test with the data used in these tables, you need to write Fixture Code. You can imagine that Fixture Code is a bridge between the table and application under test. It understands the language of table and uses this information to exercise the functionality of application.

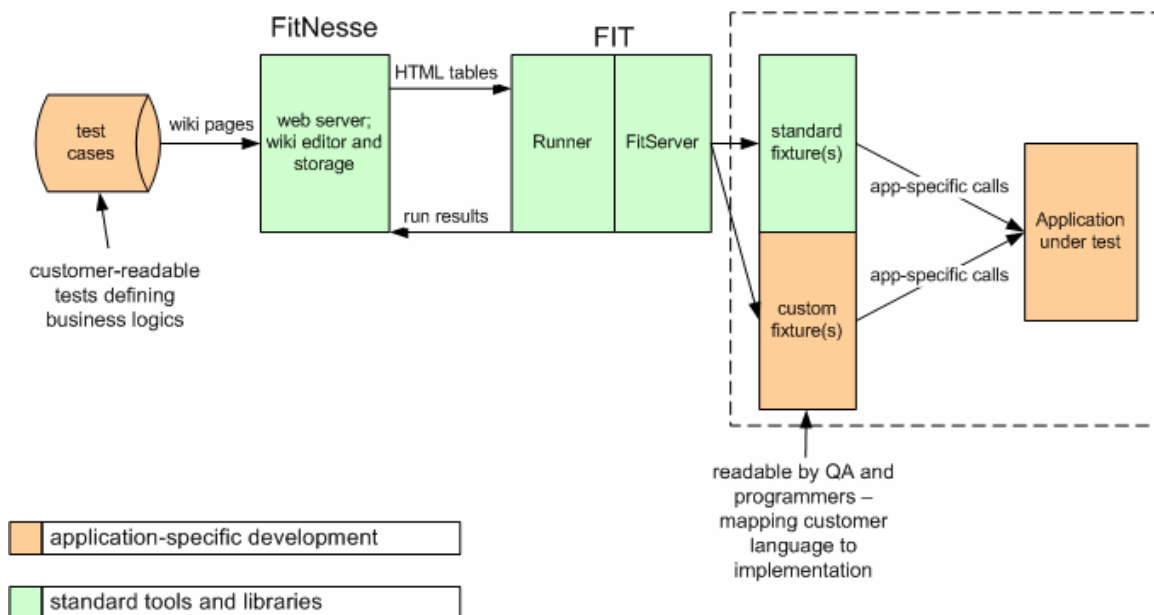


Figure1: Architecture (Source: Sabre Holdings)

## References

- [1] Miller, R. W. and Collins, C. T., Acceptance testing. XP Universe, 2001, URL <http://www.xpuniverse.com/2001/pdfs/Testing05.pdf>.
- [2] Beck, K. and Andres, C., Extreme Programming Explained: Embrace Change (2nd Edition). Addison-Wesley Professional, 2004.
- [3] Fit documentation URL <http://fit.c2.com/wiki.cgi?IntroductionToFit>.
- [4] FitNesse user's guide URL <http://fitnesse.org/FitNesse.UserGuide>
- [5] Acceptance testing URL <http://www.cs.helsinki.fi/u/jaarnial/jaarnial-testing.pdf>
- [6] <http://en.wikipedia.org/wiki/FitNesse>

**About the Author:** Mukund G K is a Senior Software Quality Engineer working with Sabre Travel Technologies Pvt Ltd, Bangalore. He has around 6 Years of experience in the software quality field which includes both the software testing and the quality assurance. He has worked in multiple domains such as Travel, Aerospace and bioinformatics.