TNG TECHNOLOGY CONSULTING

Java Development

With Scrum

16. Februar 2009

Gerhard Müller, Markus Eberle

Java User Group München

TNG Technology Consulting GmbH, http://www.tngtech.com

http://www.jugm.de

Gerhard.Mueller@tngtech.com, Markus.Eberle@tngtech.com

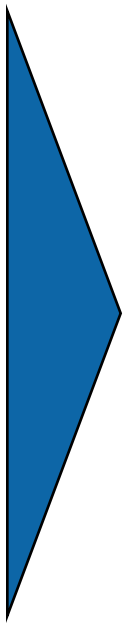
TNG TECHNOLOGY CONSULTING

Software Development Observations

- Some software is highly complex
  - Only experts understand problem domain
  - Tremendous communication effort necessary
- Software development is comparable to new product development, not manufacturing
- Requirements are often unknown a priori
- Moving target – requirements change during development
- Minimizing time to market becomes more important
- Technology innovation rate is speeding up

Agile Manifesto

Important	*More important*
Processes and tools	Individuals and interaction
Comprehensive documentation	Working software
Contract negotiation	Customer collaboration
Following a plan	Responding to change



http://www.agilemanifesto.org

Introducing Scrum

- Empirical process for managing the development and deployment of complex products
- Defined by Ken Schwaber and Jeff Sutherland in 1991-1993
- Based on the values of the Agile Manifesto
- Differs from other processes by applying an empirical process control

Source: TNG

Introducing Scrum

- Defined process control model:
  - Input is known well in advance
  - Output can thus be produced repeatedly in the same quality
- Empirical process control model
  - Input is not known or understood completely in advance to produce the same output each time
  - Problem is complex or technically advanced

Source: TNG

Scrum Roles

Team

- Responsible for success and delivery of the product increment
- Empowered to do anything needed to achieve the Sprint goal
- Self organizing, cross functional and autonomous

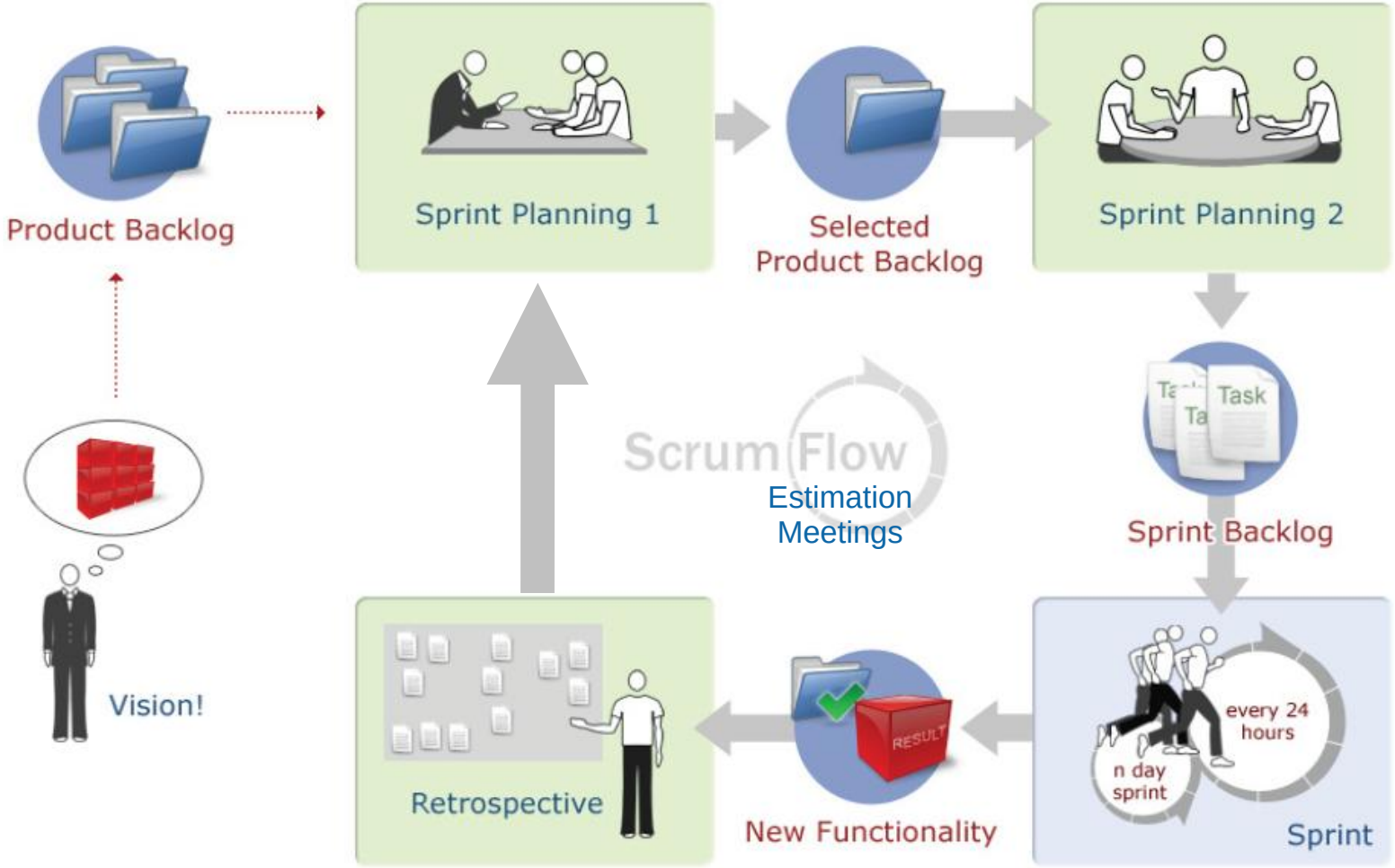
Product Owner

- Responsible for the profitability/return on investment of the product
- Defines and prioritizes requirements of the product

Scrum Master

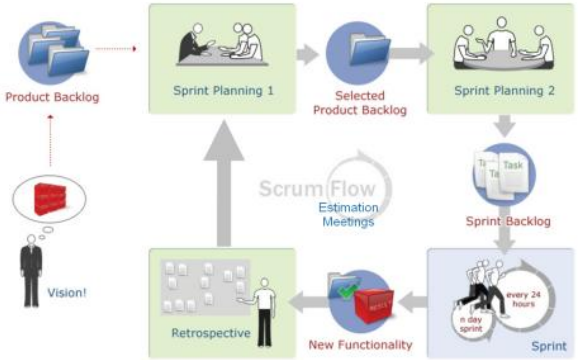
- Responsible for the success of Scrum
- Ensures everyone on team can fully concentrate on productive work

Scrum Flow: Artifacts & Meetings



Source: http://www.infoq.com/minibooks/scrum-checklists
(free registration required), modified

Agenda

- Introduction
- Scrum Overview
- Scrum Consequences
- Our Example-Project with Ingredients & Definition of Done
- 

- Our Lessons Learned
- Discussion & Questions

Scrum Consequences

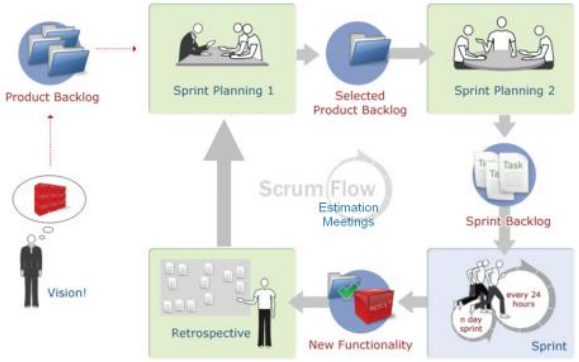
- Productivity increases
- Highly transparent development process
- Organisational deficiencies become visible
- Business retrieves control of development goals
- Much higher business involvement required
- Good engineering practices become a necessity
  - Deliverable product after each iteration require fast automated build process
  - Scrum's definition of “done” necessitates automated test execution
- Feedback, feedback, feedback!

TNG TECHNOLOGY CONSULTING

Rhythm / Scale

- Test case / Pair Programming
- Check in / Continuous Integration Build
- Task / Task Board / “Issue”
- Feature / Review
- Iteration / Sprint Demo / Retrospective
- Release, Release Train / End User Feedback
- Product / ROI
- System

Agenda

- Introduction
- Scrum Overview
- Scrum Consequences
- Our Example-Project with Ingredients & Definition of Done
- 

- Our Lessons Learned
- Discussion & Questions

Disclaimer

- TNG is an Atlassian Partner (because we love their products)
  - So there is lot of 'Jira' and 'Confluence' ☺
- Our experience demonstrated here is a concrete example, other ways may also work
  - Startup
  - 2 week iterations

Ingredients

- Wiki
- Issue Tracker
- Version Control Software
- Mind Mapping Software
- Artifact Management

Source: TNG

Ingredients

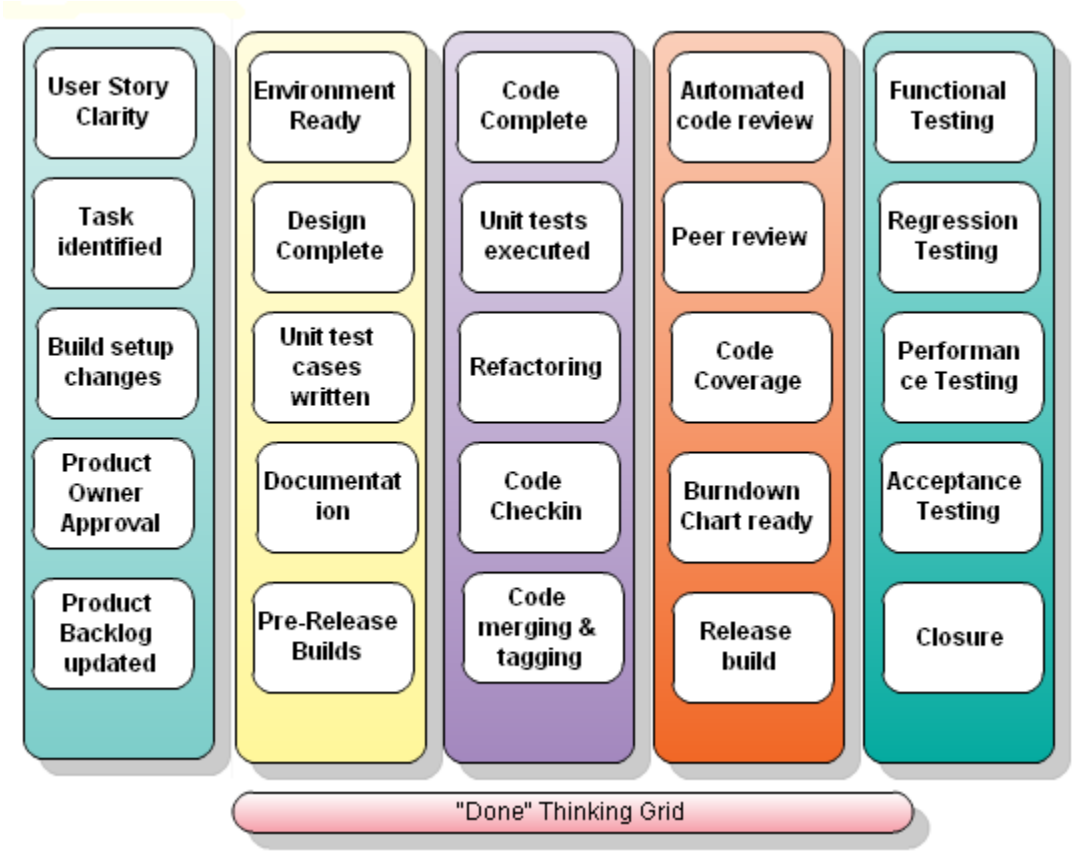
- One Room
- Digital Camera
- DIN A6 File Cards
- Development Server
- Planning Poker Cards
- Task Board

Source: TNG

This is **NOT** agile!

Source: http://www.dilbert.com

Definition of Done (DOD) thinking grid



TNG TECHNOLOGY CONSULTING

Done may be different for each team

Definition of Done

A user story is done if *all of its identified tasks* are

- implemented
- reviewed
- tested
- documented
- deployed
- making the product owner happy.

The remainder of this page details the parts of "done".

Implemented

Implemented code keeps the following principles:

- adheres to the coding styleguide
- code is compiled and run against the current version in source control
- code is commented according to the coding styleguide
- all FIXME/TODO tags in the code have been attacked (this is checked by a Maven plugin). If they are not resolved yet, either the Product Owner or [redacted] had to agree.

It must also be possible to *automatically build* the implemented code. How this is done depends on the kind of task.

- Java code must be integrated into the central Maven-based build
- Native Linux applications, Firefox plugins etc. must be packageable as APT. The process of packaging must be automated and checked in.
- For development infrastructure, it is acceptable if an installation HOWTO is provided in Confluence.

Reviewed

Every task must be peer reviewed. There is no formal process how this should be done, just make sure that some colleague has a look at the changes you made and incorporate feedback as necessary.

For non-visual design issues, make sure [redacted] was involved somehow - he is in charge of keeping the whole system consistent.

For visual design issues, make sure [redacted] was involved somehow - he is in charge of keeping the whole system consistent.

Tested

Make sure that all implemented changes are thoroughly tested to keep overall quality consistently high.

For Java code this means:

- Write unit tests for all non-super-trivial code.
- Make sure that Cobertura reports a solid test coverage (Maven runs Cobertura for you).
- Write integration tests to automatically test the end-to-end scenario described by the user story (involves database etc.).
- Write Selenium-based test to check correct behavior of Web UIs.

Deployed

Code must be deployed on test machines.

Documented

The documentation must allow a reasonably knowledgeable person to understand the implemented changes. To check this level of documentation, reviews should usually be handed over just via a link to the entry point of the documentation in the Jira comment. Examples:

- Documentation: http:/[redacted]confluence/display[redacted]
- Have a look at Javadoc of package `com.[redacted]persistence` to get an overview of the design.

In addition, you have to clarify the licensing issues whenever you use 3rd party products. See LicensingInformation for more information.

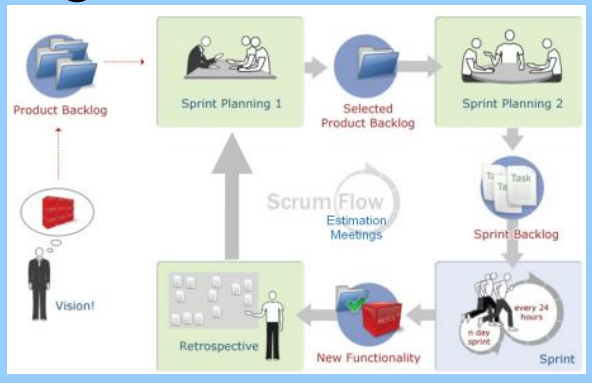
Making the product owner happy

Ask the product owner for satisfaction - preferably before the Sprint review meeting.

Please note that we are not developing for ourselves or even the product owner, but for end customers. Always keep in mind the personas representing them.

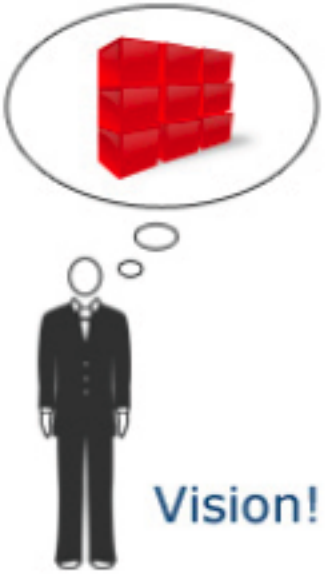
http://www.atlassian.com/software/confluence/

Agenda

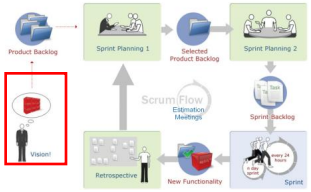
- Introduction
- Scrum Overview
- Scrum Consequences
- Our Example-Project with Ingredients & Definition of Done
- 

- Our Lessons Learned
- Discussion & Questions

Product Vision



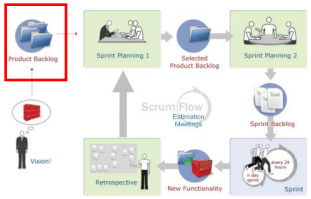
- Product presentation as for investors
- Confluence
- One sentence to carry your vision
- **Inspire the developers**



Source: TNG

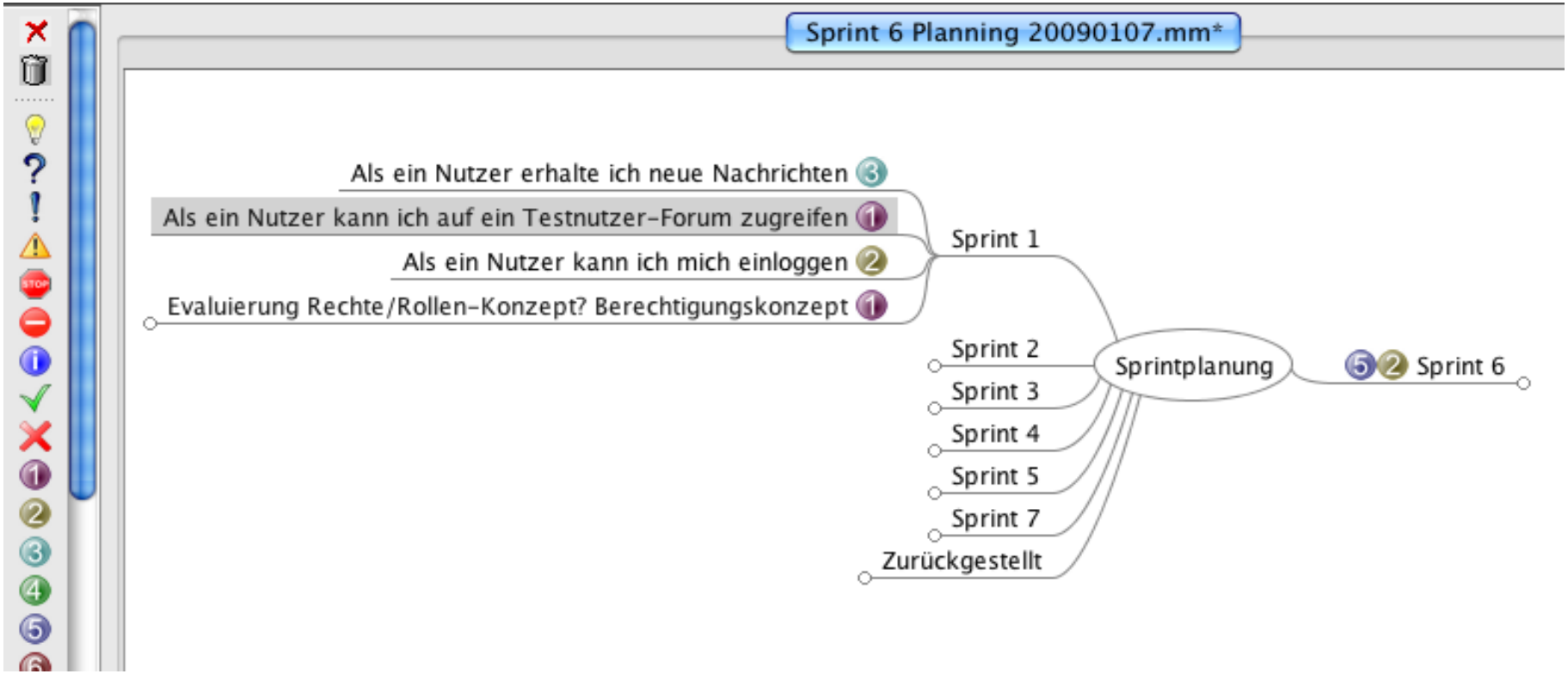
Product Backlog

- Freemind, http://freemind.sourceforge.net
  - Mind map containing crude planning for future sprints
  - Upcoming user stories
- Jira, http://www.atlassian.com/software/jira/
  - Technical Product Backlog
  - Bugs
  - Technical debts (“works for now, but has to be revised”)



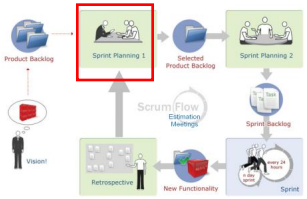
Source: TNG

Product Backlog



Sprint Planning 1

- Freemind “Online“
- Planning Poker
- Team Commitment
- **Result:** “selected product backlog” is stored in Confluence
- **Who**: whole team plus Product Owner, about 3h



Source: TNG

Selected Backlog

Summary

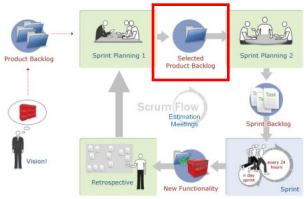
- **Project:** [redacted]
- **Components:** User Story
- **Resolutions:** Unresolved
- **Sorted by:** Priority descending

Operations

- Save

Browser (**Current Fields** | Printable | Full Content) | XML | RSS (Issues | Comments) | Word | Ex

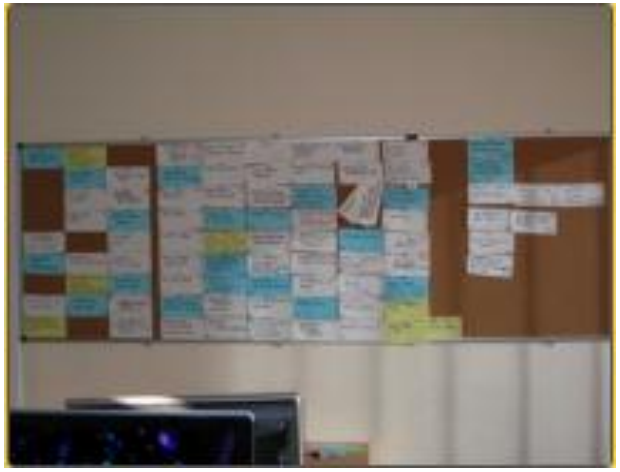
T	Key	Summary
	TT-642	Miscellaneous Tasks Sprint 7
	TT-643	Designer Task Sprint 7
	TT-520	[redacted]
	TT-658	[redacted]
	TT-538	[redacted]
	TT-664	[redacted]
	TT-665	Prepare [redacted] beta version for testing
	TT-666	Finalize installation & upgrade process
	TT-667	Remaining Tasks from Sprints 1 to 6
	TT-644	Fix remaining bugs and tasks from Sprint 6

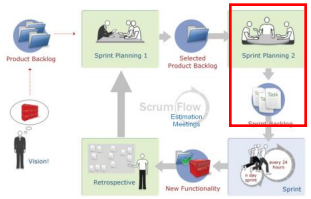


Source: TNG

Sprint Planning 2

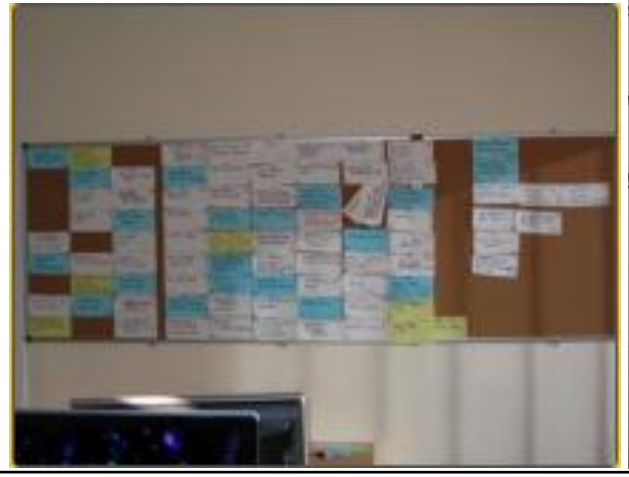
- Breakdown of User Stories into Tasks
- Per User Story one blue card
- Per User Story Card multiple white task cards
- Each User Story is one Jira Issue of type “User Story”
- Each Task for a User Story is a Sub-task in Jira
- **Result**: Colourful task board
- **Who**: whole team, about 2-3h
- Time Effort for Jira: 1-2h (one person)

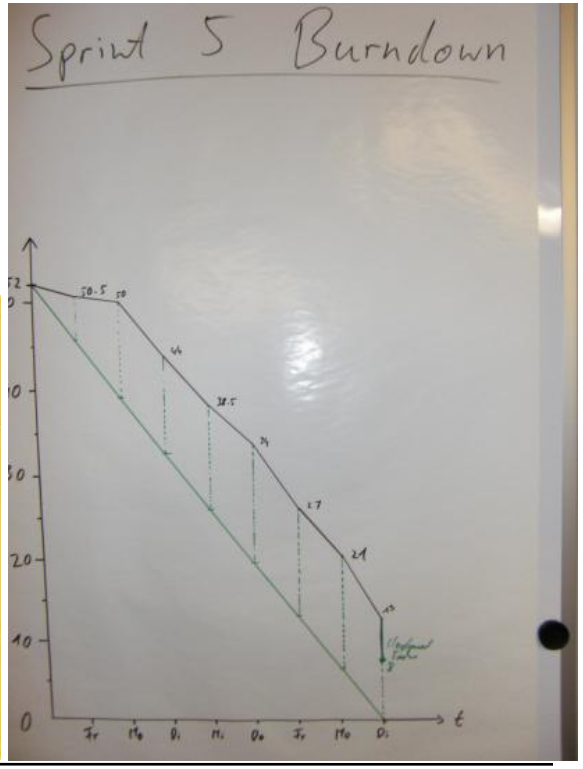


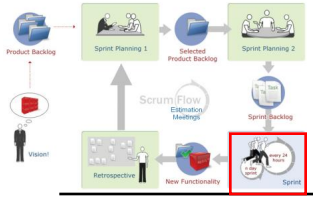


Development

- Daily Scrum
  - Completed are moved to the completed stack
    - Close issue in Jira
  - Take your task
    - Taking a card = „Assign to me“ in Jira
  - Starting a user story → specify person responsible for story
  - Story responsible does final User Story test

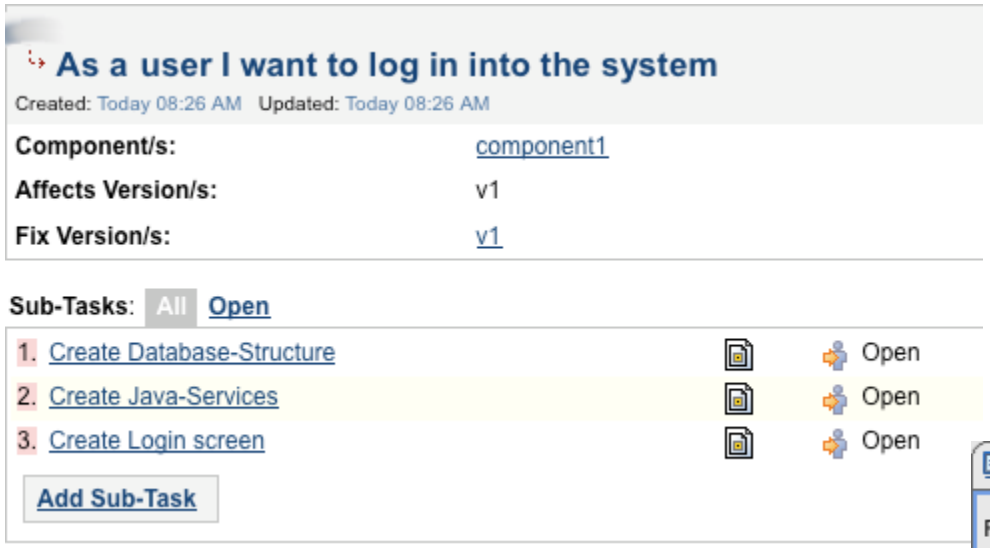




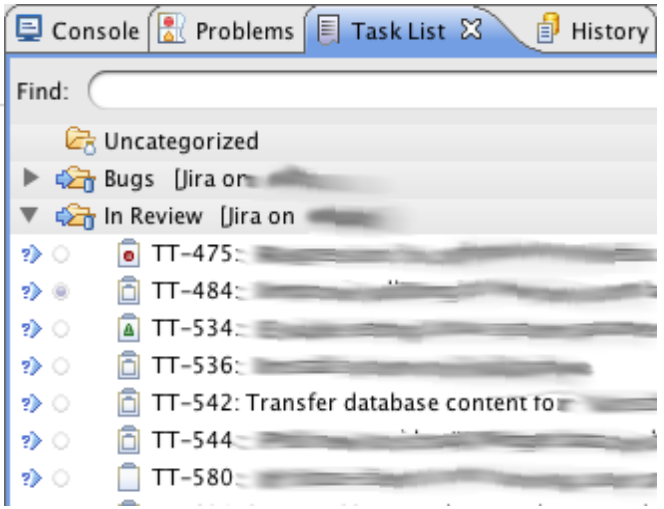


Source: TNG

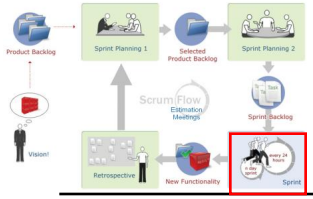
Working with Tasks



Tasks in Jira

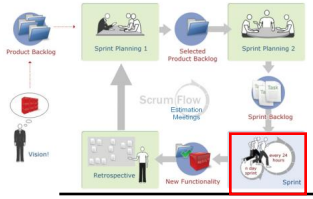


Tasks in Eclipse - Mylyn



Technology Stack (production code, excerpt) (I/II)

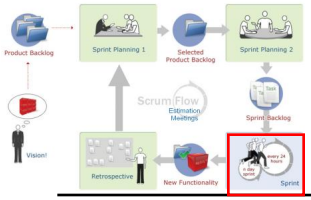
- Java 6
- Spring (incl. autowiring)
- Hibernate (with annotations)
- Apache-commons-*
- XStream
- Tapestry 5
- jQuery
- Joda-Time



Source: TNG

Technology Stack (production code, excerpt) (II/II)

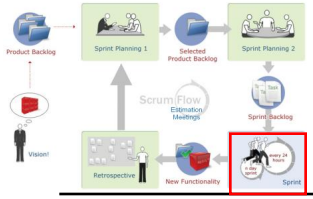
- Jetty & Tomcat
- Log4j / Slf4j
- Quartz
- Buildinfo
- PropertyLoader
- …



Source: TNG

Technology Stack (development code)

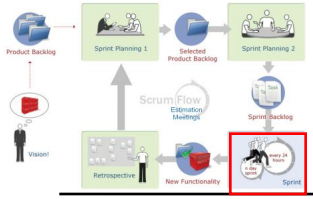
- TestNG
- Mockito
- Selenium RC / Firefox
- Maven2 / multi project
- Javascript Unit Test Library (jQuery)
- Eclipse / IntelliJ
  - Mac OS X & Linux
- AspectJ



Source: TNG

Development Technology (I/VII)

- Its all about feedback loops!
  - Subversion: checkins only possible with JIRA Issue Number
    - Commit acceptance plug-in
      - Hint: UTF-8 and ISO-8859-1 are no default Jira issues ☺
  - Continuous Integration: Hudson, https://hudson.dev.java.net/
    - Continuous Integration
    - Nightly Build
    - External Integration
    - 1-Click Relase Job

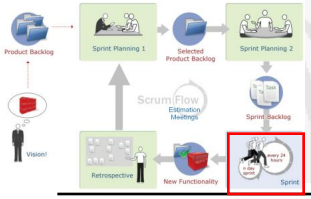


Source: TNG

TNG TECHNOLOGY CONSULTING

Hudson Jobs

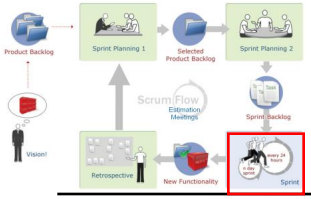
S	W	Job ↓	Last Success	Last Failure	Last Duration
		ContinuousIntegration	12 hr (#486)	14 hr (#478)	3 min 12 sec
		External_Integration	5 hr 31 min (#59)	9 days 23 hr (#45)	2 min 9 sec
		Nightly_Build	5 hr 48 min (#160)	18 hr (#158)	16 min
		Release	8 days 17 hr (#28)	13 days (#25)	29 min



Source: TNG

Development Technology (II/VII)

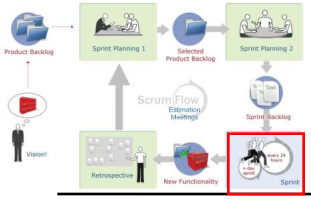
- Its all about feedback loops!
  - Testing: TestNG (categories)
    - Test Types: source code, unit, integration, external, UI, e2e (client/server), AspectJ
    - UI Tests: Selenium RC
  - Built by: maven builds
    - reproducible builds
  - Maven Repository
    - archiva



Source: TNG

Development Technology (III/VII)

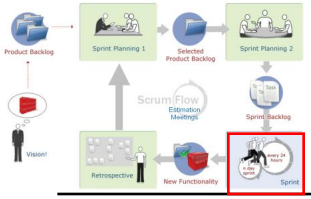
- Jira: explicit peer review, with checklists
  - Task/Story fulfilled?
  - Tests?
  - Code style?
  - Documentation?
  - Configuration?
  - …

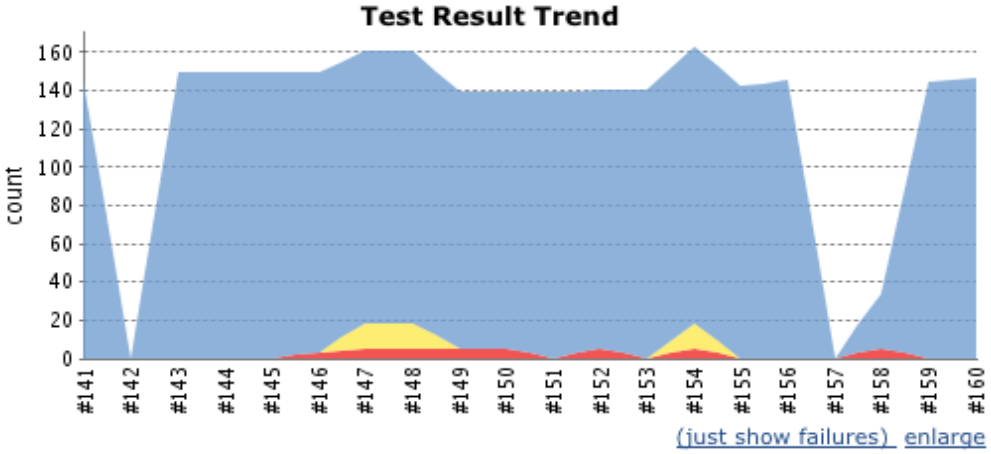


Source: TNG

Development Technology (IV/VII)

- Reporting: Hudson
  - Cobertura, FindBugs, Surefire, Javadoc, PMD, Checkstyle, Emma, VNC/Selenium
  - Emails for broken builds
  - The person breaking the build has to give free cookies to everyone ☺
    - Ice in the summer is a good substitute

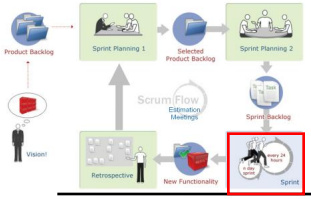




Source: TNG

Development Technology (V/VII)

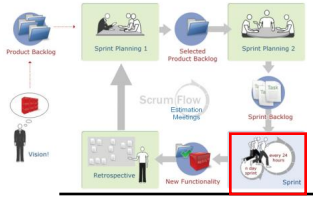
- Documentation: JavaDoc, Confluence
  - Development:
    - Domain model, deployment, architecture, branching, team, guides
  - Install Guide, Admin Guide, User Guide: Confluence
    - Separate Spaces



Source: TNG

Development Technology (VI/VII)

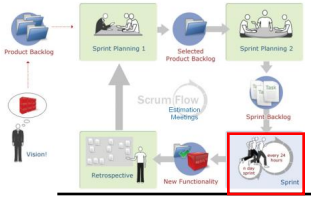
- DB issues
  - DB:
    - SQL migration scripts
    - Test Data builders
      (see http://nat.truemesh.com/archives/000714.html)
    - Meta table
    - VMs for databases with defined content
  - UUIDs to the rescue
    - Merging data is simple
    - Well-known natural ids
    - Own instances per test class



Source: TNG

Development Technology (VII/VII)

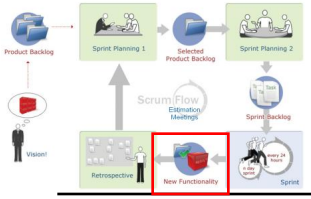
- DB issues
  - Tests in transactions that are rolled back, or cleaned up after the test execution immediately
    - Except for UI tests
  - Creating data
    - Master data
    - Test data
  - Error reporting from live system via detailed exception emails, GPG encrypted if necessary



Source: TNG

Sprint Review

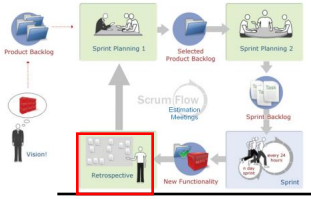
- Version from stage repository (“release build”)
- Deployed on test system
- **Who**: Whole team plus Product Owner and Users
- **Result**:
  - Happy stakeholders
  - Some new product ideas
  - Jira issues
- 1-2 h, not much preparation (no ppt!)



Source: TNG

Retrospective

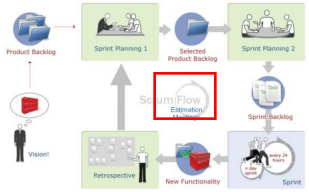
- Team plus Product Owner if invited
  - And invited persons from other departments, if necessary
- About 2 h (at the beginning longer)
- Result is documented in Confluence (pictures, action list)
  - And reviewed at the beginning of the next retrospective
- Action Items are pinned at the wall



Source: TNG

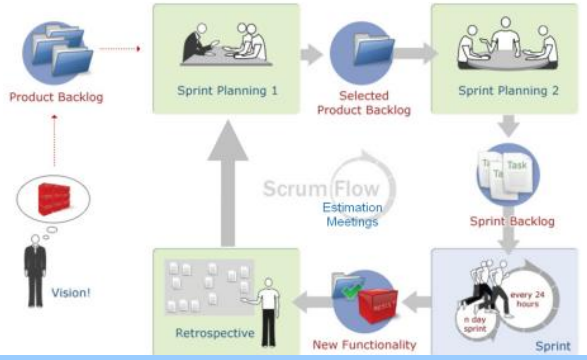
Regular Estimation Meetings

- Once per week
- Different scopes (from whole product backlog to next sprint input)
- Product Owner plus representative team members (architecture, requirements, development, test, design, documentation,…)
- **Who**: Maybe but not necessary: all team members
  - Too expensive? But Team knows what will come…
- **Result**: Rough estimates of possible User Stories



Source: TNG

Agenda

- Introduction
- Scrum Overview
- Scrum Consequences
- Our Example-Project with Ingredients & Definition of Done
- 

- Our Lessons Learned
- Discussion & Questions

Challenges

- Part-time team members
- Heterogeneous project (linux, java, javascript, design, …)
- Illnesses
  - Fall and Winter is bad for health :-(
- Team member drain
- Late visual design (as usual)
- Other external dependencies

Source: TNG

From our last project

- Good Product Owners are rare, care for them :-)
- If you have a big team (up to 13): **Timeboxing!**
- Try to create User Stories of comparable size
- Try to break User Stories into small independent tasks
  - Esp. independent from suppliers
  - Try to isolate depended tasks

What we have learned (generally)

- The Team can start with Scrum even if it's not called Scrum
  - Retrospectives
  - Stand-up meetings
  - Iterations
  - Good development practices
- But the organization will (have to) change, too.
- So management support is essential for full Scrum adoption

Ready for Scrum?

- Scrum is “Extremely simple but very hard”
- “Real Time Process Improvement”
- “Extreme Accountability”

Further Topics

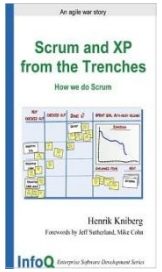
- Scrum Coaching, e.g.:
  - Conduction of Retrospectives
  - Agile Estimation and Planning
- Agile Development Practices, e.g.:
  - Test Driven Development, Continuous Integration, Build Management
  - Automation (Test, Environment setup, Rollout, …)
- Reviews (development processes, architecture, software)
- Software Development
  - Especially Java! (JEE, Spring, Hibernate, TestNG, OSGi, Eclipse RCP, Maven2, Hudson, JavaScript, …)

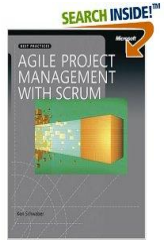
Further information about Scrum

- Scrum Alliance
  - http://www.scrumalliance.org
- Scrum Community, e.g. @ Yahoo Groups
  - http://groups.yahoo.com/group/scrumdevelopment/
  - http://groups.yahoo.com/group/deutschescrum/
- Scrum from an Organization Perspective
  - http://seattlescrum.files.wordpress.com/2007/08/scrum-from-an-organizational-perspective.ppt
- Scrum Introduction from Henrik Kniberg
  - http://www.crisp.se/henrik.kniberg/presentations/Scrum-Intro-Henrik-Kniberg.pptx

Books about Scrum

- Scrum. Produkte zuverlässig und schnell entwickeln
  - ISBN 978-3446414952 (Boris Gloger)
- Scrum - Agiles Projektmanagement erfolgreich einsetzen
  - ISBN 978-3898644785 (Roman Pichler)
- Scrum and XP from the Trenches
  - ISBN 978-1430322641 (Henrik Kniberg)
  - Free PDF: http://www.infoq.com/minibooks/scrum-xp-from-the-trenches
- Agile Project Management with Scrum
  - ISBN 978-0735619937 (Ken Schwaber)
- Agile Software Development with Scrum
  - ISBN 978-0130676344 (Mike Beedle, Ken Schwaber)
- The Art of Agile Development (James Shore, Shane Warden)
  - ISBN 0-596-52767-5





TNG TECHNOLOGY CONSULTING

Books about Scrum

- The Enterprise and Scrum
  - Microsoft Press, 2007, Ken Schwaber
- Scaling Software Agility: Best Practices for Large Enterprises
  - Addison-Wesley, 2007, Dean Leffingwell
- Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum
  - Craig Larman, Bas Vodde, 2008, Addison-Wesley
- Agile Adoption Patterns: A Roadmap to Organizational Success
  - Amr Elssamadisy, 2008, Addison-Wesley
- Scaling Scrum Colin Bird & Rachel Davies, Scrum Gathering London 2007
  - http://www.scrumalliance.org/resource_download/287