Anyone Using Executable Requirements?

Asked 16 years, 4 months ago Modified 10 years, 8 months ago

Viewed 2k times

13

In my limited experience with them executable requirements (i.e. specifying all requirements as broken automated tests) have proven to be amazingly successful. I've worked on one project in which we placed a heavy emphasis on creating high-level automated tests which exercised all the functionality of a given use case/user story. It was really amazing to me how much easier development became after we began this practice. Implementing features became so much easier after writing a test and we were able to make major architectural changes to the system with all the confidence in the world that everything still worked the same as it did yesterday.

The biggest problem we ran into was that the tools for managing these types of tests aren't very good. We used Fitnesse quite a bit and as a result I now hate the Fit framework.

I'd like to know 1) if anyone else has experience developing using this type of test-driven requirement definition and 2) what tools you all used to facilitate this.

testing process automation

Share

Improve this question

Follow

edited Apr 10, 2014 at 19:33

tshepang
12.4k ● 25 ● 95 ● 139

asked Aug 15, 2008 at 22:02

Mike Deck
18.4k ● 16 ● 70 ● 94

6 Answers

Sorted by: Highest score (default)

6

The primary tool I've also used was FitNesse. I've used it at several companies, with very good results. We did have test cases numbering in the many thousands, and we had to be very disciplined in how we organized and used them.

I've tried some other tools, including writing my own DSL (domain-specific language) and using things like RSpec. I really like RSpec, but it is certainly more of a developer tool than a business one.

I know Rick Mugridge has been working on a tool called ZiBreve (http://www.zibreve.com/visit.php?page=index) which is supposed to have stronger refactoring support. I haven't used it myself, but I know Rick and have talked to him several times. I know there was discussion at Agile

2008 on some different ways to deal with the Fitnesse tests in general.

Other than that, I haven't seen a lot of good tools out there. Even tools like WinRunner are fine for QA type tests, but for exploratory testing of requirements by the business, FitNesse or a custom DSL seem to be the ways to go right now.

Share Improve this answer Follow

answered Sep 1, 2008 at 19:31

Cory Foy
7,212 ● 4 ● 32 ● 34

2

You might want to take look at Robot Framework (http://robotframework.org). It's FIT-like but hopefully easier to integrate to different testing tools, version control and continuous integration. Different abstraction levels in the test data also make it easier to maintain the data, and when the separate test data editor gets more mature maintenance gets even easier. The quick start guide introduces the most important features of the framework and acts also as an executable demo.

Share Improve this answer Follow

answered Oct 7, 2008 at 10:23

Pekka Laukkanen

I've had to use, test and set up both fitnesse and one of it's competitor, GreenPepper for my work, and what I can say is :

2

GreenPepper is a confluence plugin (confluence is an enterprise wiki from atlassian) and have many of the things you need in an "enterprise" level tool with little to no additional work required :

- Better user friendly -rich text- wiki syntax (makes it easier to work with for non technical people)
- It integrates very well with many development tools : Eclipse, VB, maven2 and Nant plugin, I tested most and was very pleased.
- User and access rights are managed by confluence, which is to say it's good and make uses of database of your likin (which might be mandatory depending on where you work)
- Many other functionalities that might or might not be required : ssl support, remote execution (install the wiki on unix, execute on windows if you are working on a C# project, or reverse)
- Looks way better :D

Big downs for GreenPepper are : Configuration is quite **hard** and documentation is **poor** (although they seem to be working on it and they answer quite fast on their forum) and also it is **not free**, you have to pay for both confluence and GreenPepper, which might add up to quite a lot.

Fitnesse is very basic in my opinion, very easy to set up, it works but that's it, you can use some of the fitnesse plugins developed by the open source community, and even some Fit plugins, such as the Eclipse plugin (build the skeletton of the fixture from a fitnesse test file, provided it's in a .fit extension, very usefull). Integration is not ideal, authentification and access rights management is poor, but it's **FREE** and if you need something, you can do it because it's open source.

Share Improve this answer Follow

answered Jul 8, 2009 at 13:06

SAKIROGLU Koray
208 ● 1 ● 9

1

I've found that using contracts is a great approach. Metaprogramming contracts are generally lower-level than the types of integration tests you describe, but the two are certainly not mutually exclusive. I find contracts help keep documentation, implementation, and testing all in sync -- this is a major problem of TDD (not that it isn't a problem in non-TDD).

Share Improve this answer Follow

answered Aug 23, 2008 at 18:05

James A. Rosen
65.2k ● 62 ● 184 ● 263

1

I've tried Fitnesse and its really awful (particularly integration with SVN). And our company develop similar open-source tool with fit engine: FitPro

Another brilliant tool I've used is Concordion. It has the only disadvantage - requrements in html format

Share Improve this answer

Follow

answered Sep 16, 2008 at 15:01

kzotin

5,365 ● 3 ● 31 ● 36

1

My experience is limited to personal projects and found much the same advantages you mentioned. I recommend http://metacpan.org/pod/Test::Simple::Tutorial which was my inspiration for trying out testing-based development. The perl testing modules seem pretty useful and flexible, though I have nothing to compare them to.

I also believe tests are vital for the maintenance period of a project. If you have good tests to begin with, it saves a lot of time and mistakes later on. I wish I had put more work into tests on my current project.

Share Improve this answer

Follow

edited Apr 5, 2014 at 15:10

szabgab

6,262 ● 11 ● 54 ● 64

answered Aug 16, 2008 at 7:48

Jon Ericson

21.5k ● 12 ● 102 ● 151