When Agile Meets OO Testing
A Case Study

Siegfried Goeschl, Matthias Herp, Clemens Wais
The XYZ Project

- Started in the beginning of 2008
- Medium-sized client/server Java application
- First Scrum project within department
- Rather large Scrum team
- 600 inhouse user
Scrum and Concerns

• Can we cope with iterations of 5 weeks?!
• Refactoring will break existing code!
• Co-location of developers and tester will spoil the QA work!
• Will most testing done by developers?
“Can we deliver high quality software despite using Scrum?!"
“Yes, we can.”
What Does This Mean?

- 228,000 NCSS of Java production code
- One delivered defect per 1,000 NCSS
- Matches SEI CMM Level 4 quality
How to get there?

- Extensive pair programming
- Teaming of Dev & QA
- Automated unit and component tests
- Using static source code analysis
- Continuous integration and deployment
Pair Programming

- Pair Programming is expensive
- Nearly everything is done in pairs
- Usually four feature teams
  - One developer is the “feature owner”
  - The buddy acts as sanity check
- Pair programming is mentally exhausting
Teaming of Dev & QA

- All located in a single office
- Testers are part of the planning session
  - Participate implementation problems
- Informal review before declaring a feature “ready for testing”
  - No defect tracking at that stage
Unit & Component Test

- Using JUnit 4
- Extensive usage of mocking frameworks
- Represents 26% of the coding effort
- 3,941 tests and 43% statement coverage
Static Source Code Analysis

• Using built-in Eclipse and PMD analyzer

• High number of warnings and todo’s due to
  • Incomplete “Definition of Done”
  • Absence of active tracking
Continuous Integration

- Using “Hudson”
- Test coverage is always measured
- During QA each successful build is automatically deployed for testing
- Currently no gathering of historical data
“Did we have any problems?”
“Yes, we did.”
Technical Debts

- The 400 “todo” in the code are hard to fix
  - estimated 800 - 1.200 hours
- No fully controlled test environment
  - The integration test suite is brittle
- UI test automation is on hold
  - Manual regression tests takes 50 hours
- No Design by Contract or assertions
Conclusion

• XYZ is not perfect but still very good
• In production after the seventh iteration
• Delivered excellent software quality
• The project ends in Sept. 2010
• Missed the initial schedule by two iterations
• Blue-print for current and future projects
“Interested in facts and numbers?!”
Defects Found per Phase

- Production
- Acceptance
- QA

Tuesday, November 2, 2010
Test Coverage
What the reviewer said
“The testing approach has nothing to do with OO testing”

- Hardly any OO testing methods applied
- Testing is focused on user stories and not the programming language paradigm
“The test coverage makes the result questionable”

- Test coverage is only measured from automated and working tests
- Manual UI test are not measured
- Integration tests are not measured
“The test methods hardly conforms to agile testing apart from informal review together with tester”

- Testing is done alongside with development within an iteration
- Regression tests are mostly automated
- Continuous Integration and Deployment
- Feedback loop formally established by iteration retrospective