Title, abstract and biography

Dietmar Strasser
Expedition Toward 21st Century Test Automation for Agile Testing
Najaarsevent TestNet: 22 september 2009

Abstract

As more and more companies move to Agile software delivery approaches, new challenges and new dynamics impact quality assurance practices. Requirements are raised, changed and reprioritised in much shorter timescales, whilst functionality, performance and scalability delivered must be fully tested to ensure that it meets the needs of end users and the business. Manual testing processes are simply not sufficient to deliver the fast and repeatable results that underpin the Agile proposition. Yet, many automated testing tools are unsuited to the Agile approach. How does an organisation identify the tools that are usable and flexible; allowing non-technical and non-testing specific members to contribute; enabling open-source integration; and enabling test-driven development?
In this presentation, Dietmar explores the path toward 21st century test automation. He will share how his QA organisation tackled some of the above challenges during Borland's own Agile transformation and how this journey led to an enhanced delivery process, ensuring quality, reducing risk and lowering costs. He will also examine:
• The elevated importance of testing in an Agile delivery environment
• How and why process and tools play a role in supporting the people
• The differences between traditional and Agile test automation
• The need for speed: accelerating with fast, automated test scripts
• Operating Agile and traditional testing methods side-by-side
• How to deal with test automation in a distributed development environment.

Biography

Dietmar Strasser is the director of quality assurance in Borland Software's largest development facility in Linz, Austria. He has been on the Silk development team for more than 11 years in various engineering roles and is currently providing quality management services to the agile development teams. Previously, he was responsible for the overall quality across the entire suite of Borland's quality testing & management tools. Dietmar's experience in traditional, agile and requirements-based-testing approaches has resulted in Borland's delivery of high-quality, high-value products and services for customers worldwide.

Toward 21st Century Automation for Agile Testing

Dietmar Strasser
Director QA
dietmar.strasser@microfocus.com

Overview

- Agile Transformation
- Why Test Automation?
- Differences between Traditional & Agile Test Automation
- Requirements for Test tools
- Test Automation in a global enterprise company
- Q & A

Agile Transformation

- Product Owner
- Requirements Management System
- Integration of QA and Doc
- Introduction of Role „QM Coach“
- Facility adaptations & team co-location

“Agile is a journey, not a destination“

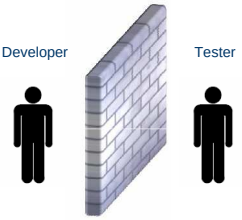
Why Test Automation?

- Cost reduction
- Easy effort estimation for test execution
- Find bugs earlier
- Shorter test execution cycles
- Replacement for repetitive, manual testing
- More, in less time and better quality
- Reusing tests for different configurations
- Assign testing resources to creative testing tasks
- Agile: Keep pace with short iterations!

Differences between Traditional & Agile Test Automation

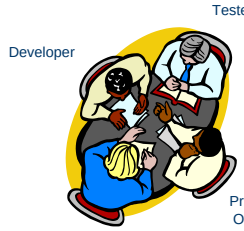
- Collaboration
- Time to start
- Testcase development
- Testcase execution

Traditional: Collaboration



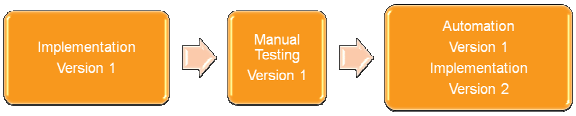
- Developers threw software over the wall
- Tester = „quality police“
- No attention to testability
- Communication over defect tracking system or email
- Conflicts due to communication issues
- Test coverage

Agile: Collaboration



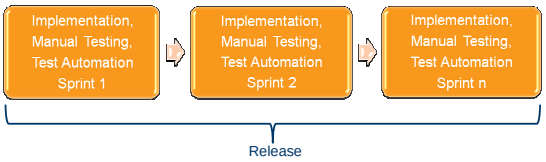
- Cross-functional team – Testers are integrated into team
- Testers involved from the beginning
- Everyone is responsible for quality
- Everyone tests and write tests
- „Face-to-face" communication
- Testability = maintainable, reliable tests
- Acceptance of user story by Product Owner, not Tester

Traditional: Time to start



- Test Automation
  - during implementation phase not possible
  - during test phase most of the time not possible
  - most of the time at first after releasing a version
  - after release most of the time just a few defects found resp. high maintenance costs because of adjustment work to new version

Agile: Time to start



- Test Automation
  - integrated part of sprint
  - constantly maintained to keep it up-to-date
  - often starts before GUI exists („Test-First")
  - done by developer, tester and/or domain expert

Traditional/Agile: Testcase development

- **Traditional:**
  - Just a few „Automation experts" could write automated tests
  - Test script language != Development language
  - No acceptance by developers to learn new tool or new language
  - Missing or not up-to-date test documentation
- **Agile:**
  - Everyone in the team can/should write automated tests
  - Familiar development environment is used
  - Automated tests can be used for manual and automated test execution
  - Testers pick up programming knowledge
  - Uniform version control
  - More unit tests than GUI tests

Traditional/Agile: Test Execution

- **Traditional:**
  - QA team is responsible for execution of automated tests
  - Test execution is done in Test Management system
  - Just a few test runs because of high maintenance and results analysis effort
  - Only smoke tests were executed on a regular base
- **Agile:**
  - Agile team is responsible for test execution
  - Test code is maintained with code changes
  - Additionally automated tests get executed locally by developers
  - Test run on each build
  - Test tools get adapted
  - Performance and Load Test scripts can be reused

Why is Speed important for Automation?

- The faster you test the **more** you can test
- The faster you test the **more often** you can test

- Example:

Test cases	Configurations	Actions per test case	Response time per action	Delay added through the testing tool	Raw execution time (hours)	Total execution time (hours)
1000	5	10	0,3	1	4,2	18,1
1000	5	10	0,3	0,01	4,2	4,3

Tool Requirements in an Agile Environment

- Support agile workflows
- Interfaces for integrations
- Means of communication (Tester-Developer, Domain Expert-Tester/Developer)
- Integrations for development environments
- Test script language = Development language
- Using same repository for test and production code = uniform version control
- Use same test for manual and automated execution
- Fast, reliable test execution
- Test maintenance should be simple
- Reports, but without additional burden for teams

Test Automation in a global enterprise company

- **How?**
  - Use a test management system
  - Use effective means of communication
  - Integrate the service team as own Agile team with local product owner
- **ROI**
  - Unburden local Agile teams
  - Cost reduction

Q & A