

FIT

Framework Integrated Testing for Writing Acceptance Tests

Agile Software Community of India
(ASCI)

Mumbai Conference, 27th August 2005

Conan Dalton
conan.dalton@bnpparibas.com

Dhaval Dalal
dalal_dhaval@yahoo.com

What are we talking about?

- Automated acceptance testing
 - testing approach that allows business users completely specify an application using automated test scripts.
- a.k.a.
 - Customer Tests
 - Functional Tests
 - System Tests

Initial Example Demo

EMI Calculator

From Concept to Completion

Dhaval Dalal

Specify → Create → Run

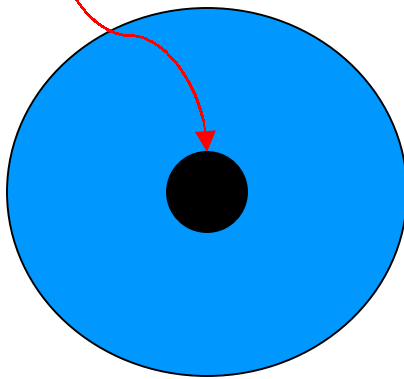
- Customers create Tables in a document.
 - Tables describe specific examples of how the software should work.
 - Programmers format tables in a *FIT-friendly* format.
- Programmers create Fixtures.
 - Fit uses "fixtures," to connect tables to software.
 - Fixtures interpret Tables.
 - Programmers write the fixture code.
- FIT runs the document against the system

fit.Summary

- FIT allows tests to be written in the domain language
 - Uses reflection to bind test script elements to "fixtures"
- FIT fosters Collaboration
 - Business Analysts along with customers specify the tests.
 - Programmers and Testers increase test coverage from customers examples.
- FIT tests are automated
 - Supports Immediate Feedback
- FIT works with HTML documents.
 - Reads/Writes results back as HTML
 - Test verdicts annotated with messages and colour
 - Summarizes test results (right, wrong, in error)

Specification → Test

Here is the
requirement



here is a requirement:

*"User may borrow up to
2 books at a time"*

Specification → Test

- step-by-step in the UI

select "Harry Potter" in "available" list

click on "borrow" button

select "Refactoring" in "available" list

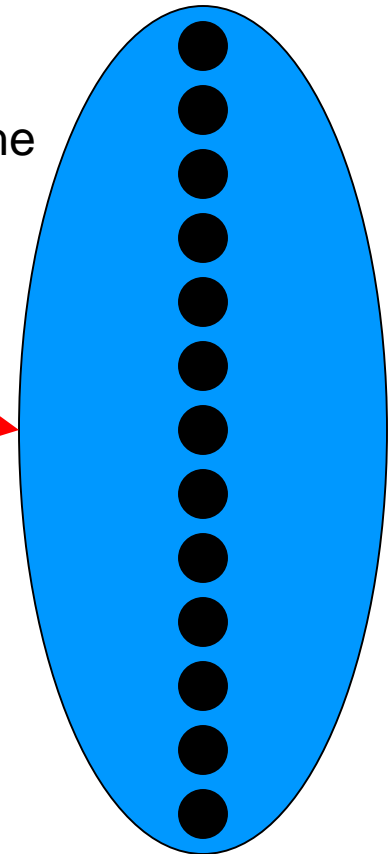
click on "borrow" button

select "XP Explained" in "available" list

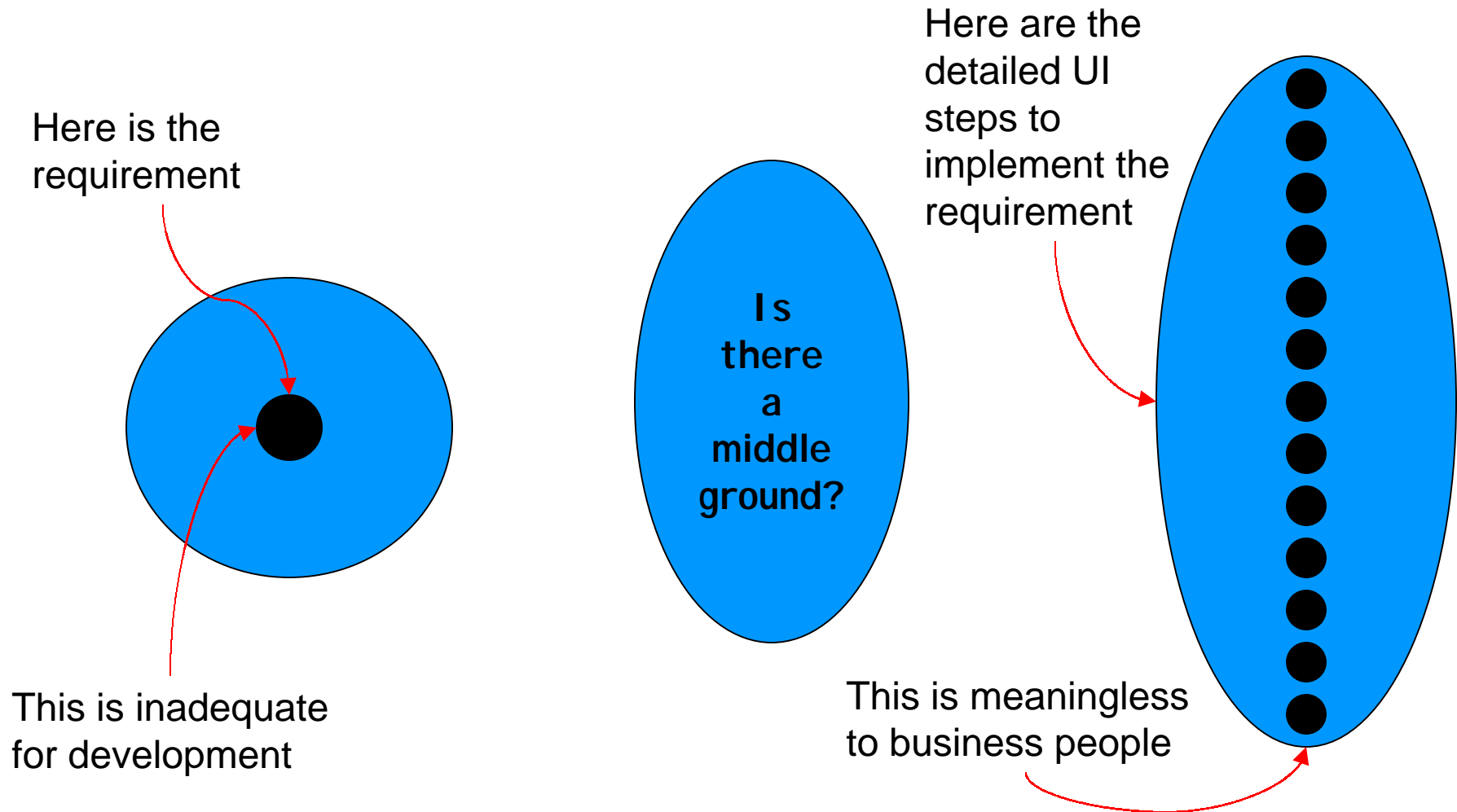
check that "borrow" button is unavailable

...

Here are the
detailed UI
steps to
implement the
requirement



Specification → Test



The Middle Ground

- Use a domain-specific language

borrow "Harry Potter"

borrow "Refactoring"

cannot borrow "XP Explained"

return "Refactoring"

borrow "XP Explained"

cannot borrow "Refactoring"

The hard part

Developing the fixtures to support automated tests at system boundaries in a language that's meaningful to the user

Web Test Example Demo

Conan Dalton

Acceptance Test Better Practices

- Automate tests
- Test continuously
- Develop tests collaboratively
- Use domain language
- Evolve tests
- Define system boundaries

FIT Ports and Alternatives

- Ports
 - Available: Java, .NET, Perl, Python
 - In Progress: C++, Ruby
- Open-source alternatives
 - Selenium, Exactor, TextTest
- Other Commercial Tools
 - WinRunner, Silk
- Any other home-made testing framework
 - it's not difficult to write!

References

- Ward Cunningham's acceptance testing framework
 - Official FIT site (<http://fit.c2.com>)
- FIT Library
 - Rick Mugridge
(<http://www.cs.auckland.ac.nz/~rick/FitLibrary>)
- Homebrew Test Automation
 - Bret Pettichord (<http://www.pettichord.com>)

Thank-You

Conan Dalton
conan.dalton@bnpparibas.com

Dhaval Dalal
dalal_dhaval@yahoo.com