Case: Implementing Agile Testing in an Agile Project

Kari Kakkonen

Director, Testing and Methodologies and Senior Testing Consultant, Endero

Chairman, FiSTB

ENDERO

The interactive art of business success

Innovative and business-oriented ICT service provider

Fastest growing technology company in Finland #1 (Deloitte 2009)

Customers include financial institutions, product companies, industries, commerce and services and public sector

Service centers in Helsinki and St. Petersburg

Over 150 employees

Strong emphasis on Microsoft technologies

Part of Know IT Group with 1 500 employees

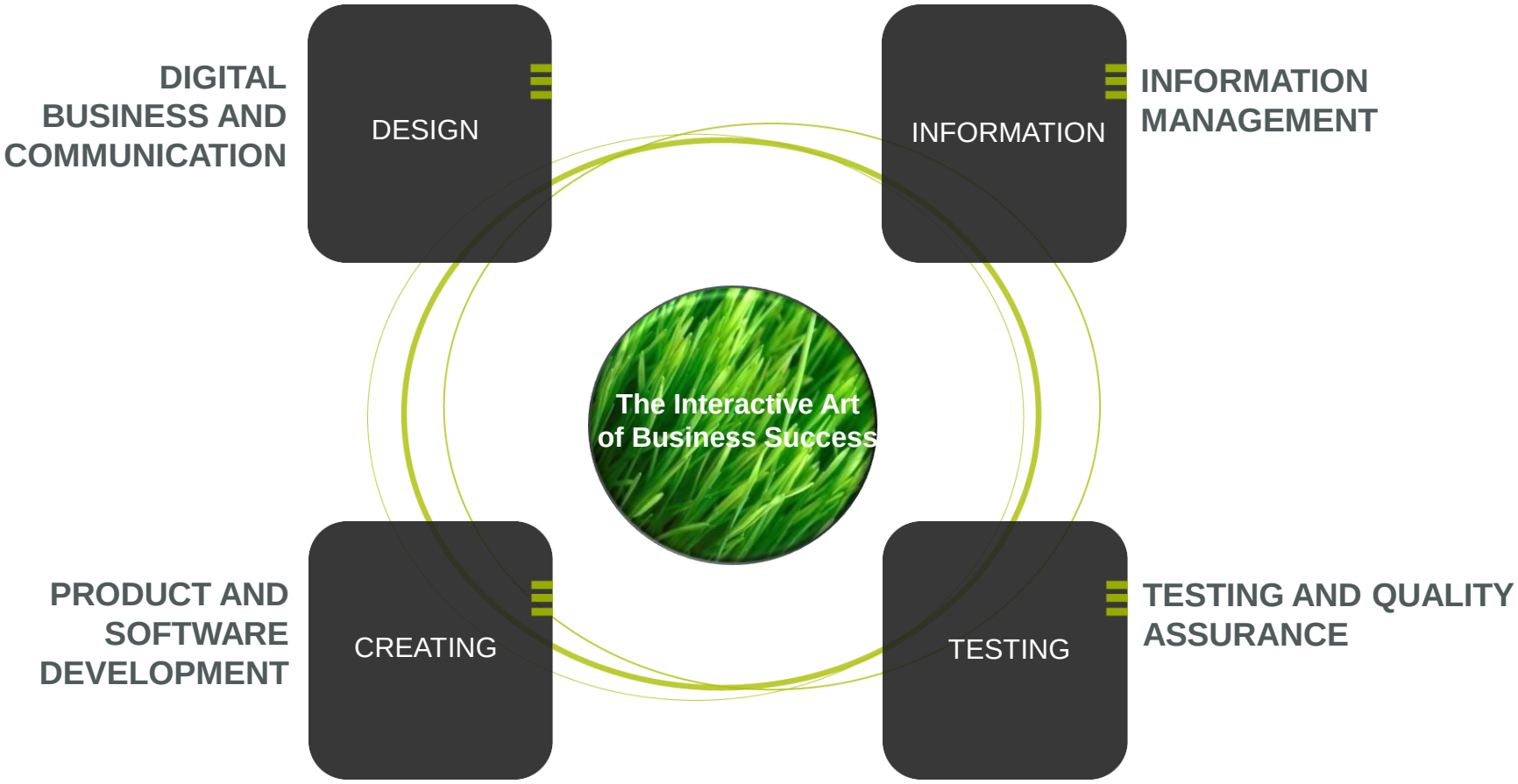
Our values:
Alignment - Appreciaton - Advancement

Know IT

- Founded 1990
- Offices in Sweden, Norway, Finland, Estonia, S:t Petersburg, Seattle
- 1500 employees
- Listed on (OMX) Nordic Stock Exchange in Stockholm
- Largest share holder– Atine Group

know it

OUR SERVICES
The Interactive Art of Business Success



ENDERO

Finnish Software Testing Board

ISTQB International Software Testing Qualifications Board

- Coordinates testing certification in Finland
- Organizes in Finland
  - ISTQB Foundation –exams in Finnish and English
  - ISTQB Advanced –exams with UKTB

http://www.ttlry.fi/fistb/

http://www.istqb.org

http://groups.yahoo.com/istqb-fi

FiSTB Finnish Software Testing Board

AGENDA

1. Some definitions
2. Case (with inserts about terminology)
   - Starting situation
   - Goals
   - Procedure
   - Discussion of success

ENDERO

1. SOME DEFINITIONS

1. **Some definitions**
2. Case

TYPICAL TO AGILE TESTING

- Adapts to changes very fast
- Not plan-driven, but lives with the project
- Testing is done from the beginning of the project, not just in the end of it
- Emphasizes the team, humanity/human-centeredness and cooperation
- Tests iteratively piece by piece
- Customer is involved even in all testing phases
- Continuous integration
- Automation of testing at least in unit-testing
- It is more important to find bugs than to write comprehensive documents

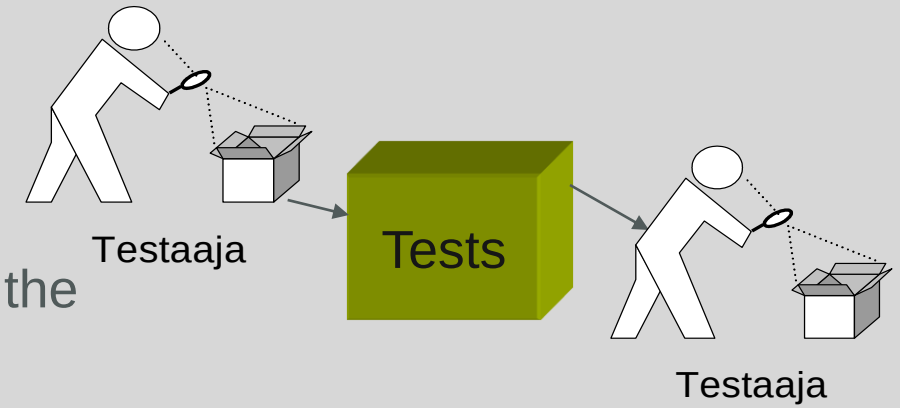
WHAT AGILE TESTING IS NOT?

- A separate phase at the end of software development project
- Unsystematic
- Documentation free
- Random
- Uncontrolled
- Straightforward operation without feedback

COMPARISON OF TESTING APPROACHES

Plan-driven

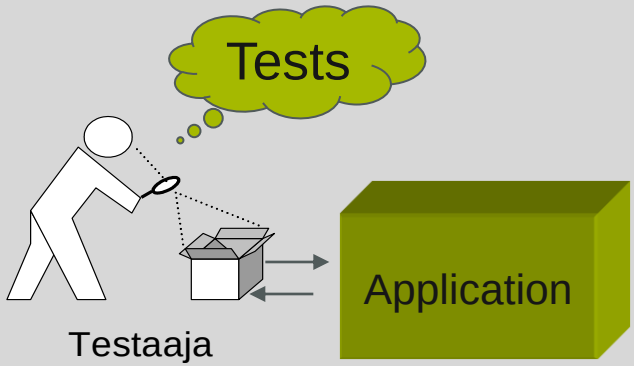
- Test cases are created and documented first
- Testing can be executed later by the same or a different person



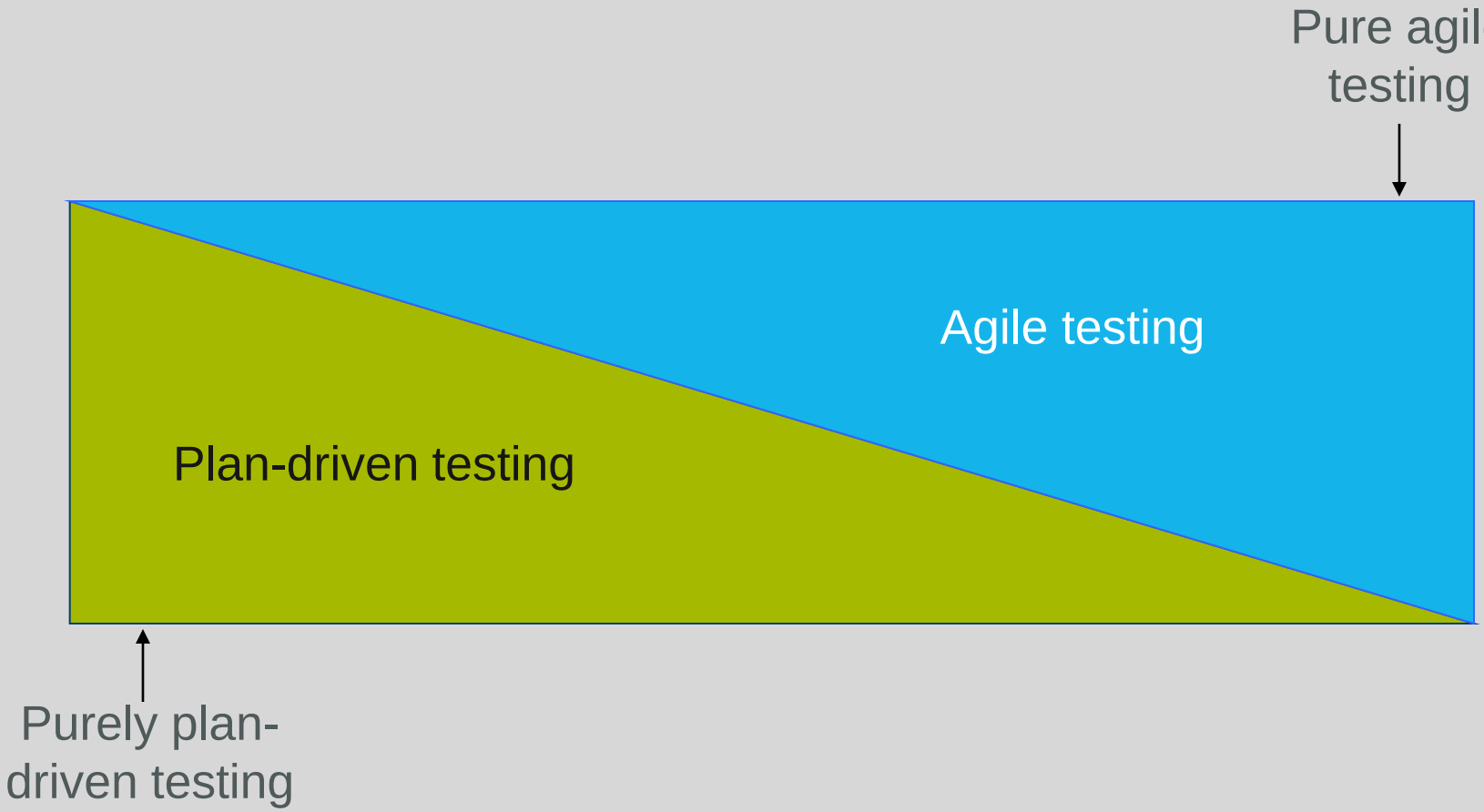
VS.

Agile

- Tests are planned and executed at the same time and they are usually not documented in detail



PLAN-DRIVEN VS. AGILE ENTITY



PLANNING OF AGILE TESTING

- Start with a light overall test planning
- Choose the proper tools
- Create a testing environment
- Prioritize things to be tested in each iteration (sprint)
- Create test cases for the first iteration on the agreed (light) accuracy level
- Outline tests for next iterations
- Automate testing

CHALLENGES IN AGILE TESTING

- The identification of omissions
  - Are we testing the right things?
  - Is testing sufficiently comprehensive?
  - Is continuous integration working?

THINGS TO CONSIDER IN AGILE TESTING

- Expand testing, for example:
  - A separate system testing level
  - Security testing already at the early stage of the project
  - Start performance testing before the product is complete
  - Internal acceptance testing when the team believes that the product is ready

SPECIALIZING AS AN AGILE TESTER

- Must have a good understanding of the agile software development process and experience from agile projects
- The ability to act and to envisage is achieved only with adequate experience
- An agile tester must understand
  - business processes
  - software development (more than in plan-driven processes)
  - the politics between organizations and people
  - psychology between people and groups
- A good tester seamlessly fits in and adapts to the team

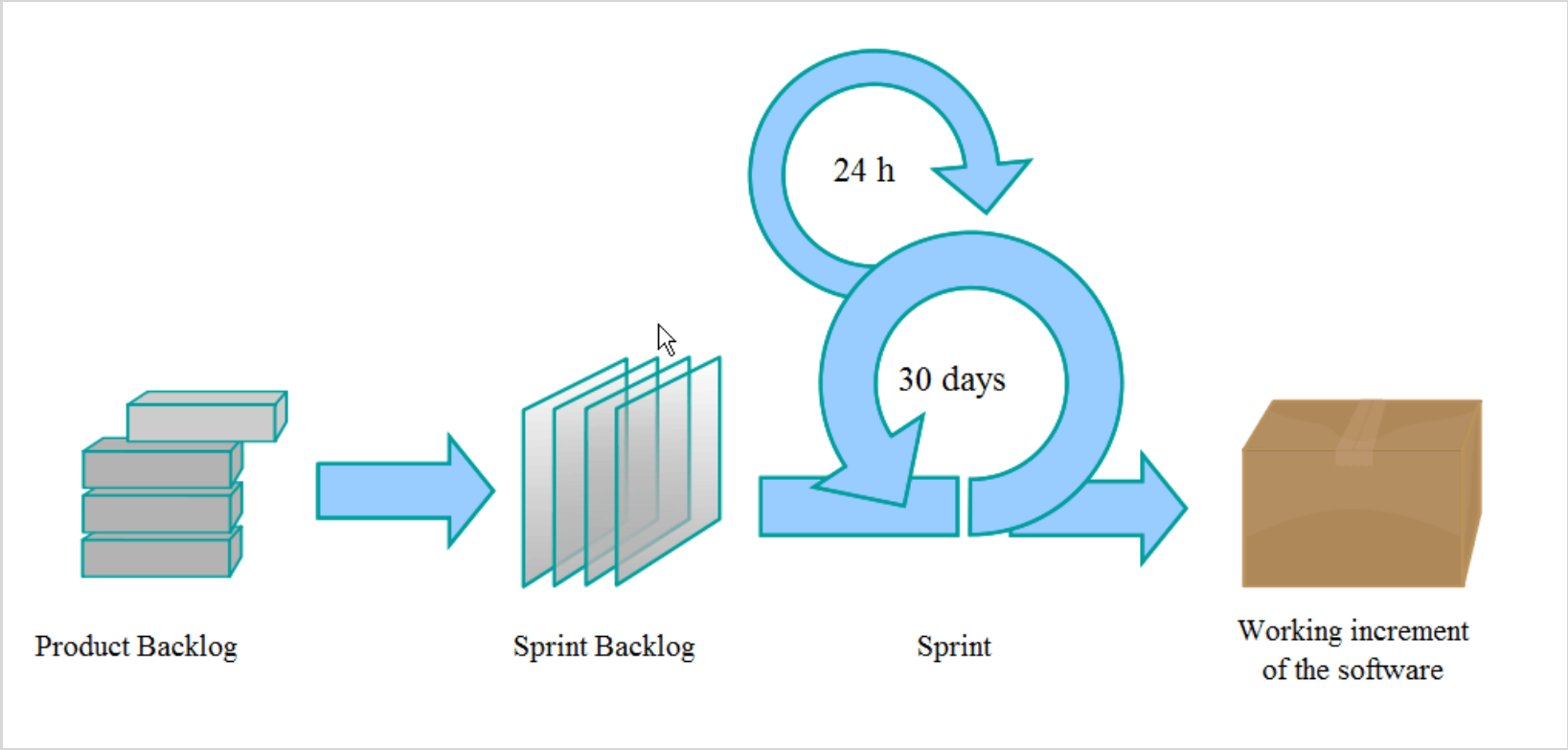
2. CASE

1. Some definitions
2. **Case**

STARTING SITUATION OF THE CASE

- Scrum was selected as a software development model for the project
- Project risks derived from options
  - No testing → delivery has a quality risk
  - Agile testing → need to compromise on the testing coverage
  - Traditional testing → ruin effective functioning of the scrum-team
- Project budget has constraints
- Agile approach was selected
  - Delegate part of the responsibility for Developers

(TERMINOLOGY SLIDE)
SCRUM PROCESS



[http://upload.wikimedia.org/wikipedia/commons/5/58/Scrum_process.svg]

(TERMINOLOGY SLIDE)
TESTER'S DAY IN SCRUM

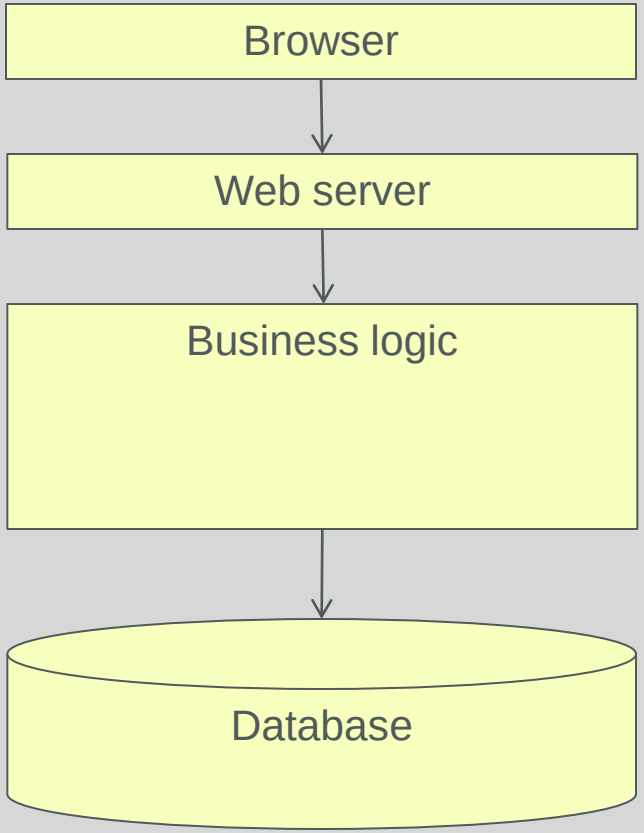
- Daily Scrum-meeting
- Keeping up-to-date in the progress of developers' work
- Testing (includes parallel test design and test run)
- Taking care of test coverage
- Clearing up defects and reporting
- Maintaining tests and test automation
- Avoiding of test lack (quality debt)
  - Keep up with the pace of the developers
  - Avoid the attitude: ”rest of the tests can be done later”

BACKGROUND OF SCRUM TEAM

- Compact scrum team
- On-site customer: daily in touch with scrum team
- Customer is in different location from the team
- 10 persons with developer background
  - Scrum master, 9 developers
  - Scrum approach is known to different extent
  - An updated scrum training given to all in the beginning
  - Everyone has programming experience at least 3 years
- Team has remote connections to customer (video conference , Skype)
- Sprint lasts one month

SYSTEM UNDER TEST

- Multi-level architecture
  - Browser user interface
  - Web server
  - Business server
  - Database server



THE MAIN CONSTRAINTS

- Not many testers → developers have the main responsibility for testing
- Tester's role consists of
  - Test the system as whole from many perspectives
  - Ensure that each developer takes care of his own area
- Risk
  - All testing is left to the few testers

NEEDS FOR THE START

- Had to find 2 testers to the team
  - Recruitment or transfer from another project → transfer was selected
  - Need to be able to test
  - Need to keep the development team in control
    - Independent
    - If necessary testers must be able to confront developers
    - Test manager type role, need charisma
- Need to get experience from agile testing quickly
  - Senior testing consultant works in the beginning as an Agile Testing Coach

GETTING STARTED

- Senior testing consultant was involved in the first two sprints
  - Created agile testing approaches
  - Helped the team to act independently
    - Testers
    - Developers
  - 50% time use in the first sprint
  - 25 % time use in the second sprint
- Testing in the first sprint
  - Created practices
  - Tried different testing tool options
  - All testing was done manually

ENDERO

APPROACH

- Objective
  - When are we successful? – Aim to support the delivery
  - Definition of Done
- Testing does things that are needed to meet the objectives
  - Things, that customer wants will be done
  - Ensure adequate testing coverage
- Choices based on Agile Manifesto

(TERMINOLOGY SLIDE)
AGILE MANIFESTO

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.

(http://agilemanifesto.org/)

ENDERO

TESTING COVERAGE

- Applied a sufficient number of perspectives
  - Test levels
  - Test types
  - Roles
- Tester’s responsibility to test comprehensively requires
  - Knowledge of test design techniques
  - Knowledge of exploratory testing
- Non-functional testing is taken into account as separate test tasks during the Sprint or is done by the customer separately

MEANS

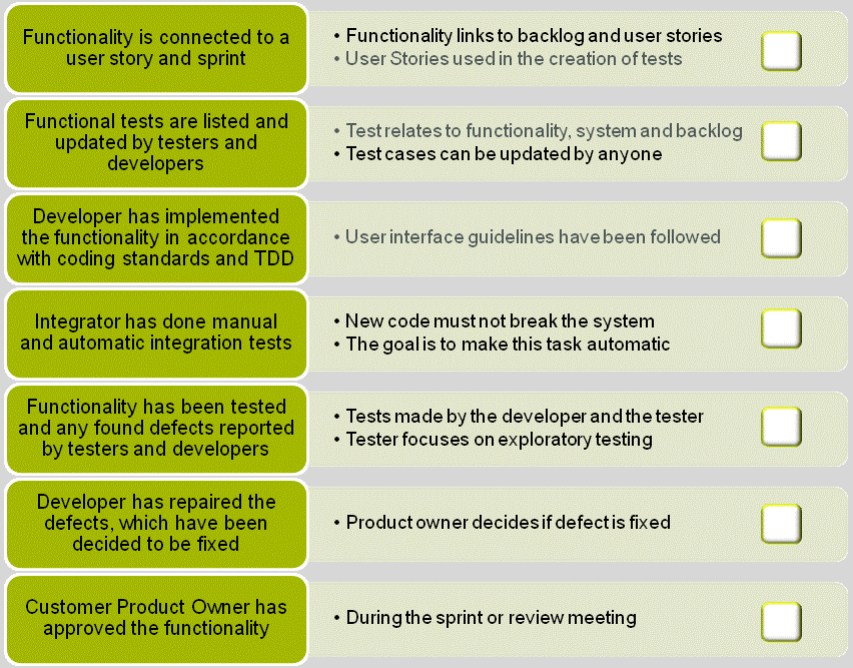
- Product backlog and sprint backlog
- Checklist card of each task in product backlog
- Master test plan
- Developer’s part of the system testing is more than a smoke test
- Tester attends to daily scrum meeting

PRODUCT BACKLOG AND SPRINT BACKLOG

- Sharepoint –based backlog-website
- Product backlog
  - General monitoring of development tasks, contains PASS/FAIL from testing perspective
  - Product backlog –level works as internal system testing
- Sprint backlog is a scheduled part of product backlog, own view
  - Development tasks that include also unit testing
  - System testing is a separate task for each functionality
  - Task level follow-up of testing

CHECKLIST CARD FOR A SINGLE TASK IN PRODUCT BACKLOG

- Printed for every task
  - Follow-up with tally marks for control
  - As a checklist
  - Not a management tool (different means for that)
- Printed also as reminders on the walls



ENDERO

Functionality is connected to a user story and sprint	• Functionality links to backlog and user stories • User Stories used in the creation of tests
Functional tests are listed and updated by testers and developers	• Test relates to functionality, system and backlog • Test cases can be updated by anyone
Developer has implemented the functionality in accordance with coding standards and TDD	• User interface guidelines have been followed
Integrator has done manual and automatic integration tests	• New code must not break the system • The goal is to make this task automatic
Functionality has been tested and any found defects reported by testers and developers	• Tests made by the developer and the tester • Tester focuses on exploratory testing
Developer has repaired the defects, which have been decided to be fixed	• Product owner decides if defect is fixed
Customer Product Owner has approved the functionality	• During the sprint or review meeting

TOOL OPTIONS

- Selection criteria
  - Cost-effectiveness, good user experience, ease of learning, supplier requirements
- All the tools were targeted for the first sprint, but only defect management tool got to full use. By third sprint all tools were adopted.
- Tools were open source
  - Bugzilla (defect management)
  - Selenium (user interface testing automation)
  - JUnit (automation of back-end system testing)
  - Bromine (test management system)

MASTER TEST PLAN

- 10 pages, very light and informative, concise text
- Cases
  - Presentation of the project in terms of testing
  - Excluded matters (Usability testing)
  - Description of architecture (1 picture)
  - Roles and responsibilities –table
  - Test levels and test types (perspectives)
  - Testing process (explanation and picture)
  - Test cases (location and methods)
  - Defect management (location, process, instructions)
  - Tools (listing)
  - Task list (management checklist, no team tasks)

ROLES FROM TESTING PERSPECTIVE

Role	Tasks
Tester	Designing and testing of system test cases Exploratory testing Follow-up of check list cards Reporting of defects
Developer	Manual and automated unit testing, Test Driven Development Fixing of defects Adherence to coding standards System testing part 1 of own functionalities, reporting of defects Supporting of testers in testing
Integrator	Selected developer will always make the build Smoke test
Scrum master	No specific testing role
Product owner	Team member, who makes the difficult decisions Supports system testing and reporting
Customer product owner	Approves the functionalities during the sprint or in final meeting Acceptance testing Support system testing and development

TEST LEVELS

- Test levels
  - Unit testing: Test Driven Development
  - Integration testing: continuous integration
  - System testing: mainly roles
  - Acceptance testing:
    - Customer tests single functionalities as soon as these are available
    - All functionalities included in sprint are tested during the last few days of the sprint
    - Acceptance in sprint review
- Not the means for working phase-by-phase, but giving responsibility

TEST TYPES

- Test types
  - Functionality testing on each level
  - Performance testing
  - Automated regression testing after each change
  - Smoke test after each installation of a new build
  - Re-testing of fixes by tester
- Giving responsibility roles for each testing type was considered important

TESTING PROCESS

Product backlog, sprint

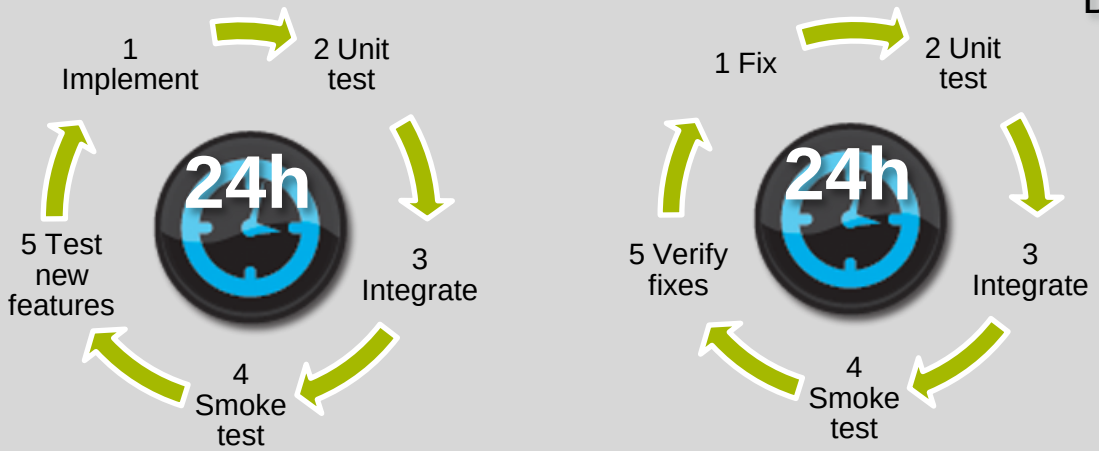
Requirements, functionalities & user stories

List of test cases

Functional testing and exploratory testing

Testers

Developers



Day1 Start

Day 2-17 Implement functionality

Day 18-20 Fix defects

ENDERO

TEST CASES IN DEVELOPER'S TESTING

- Manual unit testing based on user story
- Mark of passed tests to the check list card
- Automated tests to selected basic functionalities
- Test Driven Development method was compromised in the beginning
  - Adopted only in later sprints

TEST CASES IN SYSTEM TESTING

- Traditional, but tests designed in test idea level only with good coverage, 1-2 tests per functionality
  - 20% all of tests
  - Planning in the beginning of the sprint and just before need
  - Level of detail chosen for the tester herself
  - Part of these have been changed to automated tests in sprint 3
    - Regression testing
    - Browser versions
- Exploratory testing is focused only to functionality areas, not on specific functionalities
  - 80% of all tests
- Test log is used in both types of testing

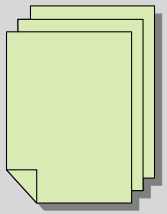
(TERMINOLOGY SLIDE)
EXPLORATORY TESTING - TERMS

- Adventure may go to sidetrack as long as you come back to mainroad again (Kaner)
- Session-based test management (Bach)
- Testing area: a bunch of functionalities
- Testing session
  - Duration about ½ - 2 hours
  - Time span of concentrated work is about 20 minutes
  - Getting back to work takes about 20 minutes

(TERMINOLOGY SLIDE)
LEARNING IN EXPLORATORY TESTING

Observations

Reporting



Interpretation

Testing

Designing actions

Opinion-forming

After reference: Psychology of Usability, Sinkkonen et al.

(TERMINOLOGY SLIDE) CHARTER

- What will be tested?
- What documents are available?
- What kind of errors are being sought?
- Tasks and what test techniques will be used?
- Targets and outputs (for example reports)

Reference: A practioner's guide to software test design. Copeland
Ref. Exploratory testing: A multiple case study. Itkonen, Rautiainen

ENDERO

(TERMINOLOGY SLIDE) CHARTER

Area	Coverage and working hours	Practice	Documents	Result	possible errors	Risks
Main page	100% Path coverage (direct paths) and the most common (80% used) loops 10h	Scripting with Functional Tester-tool	Main page display description document, navigation map (COMING FROM DEVELOPMENT)	All pages and the shopping cart are available		R1. Customer's all selected items are not added to order, Effect: 20 eur/buyer, probability 5% R2. Order can not be completed after interruption, 5, probability: ??
Shopping cart	5h		Shopping cart-UC.doc (use case)	Shopping cart can be used in the same way as a real shopping cart	☐ The same product can not be added to shopping cart several times ☐ Emptying the shopping cart causes an execption ☐ R1	
Order	?		Order-UC.doc?		☐ R2	

Reference: www.satisfice.com/articles/sbtm.pdf

ENDERO

(TERMINOLOGY SLIDE)
TESTING DASHBOARD AS A TEST REPORT - AN EXAMPLE

Test area	Workload	Coverage	Quality level/risks	Comment
Main page	!Interrupted High, 5h	Very high [all parts + stress tests etc..]	49: 1435, 36: 1469, 42: 1501	wait for more pictures of user interface
Shopping cart	!Started High, 2h (reserved 4h)	Low [main parts to testing] (High)	81: 1425 [probability 9 x effect 9; error number. 1425]	
Order	!Done 6h (reserved 4h)	High [all parts]		
Feedback	!Not done Low (reserved 1h)	Low [main parts to testing]		

MEASUREMENT

- Metrics:
  - Defect reports as status listings
  - Defect numbers per status (open, fixed)
  - Defect numbers per more detailed status (…named)
  - Number of passed Product backlog items in testing
  - Workload of testing tasks in burndown chart
- Team follows and guides itself with the metrics mentioned above

THOUGHTS ABOUT THE CHOICES MADE

- Main goal
  - Testing was able to ensure, that there was not such situation, that client product owner would not accept the product in the sprint review meeting
- First there were ideas of Test Driven Development
  - Team did not have enough knowledge – TDD was left for later adoption
- Adoption of Agile testing
  - The meaning of agile testing developed together with the team
  - Activities changed slightly during the first Sprint
- Work loads and tasks of exploratory testing are in danger of being left out of the sprint backlog, without guidance

SUMMARY OF AGILE PRACTICES

- Development lifecycle fully complies with the Scrum
- Important defect fixes were not left to the following Sprint, only a few minor defect fixes were moved to the next sprint tasks
- Each sprint produced functionalities customer approved
- Plans described interactions and responsibilities more than precise plans about how things should be done and in what order
- All roles were in one team
- Significantly less testing documentation than in traditional project
- The most important reports were defect reports and metrics derived from them - worked regardless of the level of test case descriptions