Agile Software Engineering

An introduction

… a little bit about XP and Scrum

Uwe Gühl

19 January 2012

Contents

- Introduction – Agile Approaches
- Planning Game XP (Extreme Programming)
- XP – 12 Practices
- Scrum
- Comparison XP and Scrum
- Sources / More

Introduction

„XP is the most important movement in our field today. I predict that it will be as essential to the present generation as the S.E.I. and its Capability Maturity Model were to the last.“

-- Tom DeMarco, *Preface to „Planning Extreme Programming“*, 2001

Introduction

Agile software development [Wik12]
is a group of software development methodologies

- based on iterative and incremental development,
- where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.

Idea: Lightweight software development methods – instead of heavyweight methods

Introduction – Agile Manifesto

[Agi01]

Individuals and interactions	over	processes and tools
Working software	over	comprehensive documentation
Customer collaboration	over	contract negotiation
Responding to change	over	following a plan

Introduction
Agile Approaches

- Extreme programming (XP)
  - software development methodology which is intended to improve software quality and responsiveness to changing customer requirements
  - 12 core practices
- Scrum
  - an iterative, incremental framework for project management
- Others: Crystal Clear, Feature Driven Development, Dynamic Systems Development Method, ...

invented by Kent Back [Bec99]

Paper by Sutherland and Schwaber, presented on OOPSLA 1995

Planning Game – XP

- Learning the Planning Game [Ber01]
  An Extreme Exercise by
  Joseph Bergin, Pace University
  - Email: jbergin@pace.edu
  - Web: http://csis.pace.edu/~bergin

- Idea: Understanding how it works with "User stories" respectively "Story cards"
  ⇨ Promise to communicate

Planning Game – XP

- Teams and Roles
  7 to 10 people in one team
  - 2 Customer – responsible for specification
    - Write and explain features
    - Choose features to realize
  - 2 Monitors – responsible for process
    - Keep everyone honest and communicating
    - Measuring time and looking for deadline
  - 3 to 6 Developers – responsible for built
    - Estimating time for features
    - Developing the product with realization of features

Planning Game – XP

- User stories / Storycards
  - User Stories to handle requirements in XP
  - During the project the customer
    - writes new user stories,
    - deletes old,
    - changes,
    - has to prioritize
  - Small piece of paper
  - Basic for reports
  - Template:
    "As a <role>, I want <desire> so that <benefit>"

User Story Template

As a <type of user>
I want <some goal>
so that <some reason>

Planning Game – XP

- Part 1: Writing and Estimating Stories

Iteration 1

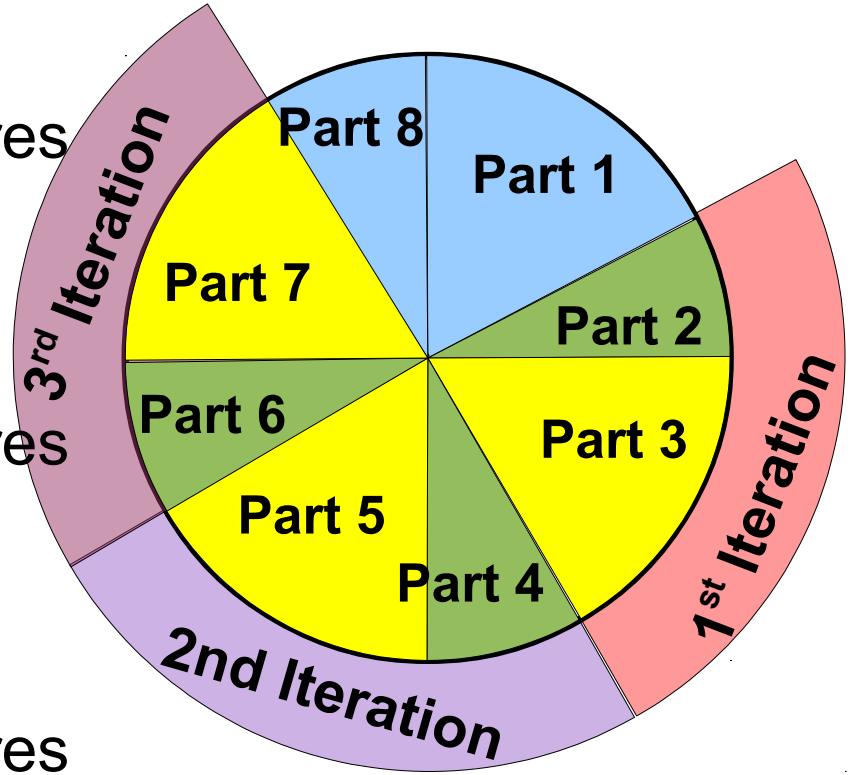
- Part 2: Planning and Selecting Features
- Part 3: Development

Iteration 2

- Part 4: Planning and Selecting Features
- Part 5: Development

Iteration 3

- Part 6: Planning and Selecting Features
- Part 7: Development
- Part 8: Retrospective



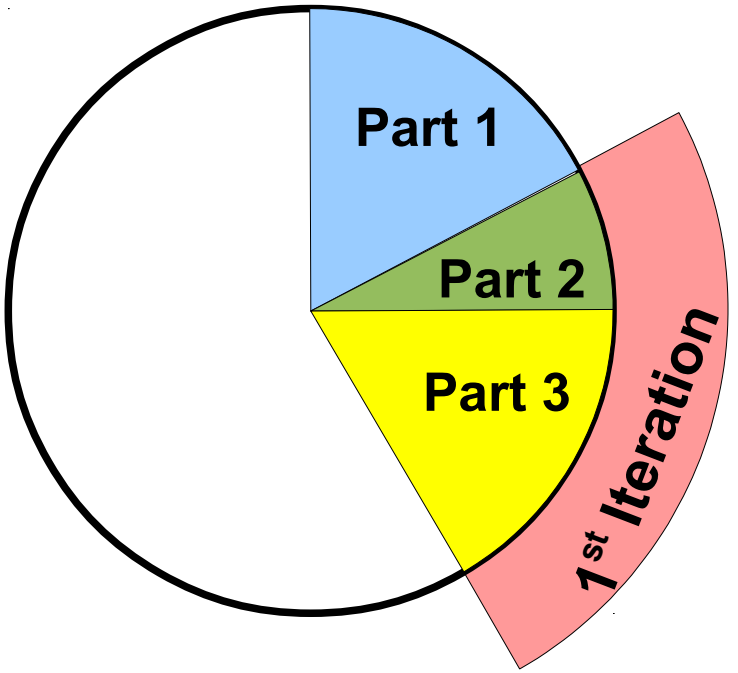
Planning Game – XP

Part 1: Writing and Estimating Stories — 10'

1st Iteration

Part 2: Planning and Selecting Features — 5'

Part 3: Development — 10'



Planning Game – XP

- Part 1: Writing and Estimating Stories (10 minutes)
  - Customers decide what they want and begin to develop feature cards – one feature per card. Keep the developers informed. Give them the cards as they are written.
  - Developers self organize and discuss things with the customers. When you get a card, estimate its time (effort for realization) in ideal minutes/seconds. Give it back to Customers velocity for the next period.
  - Monitors establish communication and are responsible for time schedule.

Planning Game – XP

- Part 2: Planning and Selecting Features (5 minutes)

1st Iteration

  - Developers decide on how many minutes of effective effort they think that they can deliver in a 10 minute "build" iteration.
  - Customers choose the most desirable features up to a time limit determined and announced by developers (velocity).
  - Monitors promote communication and are responsible for time line.

Planning Game – XP

- Part 3: Development (10 minutes)

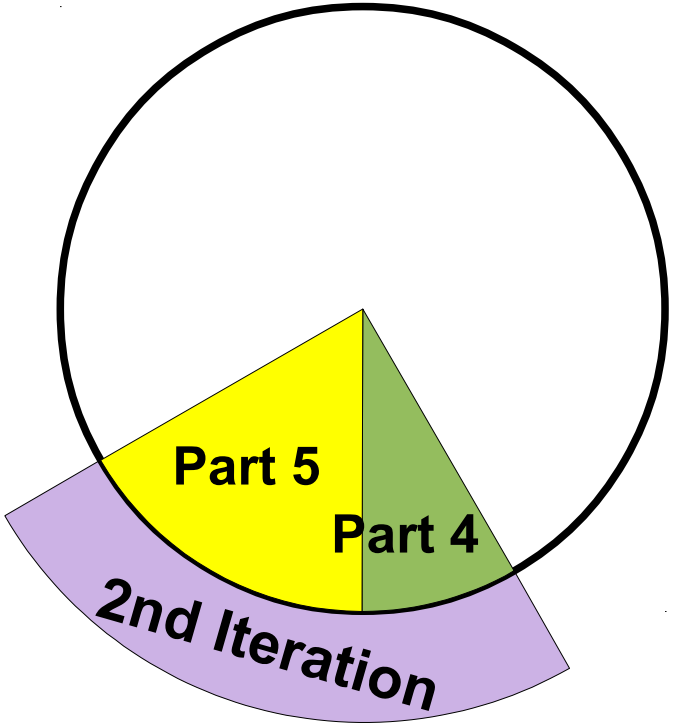
1st Iteration

  - Developers draw features on the cards given above and consult with customers as needed.
  - Customers can develop new cards as desired. These will be collected and could be used in **future** iterations.
  - Monitors help in process and meeting milestones

Planning Game – XP

2nd Iteration

Part 4: Reflection, More Stories, and Estimation	5'
Part 5: Development	10'



Planning Game – XP

- Part 4: Planning and Selecting Features (5 minutes)

2nd Iteration

  - Developers estimate its time for new cards / update estimates for given cards (effort to realize) in ideal minutes/seconds. In case they update, how many minutes out of “10” of effective effort could be given.
  - Customers choose the most desirable features up to a time limit determined and announced by developers (velocity).
  - Monitors promote communication and are responsible for time line.

Planning Game – XP

- Part 5: Development (10 minutes)

2nd Iteration

  - Developers draw features on the cards given above and consult with customers as needed
  - Customers can develop new cards as desired. These will be collected and could be used in **future** iterations.
  - Monitors help in process and meeting milestones

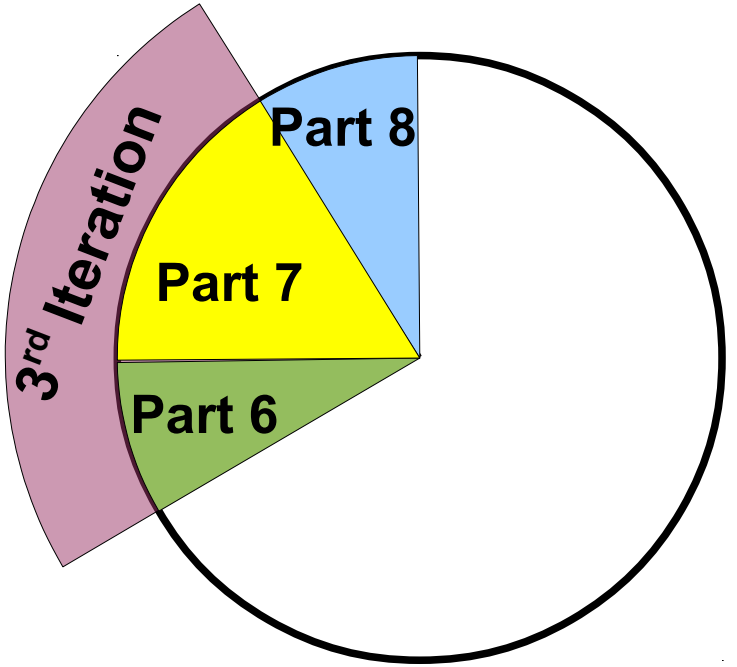
Planning Game – XP

3rd Iteration

Part 6: Reflection, More Stories, and Estimation — 5'

Part 7: Development — 10'

Part 8: Retrospective – Lessons Learned — 5'



Planning Game – XP

- Part 6: Planning and Selecting Features (5 minutes)

3rd Iteration

  - Developers estimate its time for new cards / update estimates for given cards (effort to realize) in ideal minutes/seconds.
  - Customers choose the most desirable features up to a time limit determined and announced by developers (velocity).
  - Monitors promote communication and are responsible for time line.

Planning Game – XP

- Part 7: Development (10 minutes)

3rd Iteration

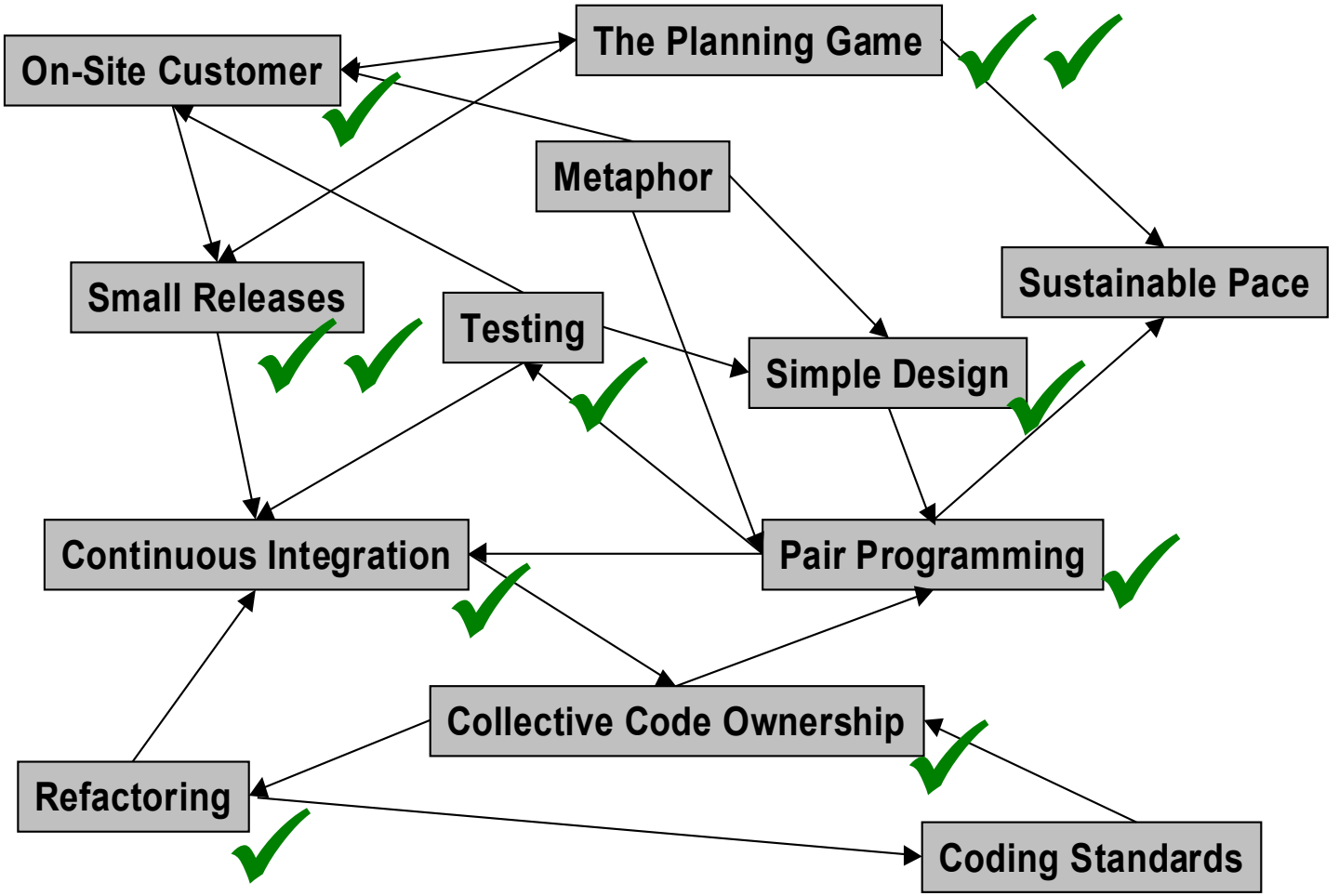
  - Developers draw features on the cards given above and consult with customers as needed.
  - Customers can develop new cards as desired. These will be collected and could be used in **future** iterations.
  - Monitors help in process and meeting milestones.

Planning Game – XP

- Part 8: Discussion
  (5 minutes)
  - How went the process?
  - What did you like?
  - What could be improved?

XP – 12 Practices

- 12 Practices



✓ = covered in Planning Game

XP – 12 Practices

- Fine scale feedback
  - Pair programming
    - 2 developers work together
    - Always at least 2 know the code
    - Change of roles as necessary (other user stories)
  - Testing – Test Driven Development
    - Developers write (to be automated) Unit-Tests before coding ("Test-First"-approach)
    - Customer defines parallel functionality tests

XP – 12 Practices

- Fine scale feedback
  - The Planning Game
    - Common release planning based on user stories
    - Prioritization by customer – Effort guess by developers
  - On-Site Customer
    - A representative of the customer is always available to discuss / answer questions and to get decisions concerning user stories and test

XP – 12 Practices

- Continuous process
  - Continuous Integration
    - If a user story is done, it gets integrated in the whole system
    - Testing before and after integration to ensure functionality
  - Refactoring
    - Every time when it is detected that the design could be improved, it has to be done
    - Unit-Tests assure, that the functionality still works
  - Small Releases
    - About every +/- 4 weeks to get early customer feedback

XP – 12 Practices

- Shared understanding
  - Coding Standards
  - Collective Code Ownership
    Everybody could change everywhere
  - Simple Design (→ Refactoring)
    - Design and Code as simple as possible
    - Not needed code gets deleted immediately
    - Implement only what is needed to fulfill an user story
  - Metaphor
    Simple story how the system should work instead of a complex architecture description

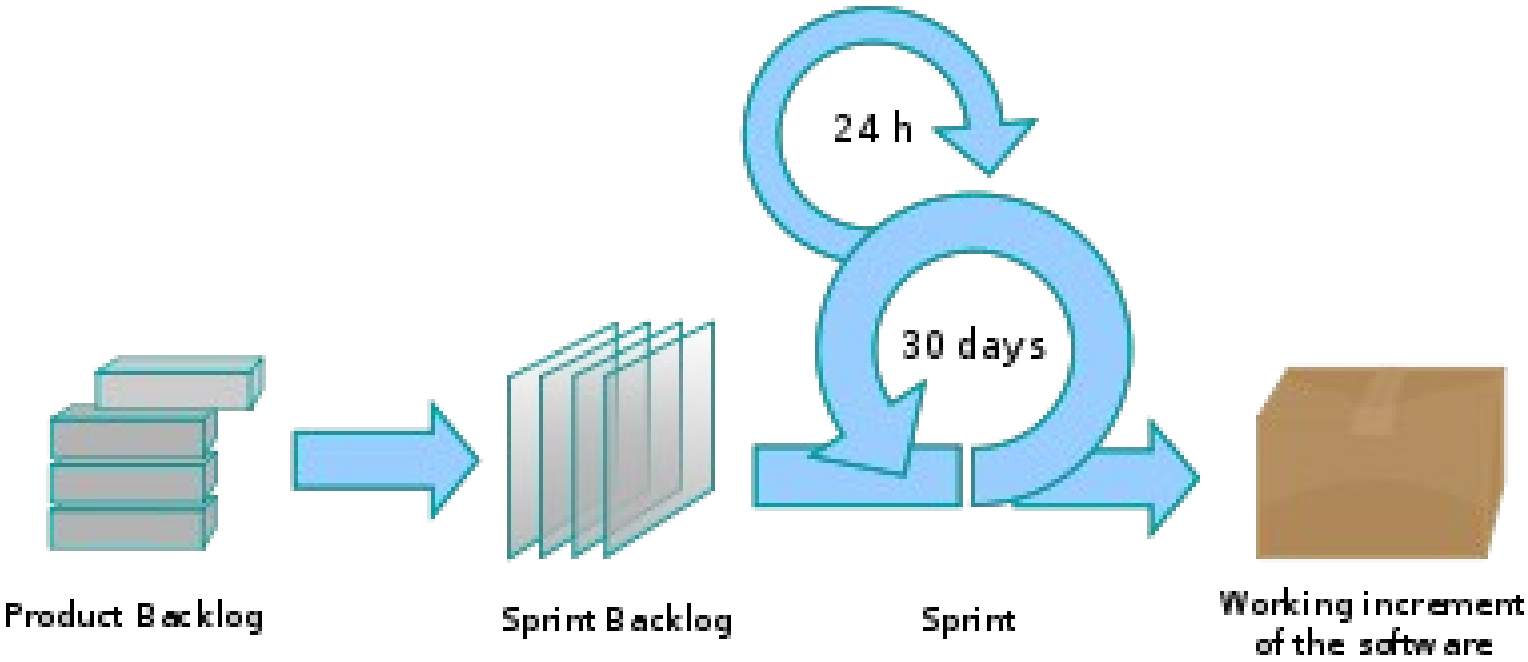
XP – 12 Practices

- Programmer welfare
  - Sustainable Pace
    The big needs in XP lead to intensive work, so that overtime should not be done

XP – Summary

- XP practices could be used in other Software Development Processes as well
  - "Test-First"-approach
  - Small releases and continuous / frequent integrations
  - Pair Programming
  - Refactoring to keep “projects well”
- A process model supporting XP is Scrum

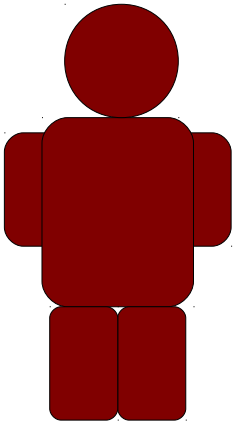
Scrum



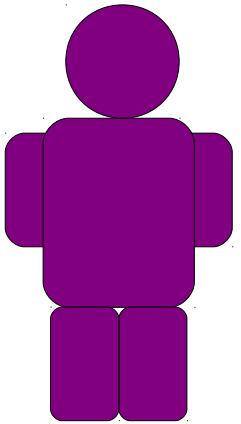
http://en.wikipedia.org/wiki/File:Scrum_process.svg

Scrum – Roles

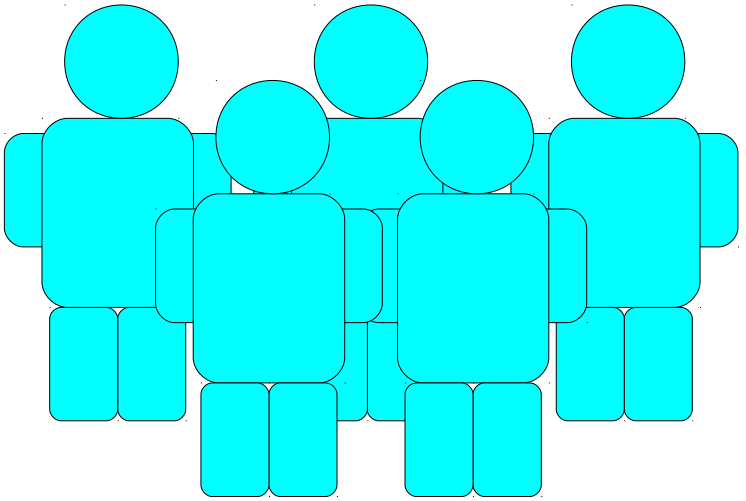
- responsible for maintaining the Product Backlog
- representing the interests of the stakeholders



Product Owner



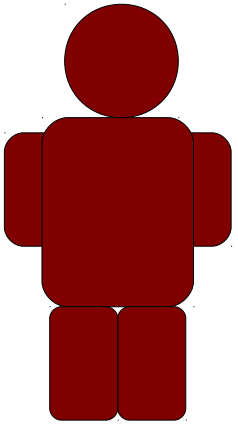
Scrum Master



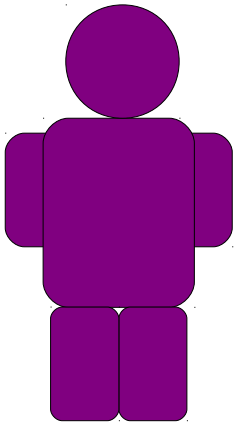
Team

Scrum – Roles

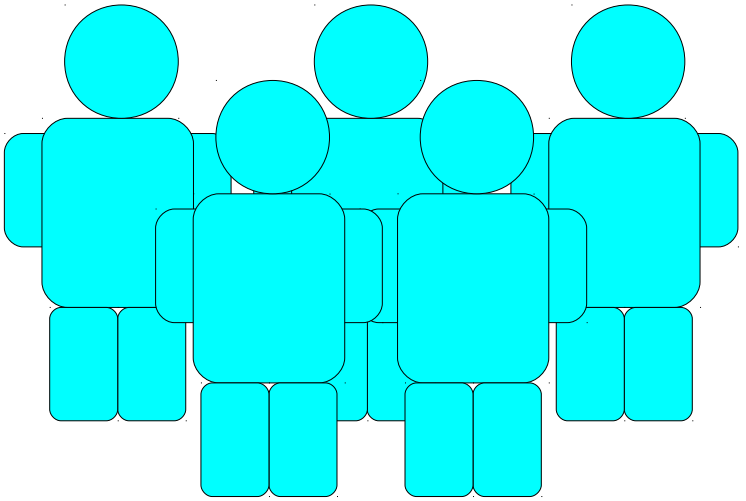
- responsible for the Scrum process,
- maximizing its benefits



Product Owner



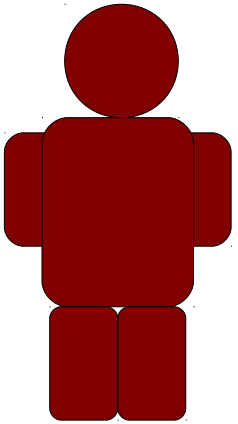
Scrum Master



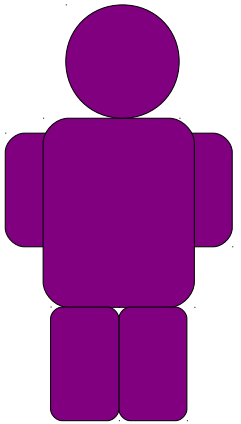
Team

Scrum – Roles

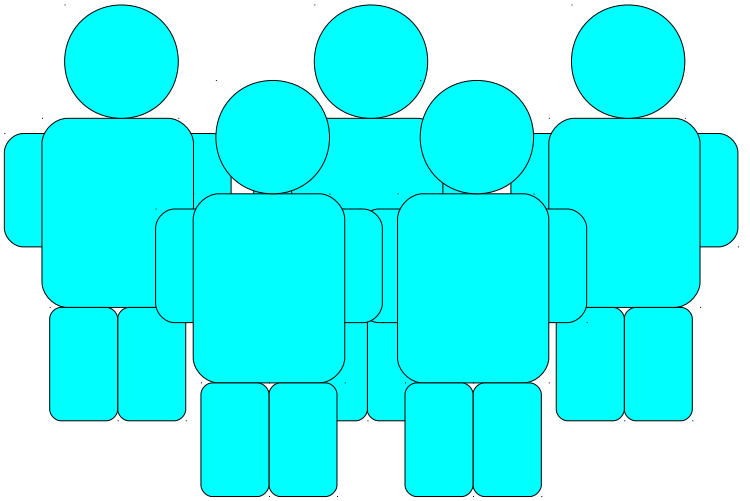
- responsible for managing itself to develop the product



Product Owner



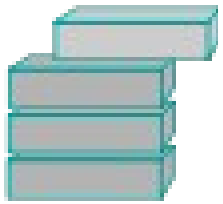
Scrum Master



Team

Scrum – Product Backlog

- Collection of User Stories as basic
  Wish list what makes the product great
- Basic for planning
- Based on given capacity of a sprint there will be an agreement, which of them should be realized in a sprint, following prioritization by customer



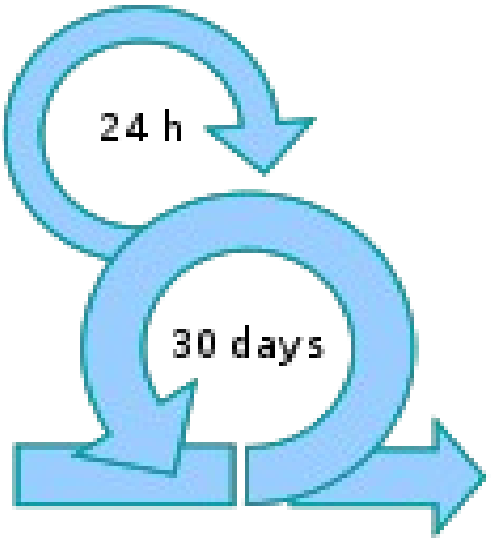
Product Backlog

Scrum – Product Backlog

- Basic of Product Backlog: Well written user stories, e. g. following the **INVEST** model [Wak03]
  - **I**ndependent – no overlap, no dependencies
  - **N**egotiable – captures the essence, not details
  - **V**aluable – a specified value for the customer
  - **E**stimable – to help in planning and prioritization
  - **S**mall – should be conducted in a sprint
  - **T**estable – more effective, if tests were written before implementation

Scrum – Sprint

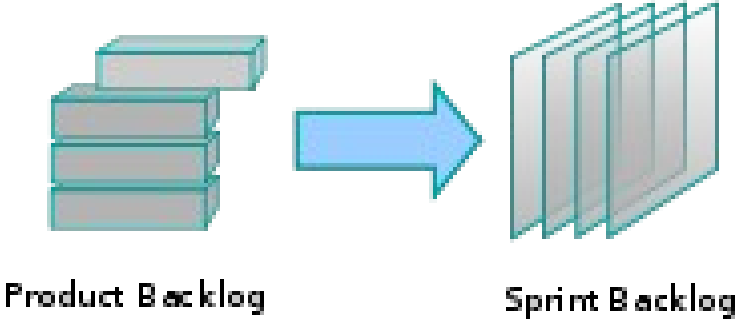
- Conducting a sprint
  - Sprint Planning
  - Daily Standup
  - Sprint Review
  - Sprint Retrospective
- Duration of a sprint depends on Release Cycle
  - Typical 2 to 4 weeks
- At the end of a sprint: Potentially deliverable product increment

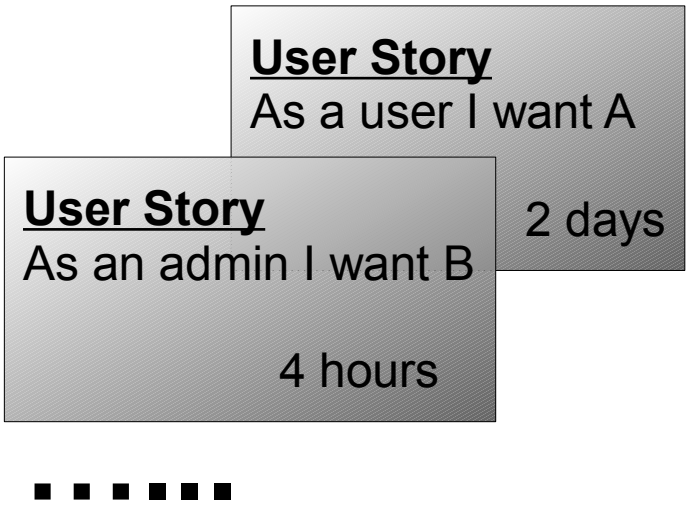


Sprint

Scrum – Sprint Planning

- Goal: Out of the Product Backlog definition of Sprint Backlog for current sprint
- Based on free capacity in a sprint agreement of user stories to be realized
  → Definition of done





Scrum – Sprint Planning

- Identifying tasks out of User stories
- Using “Planning Poker” to estimate development effort for an user story

User Story
As an scheduler I want to invite people so that the best fitting appointment date could be determined

Stories	To Do	In Progress	To Test	Done
As a User ...	Code A Test A Code B ...			
As an Admin	Code A Code B Test A			

Taskboard

Code A
Adding entity user into DB
4 h

Code B
Extending GUI with GWT
2 d

Scrum – Daily Standup

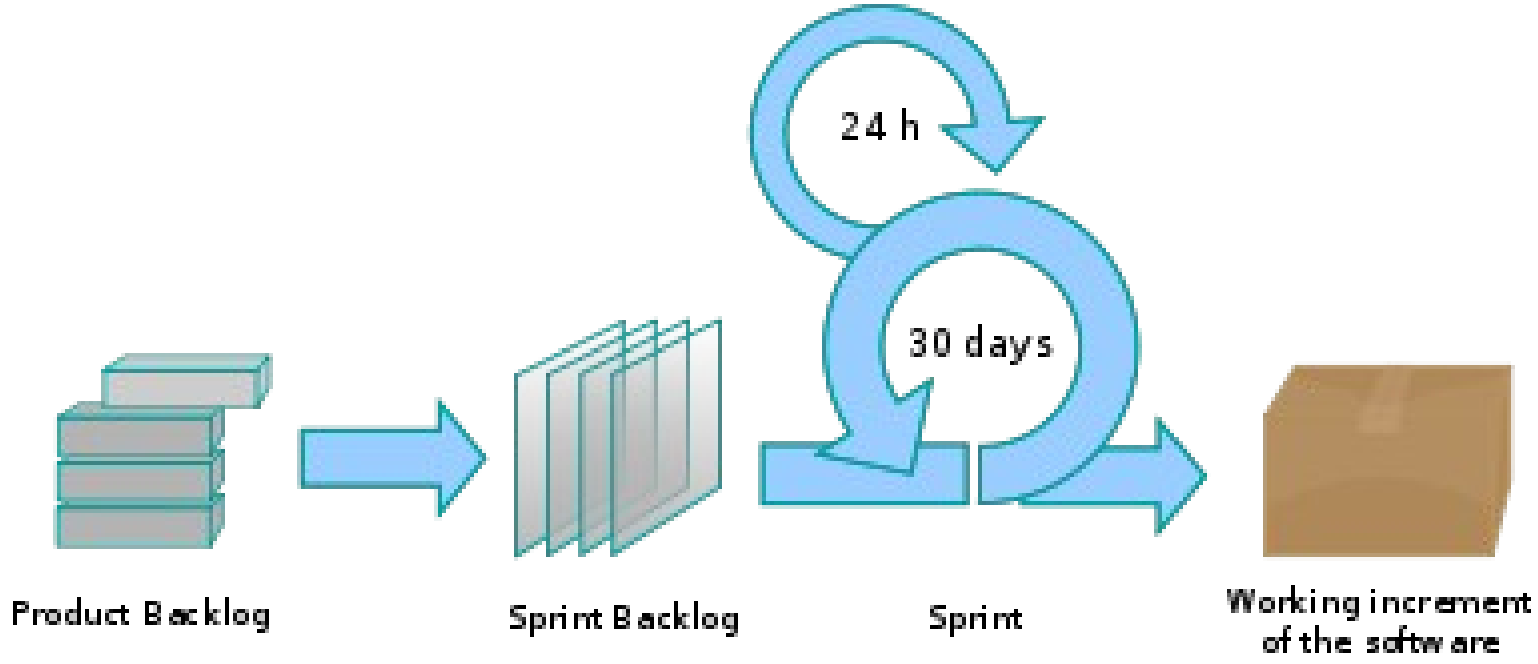
- 10 to 15 Min. – same time, same location, stand
  - What have I done since yesterday?
  - What am I am planning to do today?
  - Any impediments/stumbling blocks?
- Task board: Move of tasks

Stories	To Do	In Progress	To Test	Done
As a User ...	Code F Code G	Code B Code D Code E	Test C	Code C Code A Test A
As an Admin	Code A Code B Test A			

… out of [MGS12]

Scrum – Sprint review

- Team presents results of a sprint to the Product Owner
- Goal: Getting agreement from Product Owner to “tick” successful realized User Stories

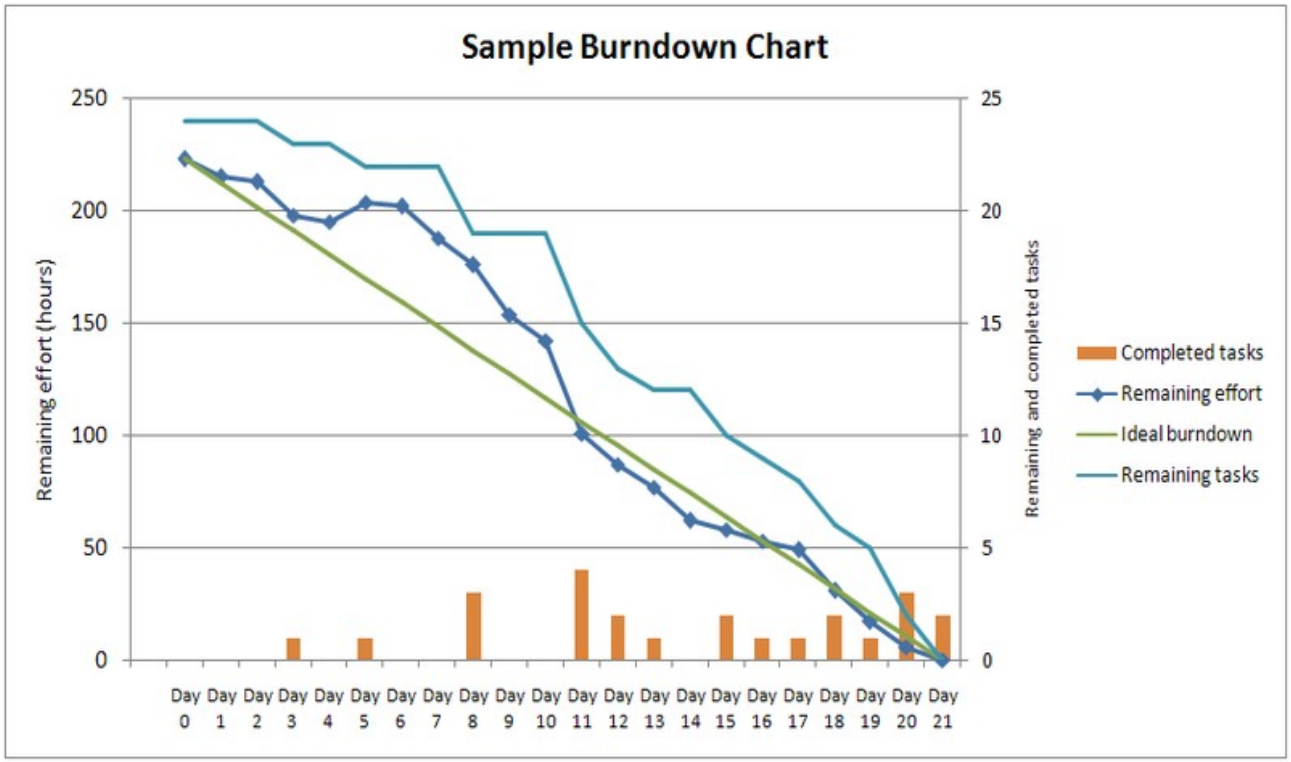


Scrum – Sprint Retrospective

- All team members reflect on the past sprint
- Make continuous process improvements
- Key questions:
  - What went well during the sprint?
    To be continued
  - What did not went will during the sprint?
    To be stopped
  - What should be started?
    To be implemented:
    Practices helping to work better

Scrum – Burndown chart

- Monitoring progress of the project with Burndown Charts – to show the current status in a sprint



http://en.wikipedia.org/wiki/File:SampleBurndownChart.png

Scrum – Summary

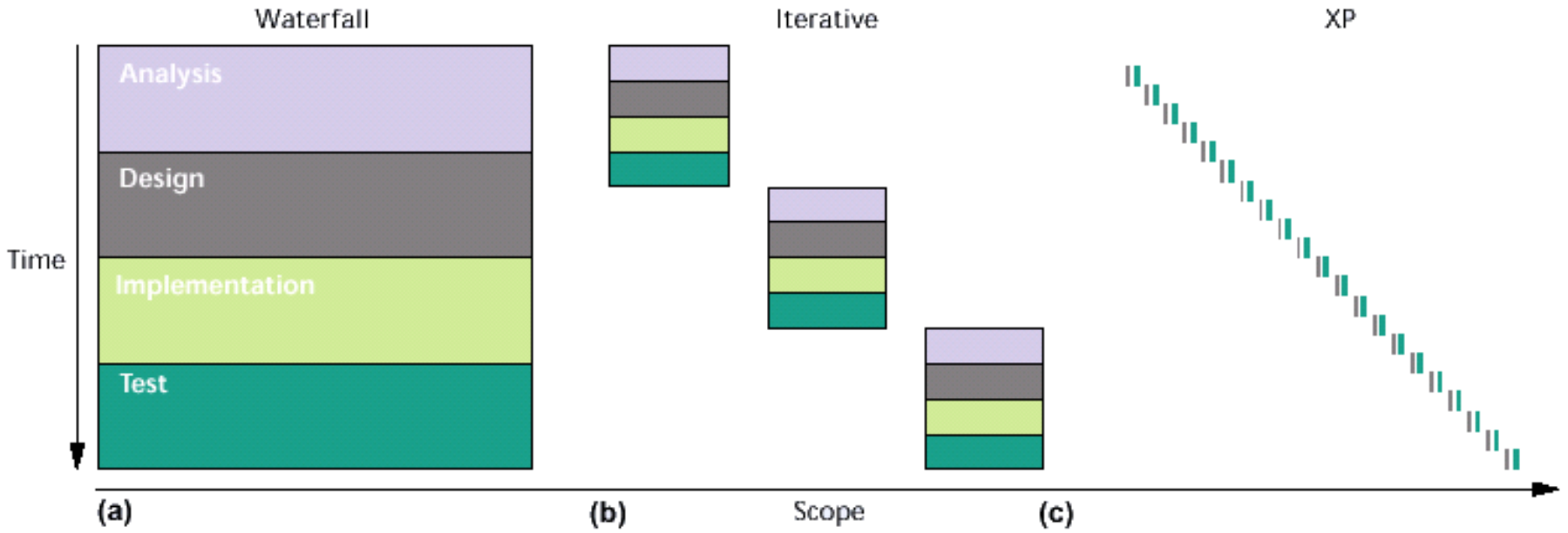
- Framework for lean software development
- 3 main roles are defined: Product Owner, Scrum Master, and the Team (7 +/- 2)
- Timebox approach with Sprints
  - Planning – prioritization
  - Run – on team's own responsibility
  - Review – for product
  - Retrospective – for process
- Daily scrum
- Scrum supports communication and learning

Comparison XP and Scrum

- XP and Scrum are philosophically very close
- Proposal: Put best of both together in practice [Kni07]!
- XP
  - takes "best engineering practices" to extreme levels.
  - introduces ideas like Test Driven Design (TDD), pair programming, programmer welfare.
- Scrum
  - Project management approach – concentrates on the management aspects of software development ('sprints').
  - Effort in removing impediments with daily scrum meetings, retrospective → Scrum Master.

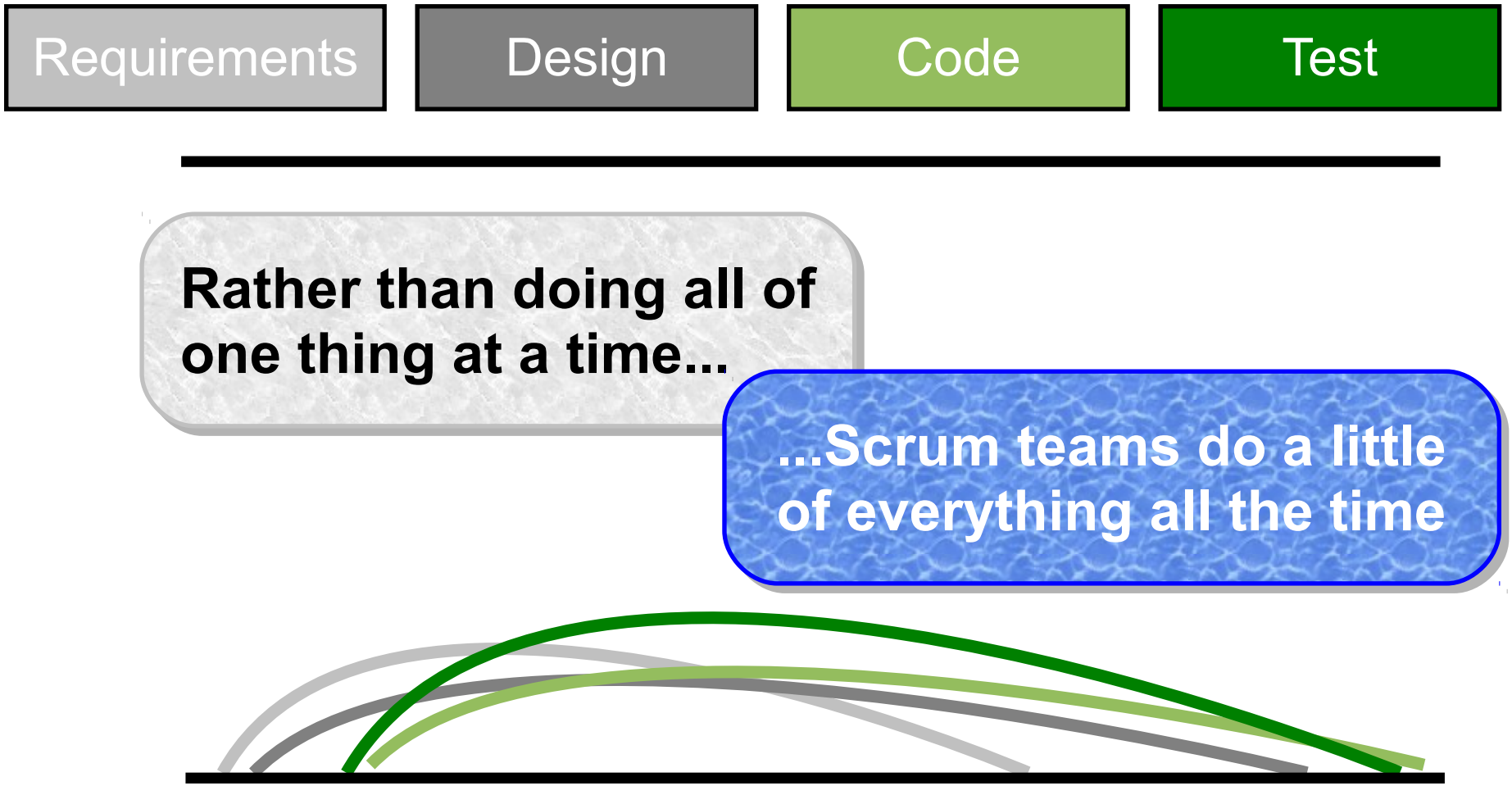
Process Model – XP

- Summary:



K. Beck, Embracing Change with Extreme Programming, IEEE Computer, Oct 1999.

Process Model – Scrum



Source: “The New New Product Development Game” by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.

… out of [MGS12]

Sources / More (1/2)

[Agi01] Agile Manifesto, 2001, http://agilemanifesto.org/

[Bec99]: Extreme Programming Explained: Embracing Change, Kent Beck, Addison Wesley, 1999.

[Ber01] Joseph Bergin: Learning the Planning Game – An Extreme Exercise, 2001, http://csis.pace.edu/~bergin/xp/planninggame.html

Methods & Tools, Software Development Magazine - Programming, Software Testing, Project Management, Jobs, www.methodsandtools.com

Hamid Shojaee: Scrum Master in Under 10 Minutes (HD) by @hamids, http://www.youtube.com/watch?v=Q5k7a9YEoUI

Scrum Alliance, Official Scrum Organization, non Profit; http://www.scrumalliance.org/ … with lots of weekly articles

Scrum.org; http://www.scrum.org/

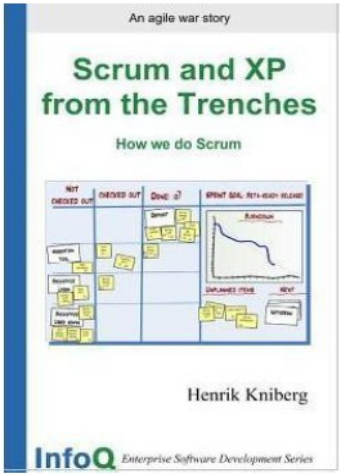
[MGS12] Mountain Goat Software: Introduction to Scrum, 2012, www.mountaingoatsoftware.com/scrum

[Wak03] Bill Wake: INVEST in Good Stories, and SMART Tasks, 2003, http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

[Wik12] Wikipeda, http://en.wikipedia.org/wiki/Agile_software_development

Sources / More (2/2)

- [Kni07] Henrik Kniberg: “Scrum and XP from the Trenches - How we do Scrum”, 2007, free version available (registration is required): http://infoq.com/minibooks/scrum-xpfrom-the-trenches



- Ken Schwaber is Author of three books on Scrum … out of [MGS12]