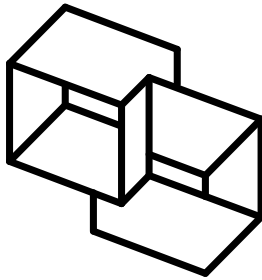


## Quality Assurance on Agile Processes



Pete McBreen, McBreen.Consulting  
pete@mcbreen.ab.ca

### Understanding the Quality Assurance implications of Agile Processes

**Why is there a shift towards the so-called *Agile Methodologies*?**

**Do we need Agile Quality Assurance?**

**Agile Testing - another buzzword or a step in the right direction?**

**Do Agile Methods require changes in the way we do Quality Assurance?**

**The Changing Face of Quality Assurance**



**Traditional software development approaches are not always successful**

**An extremist view of traditional software development is *Crummy Software Late***

**Duff O'Melia of RoleModel Software asks some hard questions**

How do you know your code works?

Have you ever been afraid to change code?

How do you know your design is right?

What happens if your star programmer is hit by a bus?

**Something is rotten in the state of software development**

**The *Dilbert Phenomenon* is a very strong hint that there are serious problems**

*Pointy Haired Boss* jokes are not funny any more

**Organizations are completely dependent on software, but coding is a lousy job**

The Suits make the decisions, but the Techs get blamed for failing to deliver the code

**Even the promise of dot com riches turned out to be a *death march* to nowhere**

## **Developers are sick and tired of processes that make developing great software harder**

### **In the last five years developers have woken up and said *It can be different!***

The era of big process software engineering is over, the SEI CMM failed to deliver

*Good Enough Software* is just a fancy face on the good old *blue screen of death*

Small wonder then that many developers now contribute to Open Source projects

### **Many developers have decided there has to be a better, more agile way to do software**

## **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.**

## **What is so different about these new Agile approaches?**

**All put people and the interaction between people as the main focus of attention**

**All embrace the idea that dealing with partial knowledge is the key to success**

**Their response to schedule pressure is to prioritize and focus on early delivery**

**They require extensive involvement by the project sponsors and users**

## **So do we need Agile Quality Assurance?**

**The Agile approaches are changing the conversation about software development**

**Agile shifted our attention to small teams incrementally delivering quality software**

The old ideas about testing at the end of the coding phase no longer applicable

**We need to think about the role of Quality Assurance in Agile Projects**

**Definitely NOT business-as-usual**



## **Brian Marick has identified two imperatives that affect testing in Agile Projects**

### **Extending the programmers Hands-On Imperative to users**

Abstractions simply *mean differently* than working software, so the requirements never represent the application adequately

### **The imperative toward human contact, face-to-face conversation**

Agile methods foster collaboration to obtain communication that documents cannot replace

## **Agile Testing - another buzzword or a step in the right direction?**

### **Agile Testing abandons the old notion about how Testers communicate**

Requirements and design docs are insufficient, as are test plans and bug reports

### **Agile Testing sees docs as interesting texts, partly fictional, often useful**

Documents are as good as they are going to get

### **Testers need to join in the conversations with developers and users**

## **Agile Testing is still evolving and being actively discussed on mailing lists**

### **Marick suggests we need new models for Test Development**

Testing may be degraded by poor or late docs, but it should not be blocked entirely

Testers should use sources of information other than project docs when designing tests

Test design must take into account what is learned from running tests

The tester must take explicit, accountable action in response to dropped handoffs, new handoffs and changes to the contents of handoffs

## **A Testing Team Motto**

**We are a service organization whose job it is to reduce damaging uncertainty about the perceived state of the product**

<http://www.testing.com/writings/purpose-of-testing.htm>

### **Metaphors shape how we think**

Testing as Flashlight ...

## **So do Agile Methods require changes in the way we do Quality Assurance?**

**More questions than answers right now**

**At one level, Quality Assurance is still looking at the same deliverables**

But the process used to create the deliverables does affect how Quality Assurance works

**Many things about Quality Assurance will change drastically with Agile Methods**

Unfortunately my crystal ball is refusing to tell me exactly what will change...

***But there are some strong hints...***

## **The Changing Face of Quality Assurance**

**Programmers have become test infected thanks to Beck, Gamma and JUnit**

Test Driven Development is now supported by most development environments

**Incremental development changes things**

Many small production releases are fundamentally different than a single planned release

**Fit.c2.com may be the next big thing**

***It could be even bigger than JUnit 😊***

**FIT Allows Users to define their own tests  
using simple HTML documents**

**FIT could lead to *specification by example***

Especially for standard business applications

**FIT derives from a paper Ward Cunningham  
wrote for XP/Agile Universe Conference**

*Acceptance Testing as Document Authoring and  
Annotation*

**Ward was heavily influenced by the work  
on *Context Driven Testing***

**The choice of process depends on context: what  
learning will have value right now**

**fit.c2.com - A Small Demonstration**



## **Quality Assurance on Agile Projects is Different ...**

### **But we do not know how different ...**

Learning the implications of the Agile Methods is going to be an interesting challenge

### **In 1998 eXtreme Programming was just starting to be talked about**

### **The Quality Assurance community is not renowned for process innovation ...**

But *Individuals and interactions, Working software, Customer collaboration and Responding to change* have a way of changing perspectives ☺