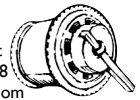


Testing in an agile environment

James Lyndsay, Workroom Productions

EuroSTAR mini event / TestNet April 2008

Testing in an Agile Environment
© Workroom Productions 2008
www.workroom-productions.com



Agility (agility)

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

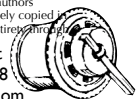
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

EuroSTAR mini event / TestNet April 2008

© 2001, the above authors
this declaration may be freely copied in
any form, but only in its entirety, through
this notice.
Testing in an Agile Environment
© Workroom Productions 2008
www.workroom-productions.com



Agility (agility)

Testing is Automatic

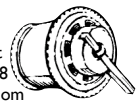
Errors should be Avoided

Friction is reduced by
Improved Communication

Grouping the 12 practices from Beck's
Extreme Programming Explained

EuroSTAR mini event / TestNet April 2008

Testing in an Agile Environment
© Workroom Productions 2008
www.workroom-productions.com

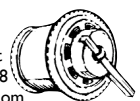


Shared Roles

Implicit in XP – but not included in practices

EuroSTAR mini event / TestNet April 2008

Testing in an Agile Environment
© Workroom Productions 2008
www.workroom-productions.com

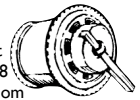


Testing is Automatic

Comprehensive testing, but not by testers

Refactoring

Continuous integration



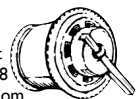
Errors should be Avoided

Metaphor

Pair work

40-hour week

Coding standard



Friction is reduced by Improved Communication

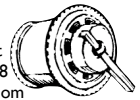
Planning game

Small releases

Simple design

Collective ownership

On-site customer



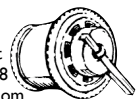
My experiences

Tester – reported bugs

Participant – part of the agile team

Mentor – to a test lead on an agile project

Consultant – after implementation problems



Bugs missed, lessons learned

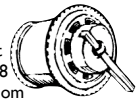
You can't get the bugs out of Microsoft Word
even automated unit test have their limits

The side-effects of unexpected truncation

Test driven design? No test = no code

Discounts that don't

It's easy to get used to a bad behaviour



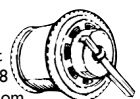
What's it like?

You'll talk to everyone, and get involved in their testing

You'll occasionally fix a bug

You'll lose your independence

You learn to live without (some) documentation



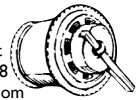
Common problems

Decision fatigue

Testing in later iterations

Keeping testing and coding in sync

Test management

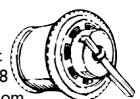


Common problems

Passing tests \neq working code

Agile \neq Exploratory

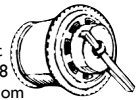
Testers can make a terrible mess



Testers help

Bring your own stories and skills

Expose and diagnose the unexpected

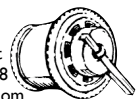


Exploration and diagnosis

Necessary

Similar spirit

Not a perfect fit



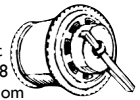
Supporting the learning process

Give fast feedback

Enhance the work of the team

Be involved in retrospectives

Re-factor required processes



Conclusion

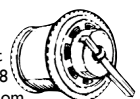
Challenge your beliefs:

the virtues of independent testing

the importance of documentation

the necessity of long-term planning

Bring your skills, and work within the team



jdl@workroom-productions.com

AIM/Skype: workroomprds