Mark Rotteveel
Who am I

- Mark Rotteveel
- Born in 1979
- Nerd since 1989
- Likes: reading, computers and beer

- In IT since 1999, tester since 2006
- Working for Pluton since 2008
- Currently as tester at TomTom
Overview

- The basics of scrum
- Planning, Sizing and Tracking
- Why use scrum
- Scrum & testing
- Scrum & open source
The basics of scrum

Source: http://scrumtraininginstitute.com/home/stream_download/scrumprimer
Basics: components & roles

- Product Backlog
  - Userstories
- Tasks
- Sprint (timebox)
- Product
- Demo

- Product Owner
- The Team
- Scrum Master
Basics: workflow

- **Product Owner updates Product Backlog**
  - New requirements
  - Priority
- **Team selects stories from backlog**
  - Sizes stories (effort, complexity)
  - Commits to stories for sprint
  - Breakdown in tasks
Basics: workflow (2)

- **Daily standup**: short status update within Team
  - Track progress
- **Designing, developing, testing, etc**
- **At end of sprint**: Demo of stories finished
  - Potentially shippable product
  - New insights for requirements/user stories
- **At end of sprint**: Retrospective
  - Lessons learned for future sprints
Planning, Sizing and Tracking

- Planning of a sprint is based on storypoints and historic velocity
- Velocity is the number of storypoints completed in a given sprint (or day)
- Every story is sized in storypoints based on expected complexity, or time required
- The team commits to a number of storypoints (and stories) for the next sprint
Planning, Sizing and Tracking

- Progress in a sprint is tracked with a burndown-chart

Source: http://en.wikipedia.org/wiki/Burn_down_chart
Why use scrum?

- **Flexibility**
  - Early feedback
  - Priorities change
  - People change their mind
- **Early shippable product**
- **Team Happiness**
- **Especially useful when there are a lot of unknowns**
Scrum & Testing

• Testers embedded in The Team
  - Focus
  - Quick turnaround time
  - Interaction + knowledge sharing

• Testers outside The Team
  - Bigger picture
  - Regression testing
  - User / Production acceptance
  - 'Independence'
Scrum & Testing : Embedded

- Create testsuite for current userstories
  - Automated
  - Manual
- Work with developers to get clear requirements
- Quick feedback during the sprint
- Guard quality of current userstories
- Handover testsuite + stories to Outside testers
Scrum & Testing: External

- Perform regression testing
  - Testsuites provided by embedded testers
  - Testsuites developed by testers
  - Based on bug reports
  - Based on customer feedback
- Guard overall quality (including old + new userstories)
- Feedback to scrumteam(s) to improve quality of future userstories
Scrum & Testing: No external team

- External team is optional
- Fully automated regression suite required if no external team is available
- All team members (developers + testers) are responsible for keeping tests from failing
Scrum & Testing: Software quality

- A userstory or task is only done when
  - Developers have completed the story
  - Testers have created tests
  - Quality of the story is good
  - The new code doesn't break existing functionality (failure of regression tests)

- Quality is a joint effort!
Scrum & Testing: Pitfalls

- Lack of (detailed) requirements => Low quality tests or lack of coverage
- Focus on current user stories can degrade quality of older functionality
- Embedded testers lose objectivity
Scrum & Open source

• Using scrum for open source development is possible

• Requires
  – One 'Product Owner' who sets the goals and priorities
  – Focus as a team
  – Team located in one location (or at least multiple members per location)
Scrum & Open source

- **Not useful when**
  - Geographically dispersed members
    - Scrum activities like standup, planning and retrospective lose effectiveness
  - Lack of cooperation or common goal
  - Lack of shared ownership
Discussion & Questions