T-76.650 Software Engineering Seminar, 3cr

SQA in Agile Software Development

Introduction to SQA in Agile Software Development

28.1.2004

Juha Itkonen
SoberIT
Contents

- The role of testing in different process models
- Testing and QA in agile methodologies
  - eXtreme and Exploratory viewpoints
- Balancing between agile and plan-driven methods
- Cycles of Control – a framework for managing iterative and incremental software development
Prosessimallit vesiputouksesta XP:hen

Functionality

Time

Requirements

Design

Implementation

Testing

Waterfall

Incremental, e.g. RUP

Agile - XP
Organizing testing (in theory)

Waterfall

Programmers

Testers

Agile (XP)

Tester

Customer

Programmer

Core idea:
Collaboration
Agile process

- Iterative and incremental in nature
- Package existing software engineering practices
  - Nothing new, except the underlying philosophy and perhaps the combination of practices
- Embrace change rather than control it
  - Suitable for extremely turbulent environments
- Focus on delivering business value
  - Customer/User involvement paramount
- Can’t be used in all projects!
Agile Software Development Manifesto

- Signed by authors of ASD, XP, Scrum, Crystal, FDD, DSDM, and “pragmatic programming” + some others
- Agree at the first level
  - Need to respond to change
- Agree at the second level
  - Four value statements
- Agree at the third level
  - 12 more detailed statements
- Don’t agree at the fourth level
  - Detailed project tactics
Agile Manifesto

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

How do these values affect quality assurance?

http://www.agilemanifesto.org/
Principles of the Agile Alliance

- Satisfy the customer through early and continuous delivery of valuable software
- Agile processes harness change for the customer’s competitive advantage
- **Deliver working software frequently**
- Working software is the primary measure of progress
- Agile processes promote sustainable development at a constant pace
- Business people and developers must work together daily
- Build projects around motivated individuals, give them support and trust them to get the job done
- The most efficient and effective way of conveying information is face-to-face conversation
- Attention to technical excellence and good design enhances agility
- **Simplicity** -- the art of maximizing the amount of work NOT done -- is essential
- The best architectures, requirements, and designs emerge from self-organizing teams
- At regular intervals the team tunes and adjusts its behavior to become more effective
Challenges for testing in agile software development

- What information is the testing based on?
  - What to test and what are the expected results?
  - How to make testing, development and business collaborate?
  - How to involve customer and business people in testing?

- How do we know where we are?
  - Working software is the primary measure of progress
  - Testing should provide the data
  - If it’s not tested it doesn’t exist

- How to keep up with the pace of the development?
  - How to produce and communicate relevant information promptly?
  - How to test early but not do anticipatory test design?
Agile methodologies and testing

- eXtreme programming
- Scrum
- DSDM
- FDD
- LD
- ASD
- RAD
- MS Synch-and-Stabilize
- Crystal family

- Some define strict disciplined testing practices

- Continuous testing and automated regression approach is common
  - XP, MS, Crystal, ...

- Some do not say much about testing approach
  - E.g. FDD: “... processes used for testing are not the main process issues with which the organisations are struggling ... and most organizations already have reasonable testing processes in place”

“Agile Testing” is a buzzword that is not yet defined
Two views of agile testing

**eXtreme Testing**
- Automated unit testing
  - Developers write tests
  - Test first development
  - Daily builds with unit tests always 100% pass
- Functional testing
  - Customer-owned
  - Comprehensive
  - Repeatable
  - Automatic
  - Timely
  - Public

Focus on automated verification — **enabling agile software development**

**Exploratory Testing**
- No prescribed test cases
- Manual testing by professional skilled testers
- Optimized to find bugs
- Minimizing time spent on documentation
- Continually adjusting plans, re-focusing on the most promising risk areas
- Following hunches
- Freedom, flexibility and fun for testers

Focus on manual validation — **making testing activities agile**
Definitions of Exploratory testing

- To the extent that the next test we do is influenced by the result of the last test we did, we are doing exploratory testing. *James Bach, 2001*

- Exploratory testing involves simultaneously learning, planning, running tests, troubleshooting and reporting results. *Cem Kaner, 2001*

- In scripted testing, tests are first designed and recorded. Then they may be executed at some later time or by a different tester.

- In exploratory testing, tests are designed and executed at the same time, and they often are not recorded.

- You build a mental model of the product while you test it. This model includes what the product is and how it behaves, and how it’s supposed to behave.

  *James Bach, Rapid Software Testing, 2002*
Scripted vs. Exploratory Tests
Quality Assurance

(1) A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
(2) A set of activities designed to evaluate the process by which products are developed or manufactured.

- Application of sound technical methods and tools
- Formal technical reviews and inspections
- Software testing
- Enforcement of standards
- Documentation
- Change control
- Extensive measurement
- Record keeping and reporting of the process

QA vs. Testing
- Different attitude
- QA is focused to build in quality and prevent defects to ever happen
  - constructive
- Testing is focused to find defects and show the level of quality
  - destructive
Agile methods and quality assurance

- Reviews
  - Pair programming
  - Bad smells in code
- Short increments
  - Increment demos
  - Fast feedback
- Customer collaboration
- Coding conventions
- Static analysis

- Even testing has a different, perhaps more QA oriented role in agile methods
  - Testing provides information with continuous pace
  - Testing is not seen purely destructive action of breaking things down
Value of testing in agile development

- Testing in agile software development should provide the information that stakeholders need to make decisions and steer the development into the right direction.
- This information must be provided promptly.
- Testing provides data on the status of the deliverables and generates *project intelligence*.
- Project intelligence is knowledge of *risks and benefits*.
- Knowledge of risks, benefits, test records and results are more valuable than test documentation and infrastructure.
- We can increase the value of testing most by
  - Improved intelligence
  - Providing intelligence earlier.
References


Lisää tietoa aiheesta:

- [www.extremeprogramming.org/](http://www.extremeprogramming.org/)
- [www.satisfice.com/articles/et-article.pdf](http://www.satisfice.com/articles/et-article.pdf)
- [www.agilemanifesto.org](http://www.agilemanifesto.org)
- [www.xprogramming.com](http://www.xprogramming.com)
- [www.testingeducation.org](http://www.testingeducation.org)
- [www.agilealliance.com/articles](http://www.agilealliance.com/articles)
- [www.context-driven-testing.com](http://www.context-driven-testing.com)