.msg .consulting .solutions .partnership

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Agile with Scrum

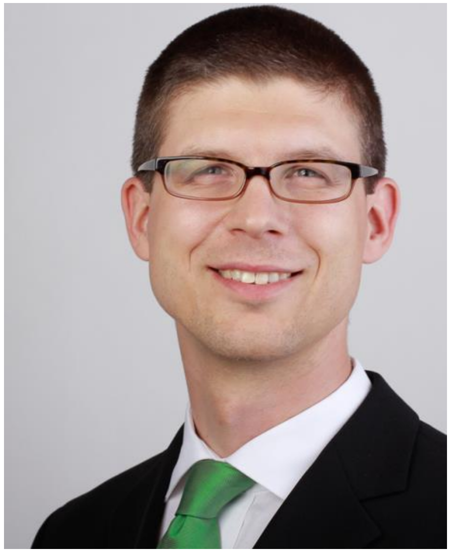
IT Solutions with Software Engineering in Practice

AGENDA

1. Agile Idea
2. Scrum: Roles and Overview
3. Scrum: Adding a Product Feature
4. Scrum: Handling a Change Request
5. Is Agile better than Classic Project Management?
6. Q&A

Alexander Schwartz

- Studied management science (Betriebswirtschaftslehre) at Philipps-Universität Marburg (DE) and University of Kent (UK)
- 15+ years of experience in web development
- Certified Scrum Master since 2010
- Today: Principal IT Consultant at msg systems ag
- Office in Eschborn
- Focus on Java and web technology
- Interests: agile project management, open source, automated tests
- Responsibilities: architecture, technology scouting



Private:

- 39 years old, married, 2 kids
- Hobbies: Family, Geocaching

AGENDA

1. Agile Idea
2. Scrum: Roles and Overview
3. Scrum: Adding a Product Feature
4. Scrum: Handling a Change Request
5. Is Agile better than Classic Project Management?
6. Q&A

Experiences with Classic Project Management

Why you might want to be agile if you used Waterfall Model:

- Scope changes often – too much time spent on re-planning
- Deploy early in order to generate revenue with new features
- Not enough people involved in planning / only one Project Manager does all planning
- Late Feedback on how the system will look and behave
- Lack of transparency for
  - costs
  - time
  - progress
- Motivate developers & business
- Recruit talents
- Make developer feel responsible for their product

Experiences with Classic Project Management

**Why you might want to be agile if
you didn’t use proper Project Management:**

- Projects are late and expensive, and you don’t know why
- Looking for a method that is easy to implement

- People complain that they don’t have enough information what is going on (both managers and developers)
- Release dates are moved / scope varies
- Customers of your company don’t know when to expect new features

Agile Myths

- No processes
- No documentation
- Always works
- Easy to implement
- You need software tools can make it work

- No Plan
- It’s not better than classic PM / waterfall

Agile Manifest

More important	Less important
individuals and interactions	processes and tools
working software	comprehensive documentation
customer collaboration	contract negotiation
responding to change	following a plan

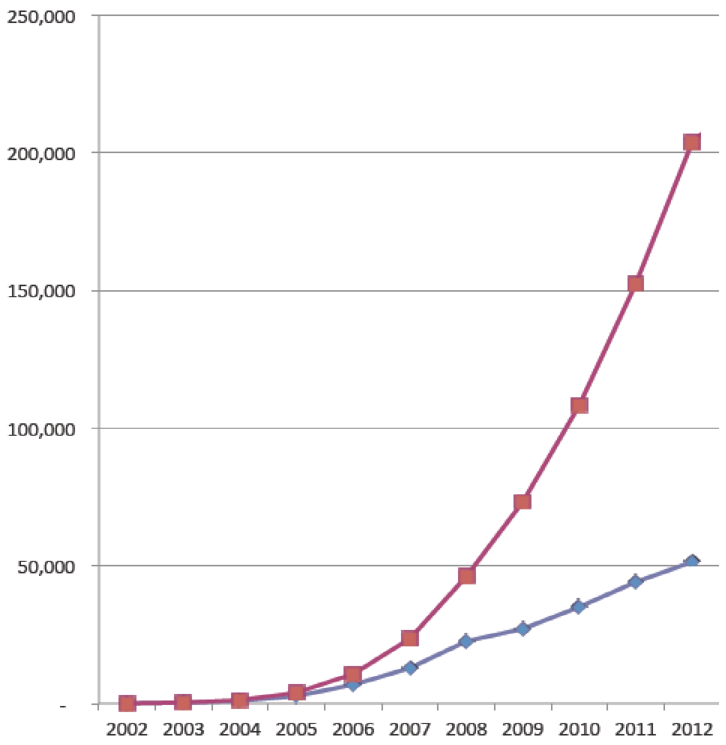
1. http://agilemanifesto.org/

Scrum

Gained momentum from 2002 with certification available from Scrum Alliance

Key concepts

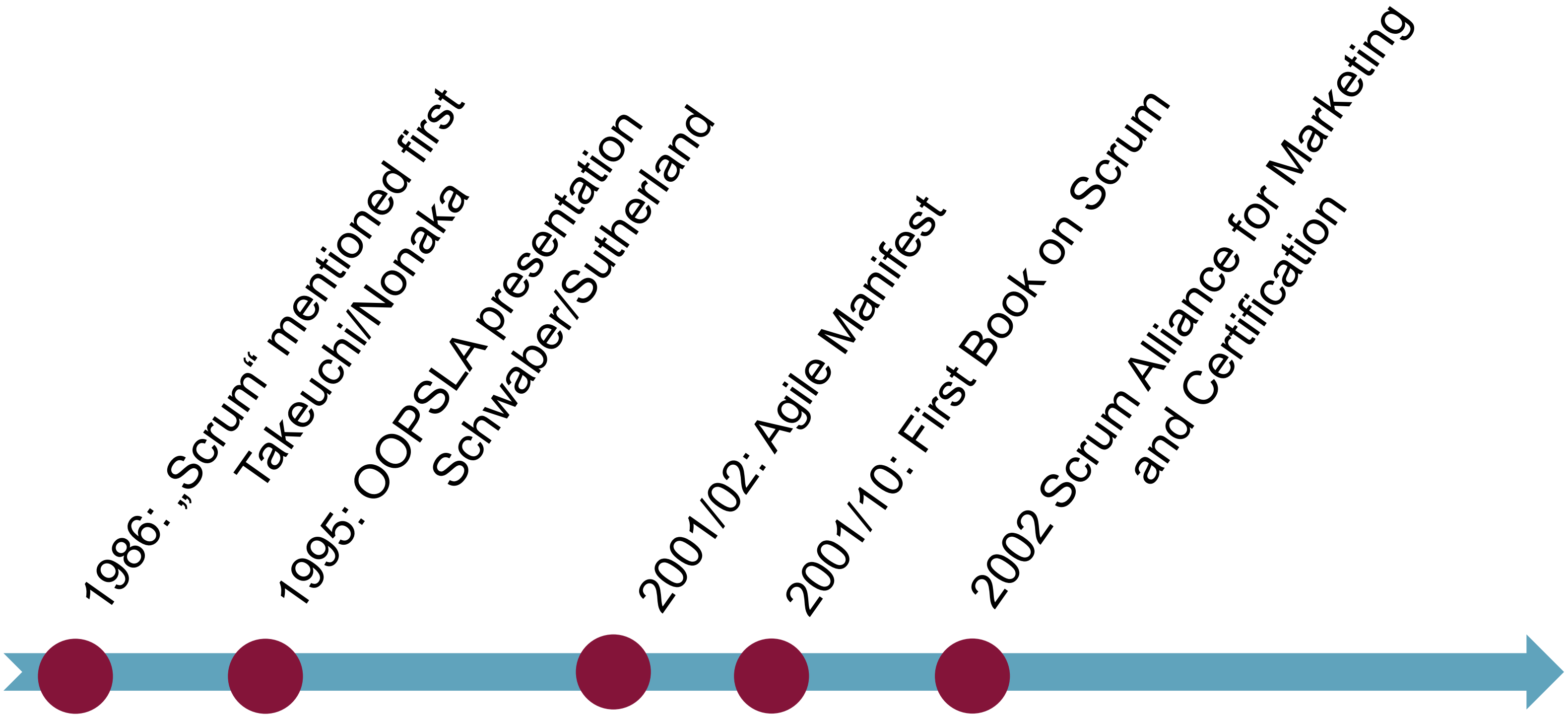
- small team (5-7 people)
- development of a software product
- well-defined roles
- aligned (small) set of key practices
- built around the iteration cycle



red: Cumulated number of certified scrum masters
blue: New scrum master certifications in a year

1. http://www.scrumalliance.org/resources/2505

Scrum Evolution



1986: „Scrum“ mentioned first Takeuchi/Nonaka

1995: OOPSLA presentation Schwaber/Sutherland

2001/02: Agile Manifest

2001/10: First Book on Scrum

2002 Scrum Alliance for Marketing and Certification

1. http://www.jeffsutherland.org/oopsla/oo95summary.html
2. http://www.techwell.com/2012/10/brief-history-scrum

Extreme Programming (XP)

started around 1997, first book 1999 (Kent Beck)

4 Values

Communication, Simplicity, Feedback, and Courage

12 Practices

Pair Programming, Collective Code Ownership, Continuous Integration, Automated Tests, Whole Team, Refactoring, Sustainable Pace,
Short Iterations, System Metaphor, Coding Standards, Simple Design, Planning Game

1. http://en.wikipedia.org/wiki/Extreme_programming
2. http://en.wikipedia.org/wiki/Chrysler_Comprehensive_Compensation_System
3. http://www.martinfowler.com/bliki/C3.html

Kanban Software Development

Based ideas of “Lean” of Toyota Production System (1970s)

Applied in 2003 to Software Development

Formalized in Kanban Software Development after 2007

Key concepts

- Continuous Flow
- Minimize Work in Progress
- Pull principle
- Stop the line
- Kaizen

1. http://www.infoq.com/articles/hiranabe-lean-agile-kanban
2. http://en.wikipedia.org/wiki/Muda_%28Japanese_term%29
3. http://en.wikipedia.org/wiki/Toyota_Production_System
4. http://en.wikipedia.org/wiki/Kanban
5. http://www.agileproductdesign.com/blog/2009/kanban_over_simplified.html

AGENDA

1. Agile Idea
2. **Scrum: Roles and Overview**
3. Scrum: Adding a Product Feature
4. Scrum: Handling a Change Request
5. Is Agile better than Classic Project Management?
6. Q&A

Roles in Scrum Projects

Product Owner:

- Knows the requirements
- Responsibility, that needed functionality is delivered at the right time

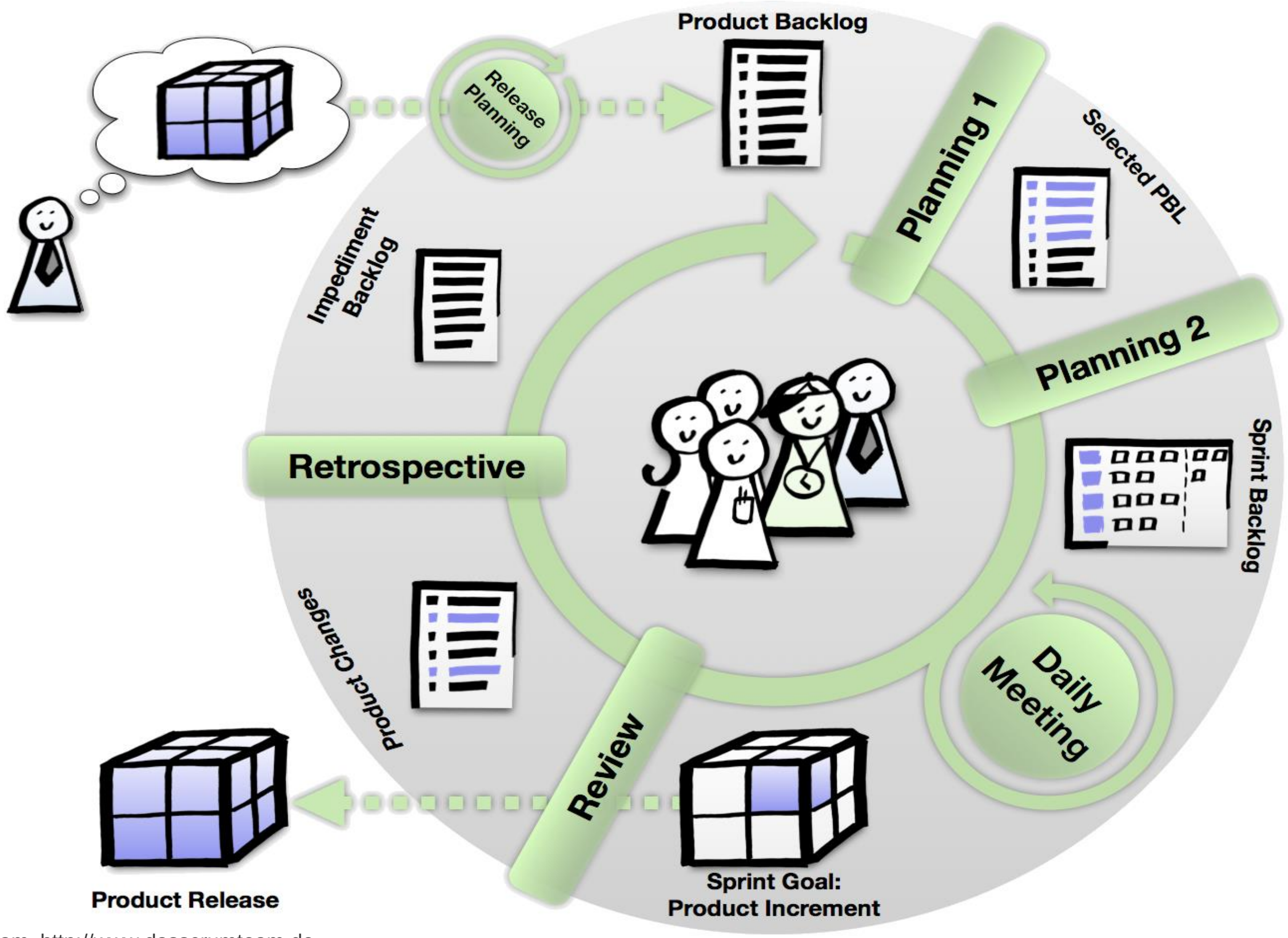
Scrum Team:

- Creates the final product (includes developers, testers, database administrators, etc.)

Scrum Master:

- Ensures that the working methods run smoothly
- Ensures that the methods are followed
- Tackles obstacles that arise outside the team

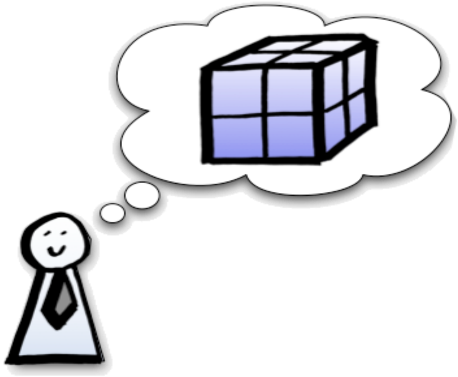
Process in Scrum Projects



1. DasScrumTeam, http://www.dasscrumteam.de

Product Owner

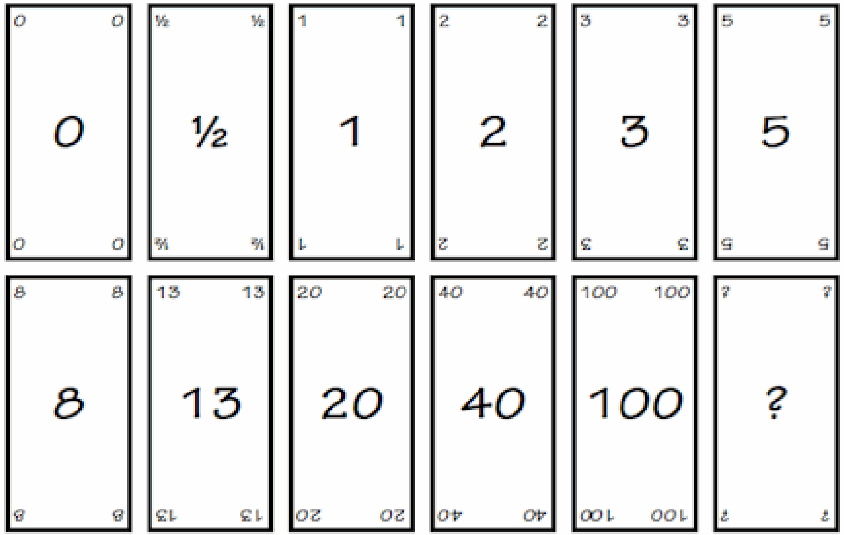
- Has a vision about how the final product should look like
- Secures the budget
- Slices the vision down smaller packages
  (user stories or epics)
- Knows when she/he needs it
- Can explain the requirements
- Can prioritize requirements
- Maximizes return on investment



Release Planning

- Done by the Scrum Team together with Product Owner
- All known user stories / epics of the software are estimated
- Items that will be developed in the next few iterations will be broken down to smaller bits to fit an iteration

Estimation of effort: Planning Poker



What is Planning Poker?

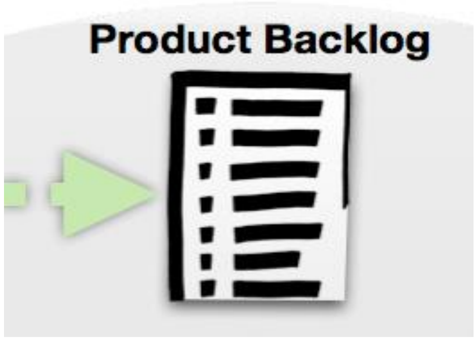
- Estimation by all implementing team members (comparable to Delphi Method)
- Evaluation is done in story points
- Each participant has a set of cards

Steps of Planning Poker

- Moderated by Scrum Master
- For each user story to be estimated:
  - Product Owner presents user story
  - Team asks Product Owner questions about it until it feels confident to estimate the user story
  - Each team member estimates the task and puts a card upside down on the desk
  - All cards are turned to show the estimations at the same time
  - Highest and lowest estimation are discussed in the group
  - Estimation is repeated up to two times or when the estimations match
- The meeting lasts for maximum of one hour

Product Backlog

- Product Owner orders the user stories and epics by their importance / return on investment
- Identification of possible release milestones



Planning 1

- Product owner and team meet at the beginning of the sprint
- Team selects as many items from the top of the backlog as they think they can finish within the sprint
- Team gives commitment to deliver at the end of the sprint

Planning 1
Selected PBL

Planning 2

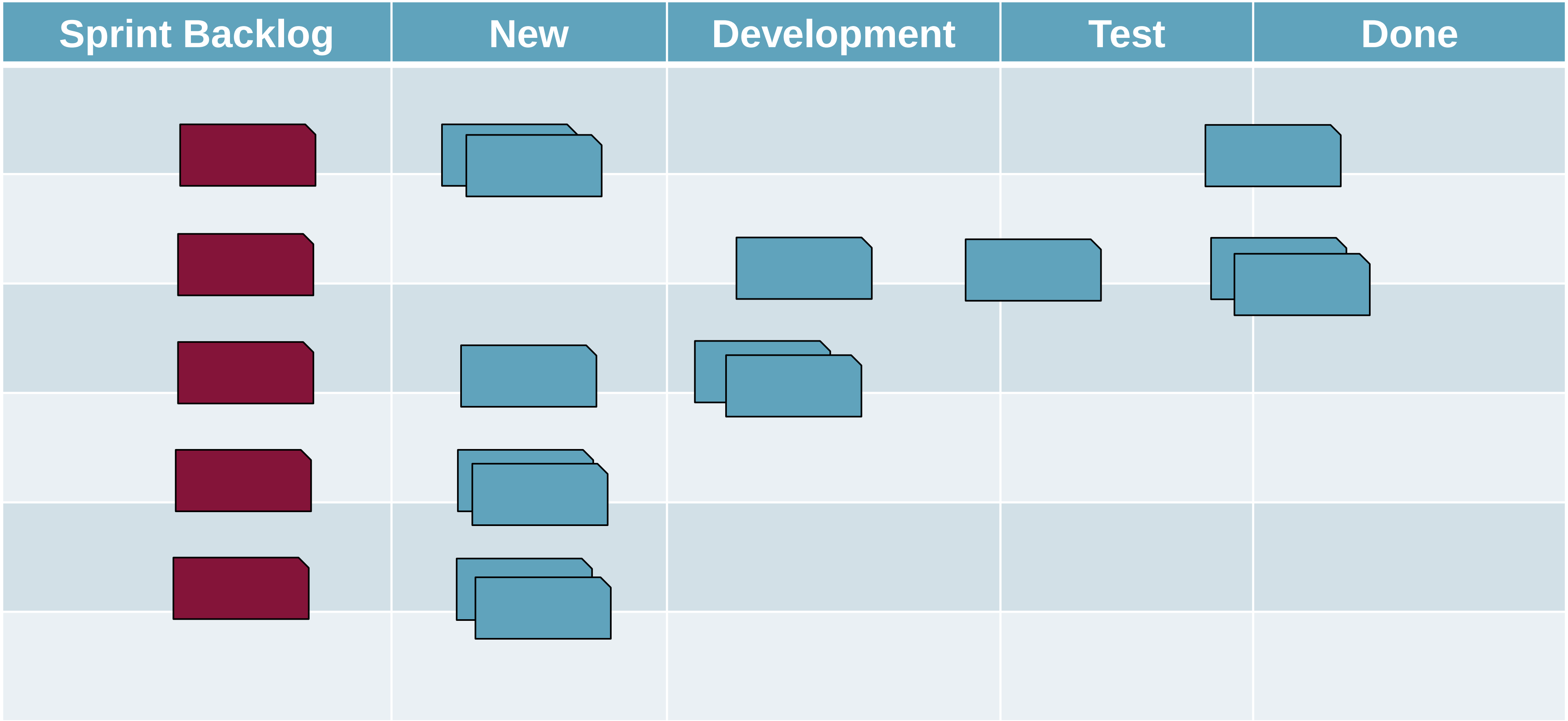
- Team divides user stories to tasks
  (bites that one person can chew in one day)
- Developers pick their first tasks on the task board

Planning 2
Sprint Backlog

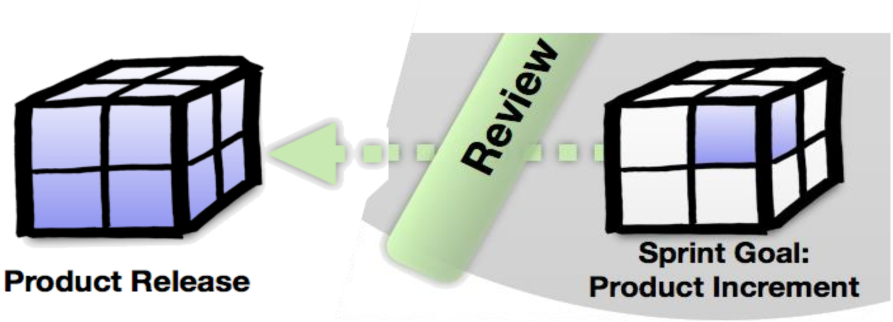
Daily Meeting / Daily Stand-up

- 15 minutes for the whole team
- Three Questions answered by each team member:
  What you have done during the last day?
  What you plan to do today?
  What is your current obstacle?
- Move the tasks together on the board

Scrum Task Board (sample)



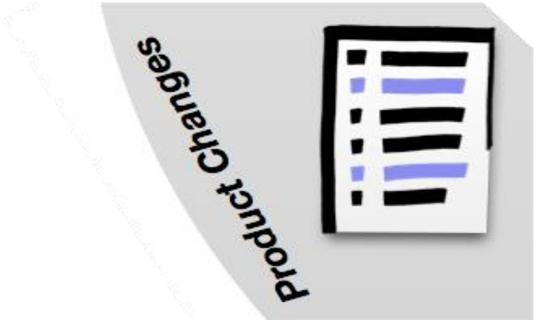
Review & Product Increment



- Product increment finished at the end of the sprint
- At least the most important stories are now complete
- Delivered as fully working product
- Review together with the product owner and maybe other stake holders
- Possible product release

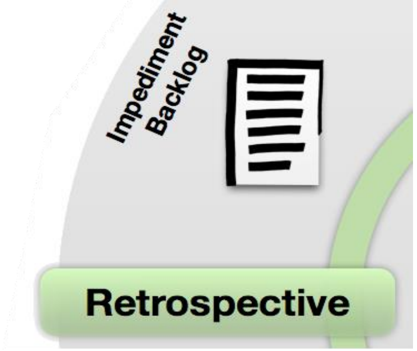
Product Changes

- Product changes are added to the product backlog

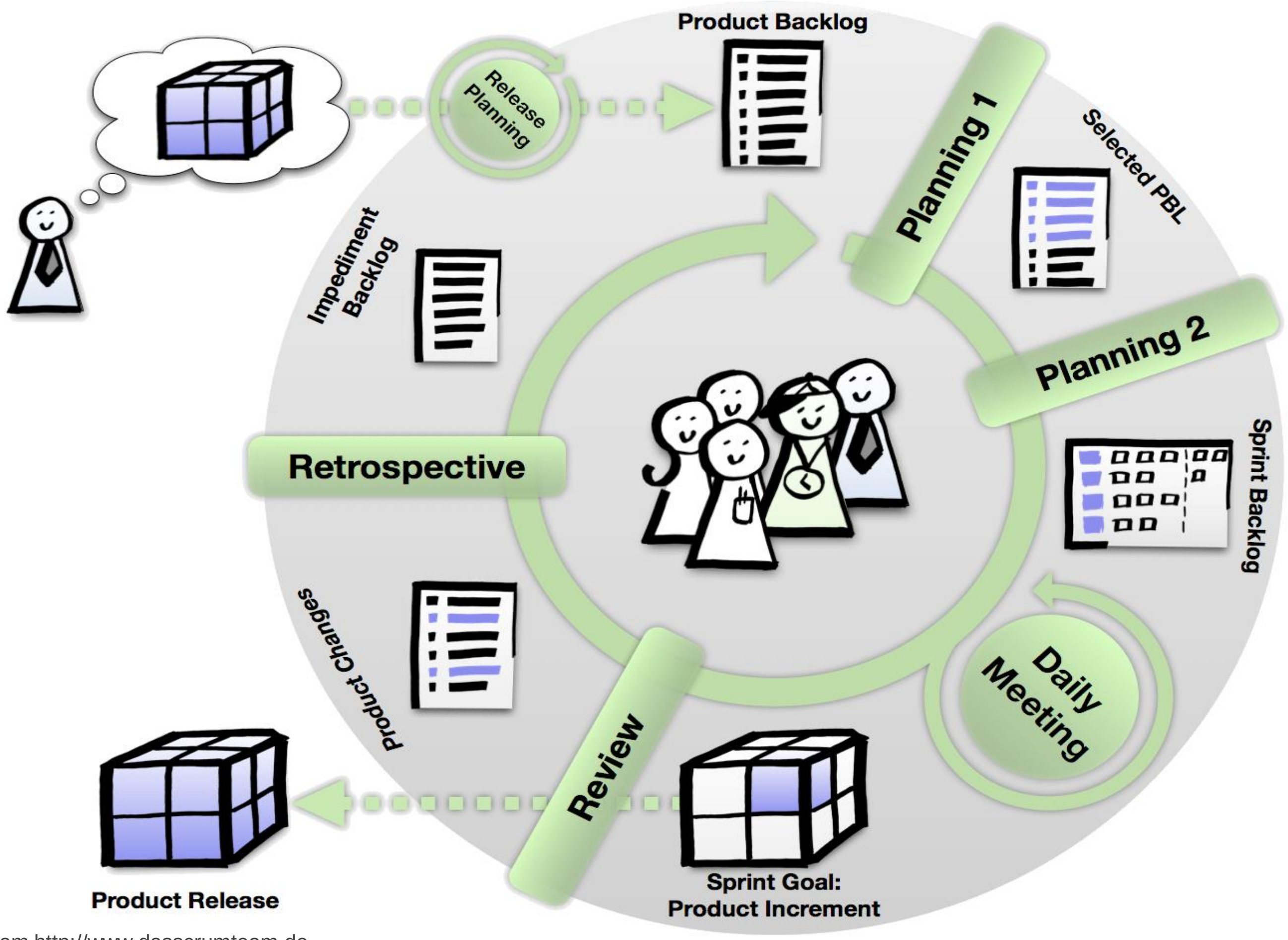


Retrospective und Impediment Backlog

- Reflection as a team to find out things that worked well and that need adaption
- Result: prioritized Impediment Backlog
- Scrum Master will handle Impediments outside the team, impediments inside the team are handled by the team



Process in Scrum Projects



1. DasScrumTeam http://www.dasscrumteam.de

Agile Myths – revisited I

- No processes **– BUSTED!**
- No documentation **– FIRST DOUBTS!**
- Always works
- Easy to implement
- You need software tools can make it work **– BUSTED!**

- No Plan **– BUSTED!**
- It’s not better than classic PM / waterfall

Invitation to an unique Scrum experience

Experience all of Scrum in just one day

- Participate in a hands on training
- Pick up a the necessary Scrum fundamentals on the go
- Build a Lego town and use Scrum to do it as a team

Next Scrum training

- November 13th 2015 @ msg systems ag, Ismaning/Munich

Next Kanban training

- October 16th 2015 @ msg systems ag, Ismaning/Munich

Registration

http://www.msg-systems.com/scrum
http://www.msg-systems.com/kanban

AGENDA

1. Agile Idea
2. Scrum: Roles and Overview
3. Scrum: Adding a Product Feature
4. Scrum: Handling a Change Request
5. Is Agile better than Classic Project Management?
6. Q&A

New requirements: An Idea

Where it starts: Business department has an idea

Sample:
„You should be able to change the address over there "

Documented as User Story:
„A logged in user should be able to change the address of the customer. This way statements, letters and cards will be sent to the new address."

(roles, functionality and motivation are now clear)

INVEST / independent, negotiable, valuable, estimatable, specific, testable

Priorities of requirements

- Is it a legal requirement?
- How much effort (Project/Change Request)?
- What is the business value?
- Ordering of projects and change requests on department and company level
- If there no result can be negotiated, automatic ordering is the last resort

Amount of documentation: one line in an Excel table

Detailing Requirements

- We will need to implement a new use case
- Analysing together with business department what needs to be changed
- Challenge: contradicting requirements, negotiation between departments, not all consequences have been considered by department
- Developers need a first summary to know what to develop
- Department may be difficult to contact during development

- Work of a business analyst (Product Owner?)

Detailing Requirements

The following documents have proven helpful in the past:

- Use Case Documentation
- Mock-up Screens
- Activity Diagram (when there is more than one screen)

Pro:

- Documents will be understood by business and developers

Contra:

- Costs?

Estimation with Planning Poker

- A first estimate might have been done by a senior developer, but not by the team
- User Stories might be split in smaller parts to make better estimates and controlled development
- Differentiation between stories with and without specification

Required Documents: User Stories

More information: asking Product Owner or those who helped writing the Mock-ups and Use Case

Result: Effort Points per User Story, Acceptance Criteria

Definition of Ready

- Quality gate before actual development starts
- Defined during the first iterations by the team
- Ensure that enough information is available to be able to complete the story in one sprint

Sprint Planning I

- Vision for the next Sprint (2 weeks)
- Presentation by Product Owner of prioritized Back Log
- Team selects the stories that can be implemented in the Sprint (taking into consideration skills)
- Commitment of the team to deliver these Stories according to the „Definition of Done“

Documents: prioritized and estimated User Stories

Result: Selected Product Backlog

Sprint Planning II

- Dividing a Story into Tasks (max. one developer day)

Documents: User Stories in Selected Product Backlog

Result: Tasks

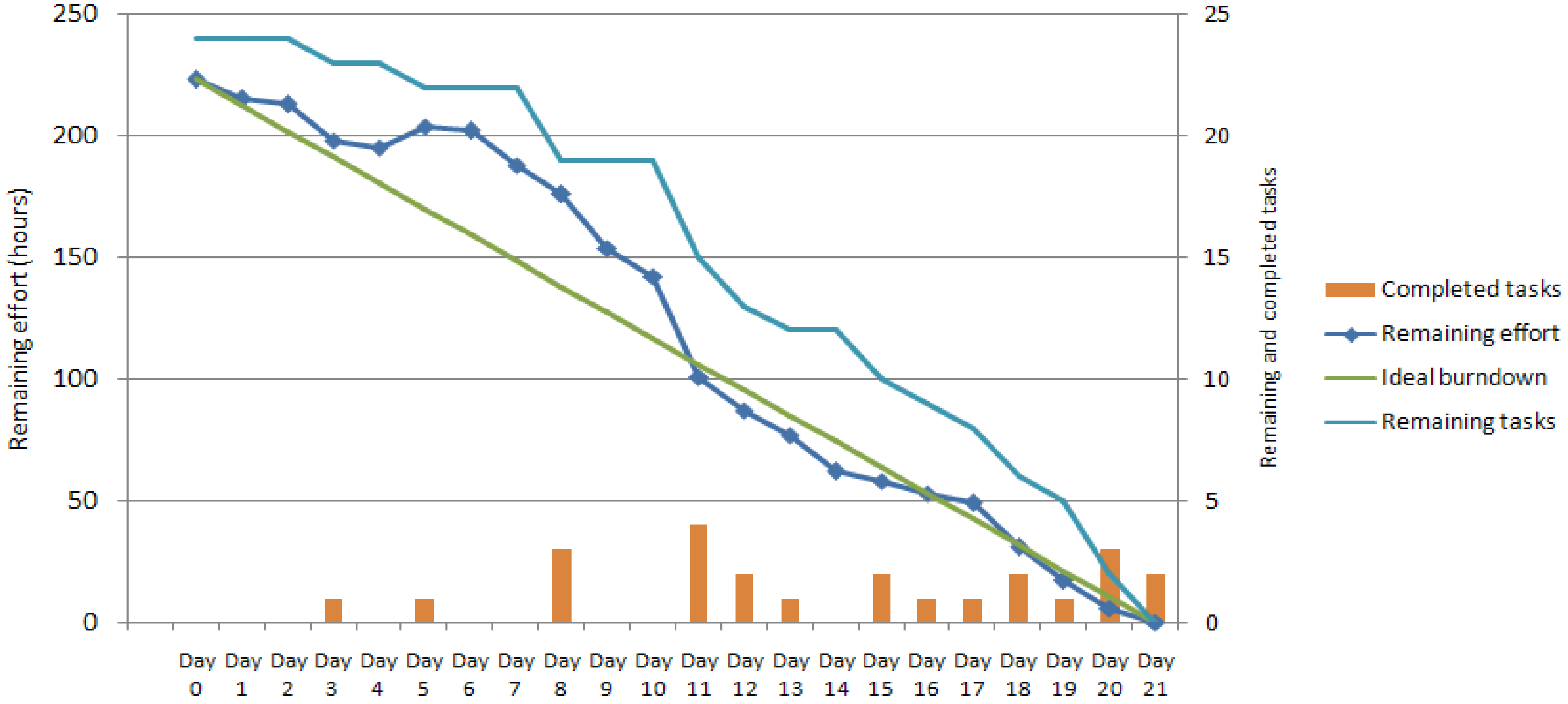
Daily Meeting / Daily Stand-up

- 15 minutes meeting
- What you have been working on, what is ready, problems, next steps

Documents: User Stories in Selected Product Backlog, Tasks

Result: Sprint Burn Down Chart, Impediments

Sample Burndown Chart



1. http://en.wikipedia.org/wiki/Burn_down_chart

Work in the Team

- Implementing the Frontend
- Implementing Web Services
- Database Schema Changes
- Unit Tests (Junit, DBUnit, etc.)
- Frontend Tests (Selenium, Jasmine, etc.)
- Co-ordination with business / business analyst
- Module User Acceptance Test
- Test Engineer

⇒ Ideally cross-functional team // no external dependencies to get the work completed

Documents: User Stories in Selected Product Backlog, Tasks, Mock-Ups, Use Cases, Activity Diagram, Acceptance Criteria

Result: Application, Developer Documentation

Definition of Done

- Specified by business with help of business analyst (checked by team member)
- Analysed by developer (checked by second team member)
- Developed, automated test, documented and committed by developer (reviewed and test by second team member)
- Automated deployment to test environment
- Module user acceptance test (approval by business)

(second quality gate, see definition of ready for first quality gate)

Documents: User Stories in Selected Product Backlog, Tasks, Mock-Ups, Use Cases, Activity Diagram, Acceptance Criteria

Result: Application, Developer Documentation

Sprint Review

- Implemented Stories presented to Product Owner and Stakeholders

Documents: User Stories in Selected Product Backlog, Mock-ups, Use Cases, Activity Diagram, Acceptance Criteria

Sprint Retrospective

- Team reviews development process and discusses and decides about changes

Documents: (Impediment Backlog)

AGENDA

1. Agile Idea
2. Scrum: Roles and Overview
3. Scrum: Adding a Product Feature
4. Scrum: Handling a Change Request
5. Is Agile better than Classic Project Management?
6. Q&A

Someone asks for a change

- Someone asks for added functionality: that change of an address should done with a date in the future
- This is someone else who is not the orderer of the original feature

Documents: Use Case, Mock-up Screen, Activity Diagram

Same process…

… documentation is being updated, kept up to date

Other Documents you might need

- „Getting Started“ und „Development Cycle“ for new developers
- JavaDoc – ensured i.e. via CheckStyle
  - How much JavaDoc do you need?
- Documentation of single technical components
- Architecture Documentation (arc42 as a template)
- Product presentation (Product Box)
- Checklists Code Review
- Role and Permission concept of the application

⇒ Can you do with less?

⇒ Write down only what others will read!

Agile Myths – revisited II

- No processes – BUSTED!
- No documentation **– BUSTED!**
- Always works **– BUSTED!**
- Easy to implement **– BUSTED!**
- You need software tools can make it work – BUSTED!

- No Plan – BUSTED!
- It’s not better than classic PM / waterfall

AGENDA

1. Agile Idea
2. Scrum: Roles and Overview
3. Scrum: Adding a Product Feature
4. Scrum: Handling a Change Request
5. Is Agile better than Classic Project Management?
6. Q&A

Experiences with Classic Project Management

Why you might want to be agile if you used Waterfall Model:

- Scope changes often – too much time spent on re-planning
- Deploy early in order to generate revenue with new features
- Not enough people involved in planning / only one Project Manager does all planning
- Late Feedback on how the system will look and behave
- Lack of transparency for
  - costs
  - time
  - progress
- Motivate developers & business
- Recruit talents
- Make developer feel responsible for their product

Experiences with Classic Project Management

**Why you might want to be agile if
you didn’t use proper Project Management:**

- Projects are late and expensive, and you don’t know why
- Looking for a method that is easy to implement

- People complain that they don’t have enough information what is going on (both managers and developers)
- Release dates are moved / scope varies
- Customers of your company don’t know when to expect new features

Obstacles for agile teams

- No access to product owner or similar person
- No single sponsor
- Missing cross-functional and skilled team
- Technology doesn’t support continuous integration and automated tests
- Team members not full time on the project
- Transparency and inspect & adopt are not compatible with organisation’s culture
- You’re not allowed to fail
- Not enough urgency / complexity / novelty
- Large scope, very few releases
- Large teams, multiple geographies, different time zones
- High Visibility already in early phases of the project

Agile projects have a better chance to succeed

- CHAOS Report 2012: “agile projects succeed three times more often than non-agile projects” (are the non-agile projects doing any project management at all?)

- oose PM study:
  40% of classic PM projects are successful
  65% of agile PM projects are successful
  (this gives more details what techniques support successful projects)

1. http://www.oose.de/nuetzliches/fachliches/pm-studie/
2. http://www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall
3. http://www.guerrillaprojectmanagement.com/the-chaos-report-myth-busters
4. http://davidfrico.com/rico-apm-roi.pdf

AGENDA

1. Agile Idea
2. Scrum: Roles and Overview
3. Scrum: Adding a Product Feature
4. Scrum: Handling a Change Request
5. Is Agile better than Classic Project Management?
6. Q&A

Resources (not only) on Scrum

The Scrum Guide
Ken Schwaber, Jeff Sutherland (2011)
http://www.scrum.org/Scrum-Guides

Scrum – Agiles Projektmanagement erfolgreich einsetzen
Roman Pichler (2007)

Product Owner Manual
Scrumsense (2010)
http://www.scrumsense.com/resources/product-owner-manual/

Do Better Scrum
Peter Hundermark (2009)
http://www.scrumsense.com/resources/do-better-scrum/

Gezielte Wahl / Agil oder klassisch – Hinweise zur Methodenwahl
Judith Andresen (iX 3/2013, pp. 50-55)

Agile Retrospectives – Making Good Teams Great
Esther Derby, Diana Larsen (2006)

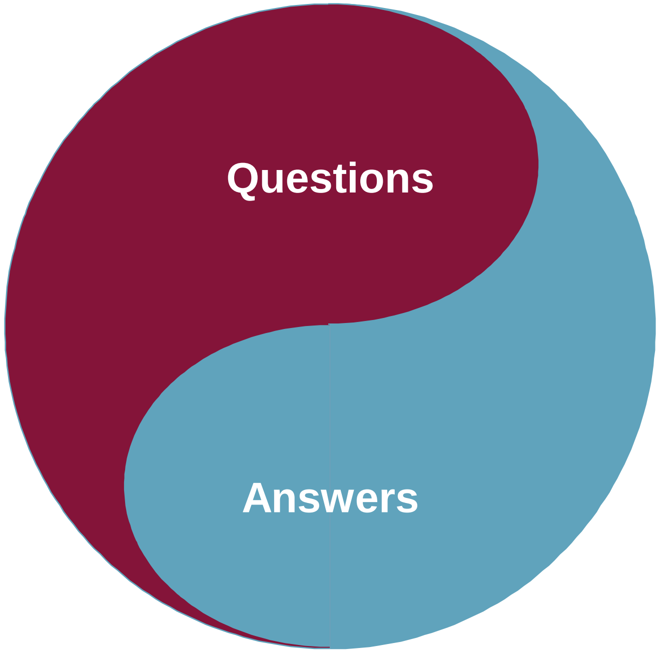
Early Books on Agile Software Development:

Extreme Programming Explained: Embrace Change
Kent Beck (1999)

Agile Software Development
Alistair Cockburn (2002)

Extreme Programming Refactored: The Case Against XP
Matt Stephens, Dough Rosenberg (2003)

Q&A



Alexander Schwartz
GB L / Principal IT Consultant

+49 171 5625767
alexander.schwartz@msg-systems.com

msg systems ag
Mergenthalerallee 73-75
65760 Eschborn

www.msg-systems.com

.msg .consulting .solutions .partnership