

# Testing and Scrum

FOCUS ON EFFICIENCY

Fall 2007 Scrum Gathering  
Ralph van Roosmalen

**PLANON**  
INTEGRATED WORKPLACE MANAGEMENT SOLUTION

## Agenda

- Introduction
- The Classical Test Approach
- Organization
- Test Documentation
- Test Activities
- Recruitment
- Reporting
- Test Automation
- Lessons Learned

**PLANON**  
INTEGRATED WORKPLACE MANAGEMENT SOLUTION



# Introduction



## Planon > Personal introduction

- Ralph van Roosmalen, [r.vanroosmalen@planon.nl](mailto:r.vanroosmalen@planon.nl)
- Computer Science ba, studied at the Institute of Technology
- Certified Scrum master
- ISTQB Practitioner
- In Software Industry since 1997 as Software Developer, Project Manager and currently Test Manager and Scrum master at Planon.

## Planon > Company



- Standard software for Integrated Workplace Management Solutions
- Private Dutch Company, Nijmegen, ± 300 employees, ± 50 employees in Software development.
- Last five years at least 23% growth in revenue, and turnover of € 21 million Euro (2006).
- Active in The Netherlands, Germany, Belgium, UK, France, Spain, Austria, Switzerland, Poland and the United States.
- Goal: to become world leader in *Integrated Workplace Management Solutions in 2010*



## Planon > Software Development

- ± 50 employees
- Six scrum teams
  - Five working on a product
  - One support team
- All feature teams
  - Team members are interchangeable
- Using Scrum since 2005
- Product Owners are not part of Software Development, customer – vendor relation
- Will engage in outsourcing development in 2008 by using distributed Scrum teams



## Planon > Products

- Planon ProCenter Windows Client
  - Legacy product, ± 15 years old, Delphi
  - Supported by team SE
- Planon Self-Service
  - ± 5 years old, Delphi
  - Supported by team Net
- Planon Talk
  - ± 5 years old, Delphi
  - Supported by team Net
- Planon ProCenter Java Client\Web2.0 Client
  - 3 years old, J2EE environment
  - Supported by teams Alpha, Beta and Gamma



## The Classical Test Approach



## The Classical Test Approach

- Test team is asked to test a project
- Test Manager starts with Project Test Plan
- Test Manager or Test Leads creates Phase Test plan
- Start reviewing requirements and specifications
- Start writing test cases by test engineers
- *Wait for software, until it is almost too late to execute the test plans.*
- Start test execution phase
- Solve problems found
- 1<sup>st</sup> retesting

## The Classical Test Approach

- *Complaints from the project manager, why is the testing phase taking so much time.*
- Solve problems found during 1<sup>st</sup> retesting
- 2<sup>nd</sup> retesting
- Regression test and solve problems found during 2<sup>nd</sup> retesting
- Deliver software
- Project evaluation
- *Involvement in the next project too late: it has already started because the developers were finished with the previous project.*

## The Classical Test Approach

- Does not work in Scrum because...
- There is no test team
- You deliver potentially shippable software each sprint
  - There is no time to write test plans
  - You can't wait until the software is ready
  - There is no time to perform extensive retesting
- Requirements and specifications are lean, so
  - Reviewing of documentation cannot wait until it is finished
  - Writing test cases based on specifications is difficult
- There is no test manager within Scrum

## The Classical Test Approach

- To make testing work in a Scrum environment, you have to change...
  - The organization
  - Testing Documentation
  - Test Activities
  - Recruitment
  - Reporting
  - Regression testing approach



## Organization



## Organization

- Testers are integrated into the teams
  - So no separate test team
  - Testers are always up-to-date
  - Testers can easily communicate with developers, technical writers and functional designers
- Test manager is still there, but not in the project

## Organization

- Role of the tester
  - Ask questions
  - Bring people together
  - Act as the team's quality conscience
  - Testing
- Tester has to earn credit with the rest of the team to fulfil this role.



## Testing Documentation



## Testing Documentation

- Test Policy
  - High level document, used for all projects
  - Just 1 A4 in size
- Test Strategy
  - Per Test Level, used for all projects
  - Just 1 A4 in size per Test Level
- Project Test Plan
  - Describes the deflections compared to the Test Policy and Test Strategy

## Testing Documentation – Test Policy 1/2

- A Test Policy contains the following sections:
  - Mission
  - Organization
    - All testers hold the ISTQB Foundation certificate.
    - On average, each team that builds a software product should have one specialized tester on three developers.
    - The team is ultimately responsible for the quality of the delivered software.
  - Testing Approach
    - The testing approach is aligned with the values of the Agile manifesto.
    - The testing strategy is based on the product risk matrix.
    - An automated regression test is available for each standard Planon software product. The regression test covers at least the product's high risks areas.

# Testing Documentation – Test Policy 2/2

- Standards
- Quality Attributes
  - Functionality
  - Efficiency
- Test Improvement
  - The testing process is continuously improved by applying the improvement actions from the team- and a testing retrospective that is held after each sprint. This continuously improvement is embedded in our software process, Scrum.
- Evaluation of testing (Performance indicators)
  - Data is collected on test effort and defects; this data enables creating metrics to provide input for the test improvement process.



# Testing Documentation – Test Strategy 1/2

Unit test				
The testing of software components. Is planned and designed early in the life cycle, the tests are based on the detailed design specifications.				
<b>Objective</b>	Test the business logic and the application framework.			
<b>Responsibility</b>	The team is responsible; the developers are the operators and the testers in some cases the reviewers.			
	Risk I	Risk II	Risk III	Risk IV
<b>Entry criteria</b>	-	-	-	-
<b>Exit criteria</b>	<ul style="list-style-type: none"> <li>↻ Jalopy executed</li> <li>↻ Find Bugs executed; no Correctness bugs and Bad Practices are left</li> <li>↻ 100% tests successful</li> <li>↻ Unit tests are reviewed</li> <li>↻ High coverage</li> </ul>	<ul style="list-style-type: none"> <li>↻ Jalopy executed</li> <li>↻ Find Bugs executed; no Correctness bugs and Bad Practices are left</li> <li>↻ 100% tests successful</li> <li>↻ Unit tests are reviewed</li> <li>↻ High coverage</li> </ul>	<ul style="list-style-type: none"> <li>↻ Jalopy executed</li> <li>↻ Find Bugs executed and no Correctness bugs are left</li> <li>↻ 100% tests successful</li> <li>↻ Medium coverage</li> </ul>	<ul style="list-style-type: none"> <li>↻ Jalopy executed</li> <li>↻ Find Bugs executed and no Correctness bugs are left</li> <li>↻ 100% tests successful</li> <li>↻ Low coverage</li> </ul>
<b>Test process</b>	A developer creates Unit tests, often they are designed by a tester. The unit tests are reviewed by a tester.	A developer creates Unit tests, often they are designed by a tester. The unit tests are reviewed by a tester.	A developer creates Unit tests, in some cases they are designed by a tester. Sometimes they are reviewed by a tester.	A developer creates Unit tests.



# Testing Documentation – Test Strategy 2/2

<b>Milestones deliverables</b>	Unit tests, coverage results and Unit results.			
<b>Test case design techniques</b>	<ul style="list-style-type: none"> <li>↻ Boundary value analysis</li> <li>↻ Equivalence partitioning</li> <li>↻ Statement testing</li> <li>↻ Cause/Effect graphing</li> </ul>	<ul style="list-style-type: none"> <li>↻ Boundary value analysis</li> <li>↻ Equivalence partitioning</li> <li>↻ Statement testing</li> </ul>	<ul style="list-style-type: none"> <li>↻ Boundary value analysis</li> <li>↻ Statement testing</li> </ul>	<ul style="list-style-type: none"> <li>↻ Boundary value analysis</li> <li>↻ Statement testing</li> </ul>
<b>Type of tools that will be applied</b>	Find Bugs (Code analysis tool), Emma (Coverage tool), Unit framework			
<b>Environments in which the tests will be executed</b>	Nightly test environment, subset of unit tests in a continuous build environment.			
<b>Typical non-functional test types</b>	Efficiency.			
<b>Metrics</b>	<ul style="list-style-type: none"> <li>↻ Number of successful unit tests</li> <li>↻ Statement coverage of successful unit tests</li> <li>↻ Time behavior of the executed tests</li> </ul>			
<b>The approach to test automation</b>	All tests are automated in a unit test framework, their will be no manual testing in this level.			
<b>The approach to retesting and regression testing</b>	If a problem is fixed in the source, all the automated unit tests will be executed and there will be one or more new unit tests to prevent reintroduction of the problem.			



# Testing Documentation – Project Test Plan

- If a project does something different than described in the Test Policy or Test Strategies, this can be described in the Project Test Plan.



## Testing Documentation – Test Policy And Test Strategy

- Why write documentation? You say it is reverse engineering, so why write it down?
- Enables to discuss test approach with management and teams
- Used to train new employees
- Regularly check to see whether we are still on the right track
- Used in presentations to (potential) customers
- It's a living document, so we adapt it when necessary.



## Test Activities

## Test Activities – Sprint Planning

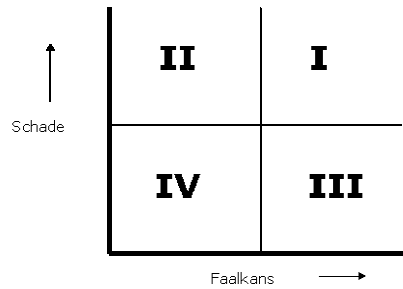
- Testers are present during the sprint planning
- Use risk analysis to identify testing tasks
- Testers estimate test tasks; testing capacity is finite

## Test Activities – Sprint

- Test the software
- Test Scrum; once a week
- Testers assist Developers in writing unit tests
- Testers review unit tests and/or specifications
- Testers personify the team's quality conscience
  - Daily priority of the team
  - Results unit tests
  - Keep track of problems

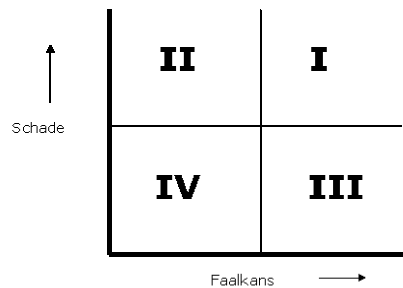
## Test Activities – Sprint

- Impact
  - Part of the primary company process
  - Possibility of corrupt data
  - Number of users
- Risk of failure
  - Tools and technology
  - What kind of software: maintenance or new software
  - Number of people involved



## Test Activities – Sprint

- Risk I
  - **Extensive** Exploratory Testing
  - Structured testing, all executed test are automated
  - Testing is done by the most experienced testers
  - Review of Unit tests
  - Review of the test cases on coverage and content
  - Code review
  - Review specifications
- Risk IV
  - Exploratory Testing
  - Some basic unit testing
  - Testing can be done by non-specialized testers



## Test Activities – Sprint

- ET is very important, also at high risk items
  - Writing test cases based on specifications is difficult
  - ET in combination with formal techniques
- You can't wait until the software is ready
  - Testers start with automated test scripts
  - Developers have to deliver software piece by piece

## Test Activities – End of the Sprint

- Testers participate in the Sprint Retrospective
- Testers are present during the Sprint Review meeting
- Test Retrospective, retrospective about the test process
- Prepare the next sprint, investigate product backlog



# Recruitment



## Recruitment – Process

- Looking for an “Agile Tester”
- Interview with Development Manager or Test Manager and Recruitment Manager.
- Workshop
  - Four hours – a small effort for a new job
  - Every one can say (s)he has knowledge of software testing, but can they prove it?
- Final interview

## Recruitment – Workshop

- 60 Questions about testing
  - What does the applicant know about testing (ISTQB)
  - Possibility to discuss testing with the applicant
  - How does (s)he handle stress
- Describe at least 20 test cases to test the Font dialog in Word
  - How creative is the applicant
  - Does (s)he use test techniques
  - How does (s)he describe the tests
- Case “communicate with the developer”
  - How does (s)he handle with the stereotype developer
  - Can (s)he set the correct priority and severity

## Recruitment

- We believe that soft skills are very important for testers in a Scrum environment
  - Communication
  - Flexibility
  - Collaboration
- Test techniques is something you can learn



## Reporting



## Reporting

- Test Reports
  - No separate Management report
  - No *hidden* link on a corporate website
- But...
  - Visible for all team members and stakeholders in a public place
  - Simple and “less is more”

## Reporting

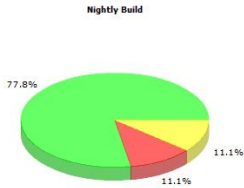
- What to report?
  - Open issues per module
  - Open issues per team
  - Open issues displayed in time
  - % successful unit tests
- What to report depends per organization, but...
  - Report per team, not per individual
  - Use colors: green is good and red is bad (In Western oriented countries)
  - Keep it really simple

## Reporting – Monitors in the hall



# Reporting – Example report

## Nightly Build



Successful: **77.78 %**  
 Failed: **11.11 %**  
 Running builds: **11.11 %**

## Continuous Build

Date	Result
ma, 15 okt 2007 12:45:31 +0200	Build passed
ma, 15 okt 2007 11:52:00 +0200	Build passed
ma, 15 okt 2007 10:58:43 +0200	Build passed
ma, 15 okt 2007 10:05:04 +0200	Build passed

## Build History for Nightly Building Environments

	now	14	13	12	11	10	09
Oracle (DIST_PEE50)							
MSSQL (DIST_PEE50M)							
Cumm. Oracle (DTS2_PEE50)							
Customer Latest Oracle (PTST_CUSTLATEST)							
Customer Latest MSSQL (PTST_CUSTLATESTM)							

- Build OK
- Build Failed
- Build Running, before test stage
- Build Running, in test stage
- Build did not run

## Unit

Oracle Plain	<b>96.76 %</b> 8 oktober 2007 23:04
Oracle Obfuscated	<b>97.89 %</b> 6 oktober 2007 8:23



# Test Automation



## Test Automation – Why

- There is no time for extensive manual regression testing; automation is essential.
- Necessary because of incremental iterations and to support refactoring
- As Product Vendor we need to support our products for years
- Instantaneous feedback: all tests are run each night
- Makes the daily work for testers more fun

## Test Automation – Which tools

- JUnit framework
  - No testing of classes but on a higher level
  - Used by developers to test framework and business logic
- QF-Test
  - Commercial product for automating tests of Java applications with a graphical user interface
  - Used to test our Swing client
- Selenium
  - A test tool for web applications
  - Used to test our Web2.0 client

## Test Automation – Approach

- Planon Test Policy: “An automated regression test is available for each standard Planon software product. The regression test covers at least the product’s high risks areas.”
- Developers and tester discuss test automation
- Developers builds unit tests, testers reviews them or describes the unit tests, depends on the risk and available capacity .
- Tester builds a QF-Test test script to touch all GUI components and test main use-case.
- If there is special coding needed for the Web2.0 client, the tester builds a Selenium test script.
- Developer and Testers need to communicate to have the right test automation approach!

## Test Automation – Approach

- Tool smith
  - Responsible for analyzing the automated test results
  - Develops and maintains the automated test framework
  - Reviews the tests scripts
  - Prevent this role becoming a bottleneck in creating automatic test scripts



## Lessons Learned



### Lessons Learned – General

- Make no statement about the quality at the end of the sprint
  - quality should always be good, and if not a product backlog item is not finished
  - Quality is a team issue not the responsibility of testers
- Testing in Scrum is different, but you can still use the same old test techniques; use them lean

## Lessons Learned – General

- How to handle open (customers) issues
  - Prioritize issues
  - Add the issues to the product backlog
  - If you do not want to solve an issue, cancel or close it.
- If you plan a stabilization phase, you are going to need it.

## Lessons Learned – Test Manager

- Coach testers
- Review test approach of a product backlog item every sprint of every team
- Develop a vision on testing together with testers
- Develops and maintain the Test Policy, Test Strategy and Test Project Plan
- Increase the testing knowledge of testers; for example, hold a monthly professional circle

## Links

- Agile Testing User Group, <http://tech.groups.yahoo.com/group/agile-testing/>
- International Software Testing Qualifications Board, [www.istqb.org](http://www.istqb.org)
- Planon, [www.planon-fm.com](http://www.planon-fm.com)
- QFS, [www.qftest.com](http://www.qftest.com)
- Scrum, [www.controlchaos.com](http://www.controlchaos.com)
- Scrum User Group, <http://groups.yahoo.com/group/scrumdevelopment/>
- Selenium, [www.openqa.org](http://www.openqa.org)
- TestComplete, [www.automatedqa.com](http://www.automatedqa.com)