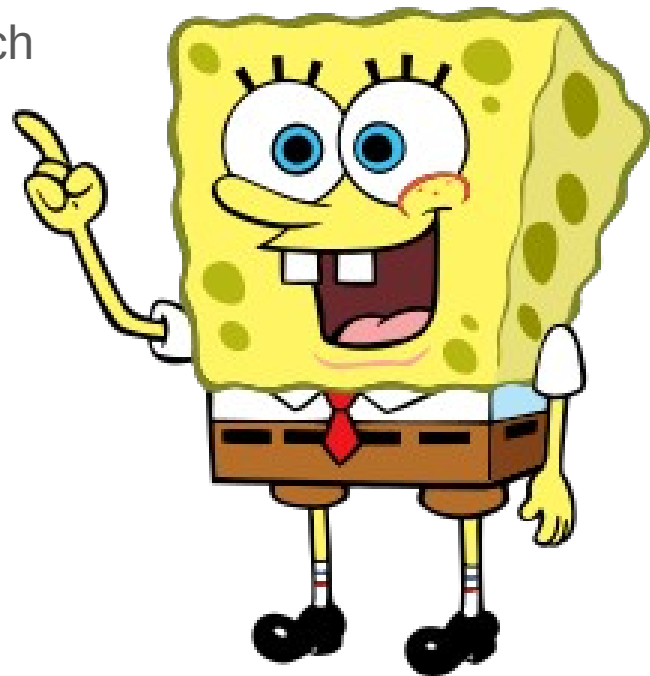# Scrum vs. XP

A quick introduction and comparison

Scrum Overview

- Scrum is an AGILE software development approach
    - iterative & incremental development
    - working software
    - release planning
    - iteration planning
    - daily meetings
    - retrospectives

- Scrum is a set of management principles …
    - practices
    - processes
    - … to complete a project - ANY project

Scrum Roles

- Product Owner
  - Understands the customer's needs
    & the business value
  - Creates & prioritizes the product backlog
    (Req. ⇒ User Stories)
  - Responsible for Return on investment (ROI)
  - Release plan
  - Responsible for the final product

source: http://manifesto.co.uk

Scrum Roles

- The Team
  - Self organized
  - Cross functional
  - Every team member is jointly responsible for the team's success:
    - SW-Quality
    - Technical implementation of the User Story
    - Delivery of functional Software (Testing/verification/validation)

source: http://manifesto.co.uk

Scrum Roles

- Scrum Master
  - Responsible for the Process
  - Facilitates the agile process of the company and the team
  - Keeps track of team process
  - Removes impediments
  - Scrum Master is NOT part of the team
  - NOT the BOSS

Product Owner
The Holder of Product Value

Determines what needs to be done and sets the priorities to deliver the highest value

Traditional approach:
Controls the work

ScrumMaster®
The Servant Leader

Protecting the Scrum process and preventing distractions

Traditional approach:
No equivalent

Development Team
The Self-Organizing Group

Takes on and determines how to deliver chunks of work in frequent increments

Traditional approach:
Gets told what to do by the project manager

Scrum Alliance®

source: https://www.scrumalliance.org/agile-resources/scrum-roles-demystified
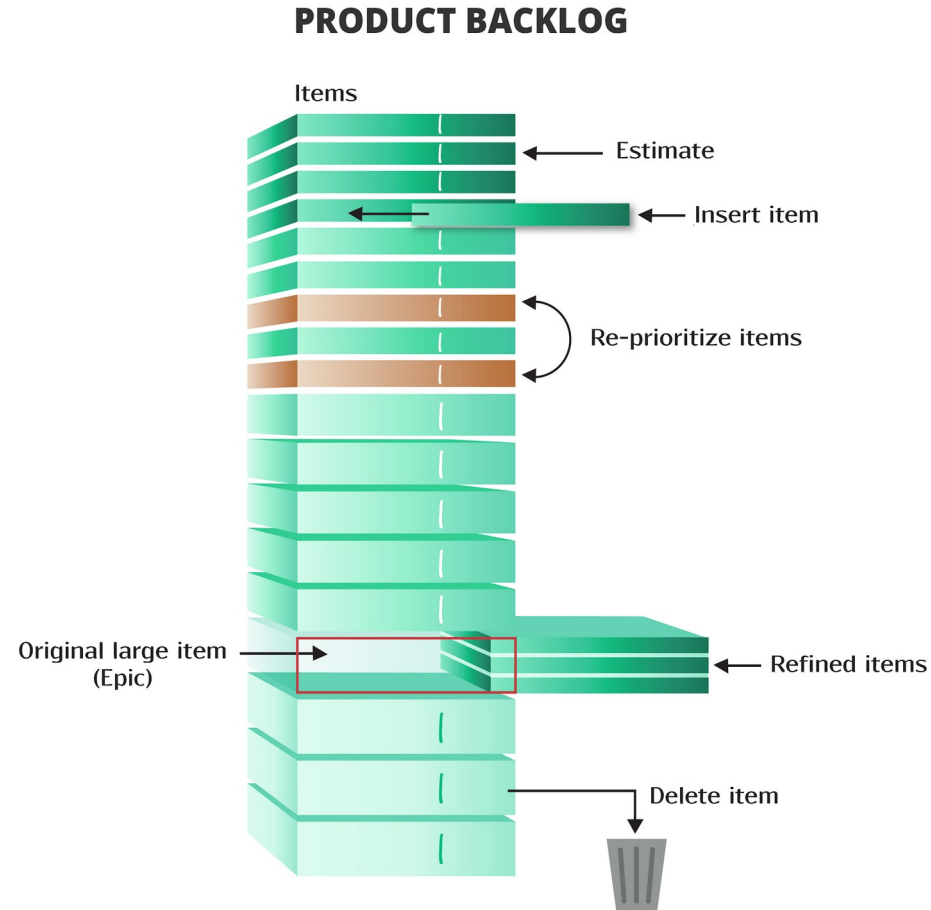
Scrum Artifacts

- Product Backlog
  - Everything or wanted for the entire product
  - Requirements are broken into chunks and written as User stories (As a user I want …)
  - Constantly changing and in revision
  - Prioritized
- Sprint Backlog
  - All committed stories the team works on in the iteration
  - Broken into tasks (Scrum board)
  - prioritized and estimated
- Burndown Chart
  - Shows the remaining work

PRODUCT BACKLOG

Items

Estimate

Insert item

Re-prioritize items

Original large item (Epic)

Refined items

Delete item

https://www.quickscrum.com/ScrumGuide/171/sg-Product-Backlog

Sprint

- The actual iteration lasting 1-4 weeks (usually 2-3 weeks)
- At the beginning the planning meetings take place
- The development takes place
- The daily Scrum takes place
- The sprint review takes place
- The Retrospective takes place
- IMPORTANT that there is NO overtime (to keep a consistent the pace)

Scrum Meetings

- Sprint planning meeting
  - PO selected stories are discussed
  - Team selects stories they believe can commit in sprint
  - Roughly estimate
- Often Sprint Planning is split
  - Sprint Planning 1 (mainly PO-Dev meeting)
    - Discussion of Stories
  - Sprint Planning 2 (more a Dev-meeting)
    - Break stories into tasks
    - Provide detailed/revised estimates
    - Set the scope and final commit to stories

Scrum Meetings

- Daily Standup Meeting
  - Who:
    - whole team attends (mandatory)
    - anyone MAY attend (e.g., PO)
  - Timeboxed (15 min)
  - Purpose:
    - Share progress with the whole team
  - Everyone answers three Questions
    - What have I done
    - What were the problems/Where do I need help
    - What will I concentrate on today
  - Lead by Scrum Master
   - Update burndown chart

Scrum Meetings

- The Sprint Review
  - PO uses the sprint's potentially shippable product increment
    (without guidance) => provides feedback
  - Developers may also demo work
  - Team owns up to deficiencies with the product helping to build trust with the Product Owner

- The Team Retrospective
  - Internal reviews
    - what went well
    - what must be improved
  - Use retrospection techniques to find potential for improvement.
  - Prioritize Focus areas for improvement
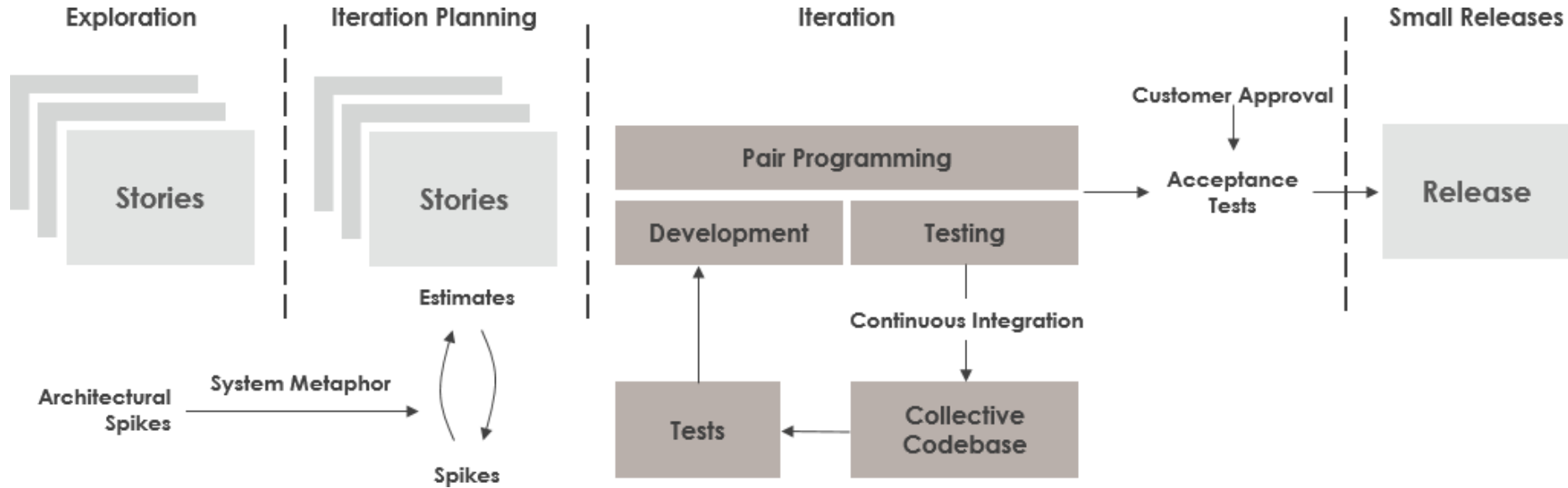 - Pick a driver for improvements

The Scrum cycle (says it all)
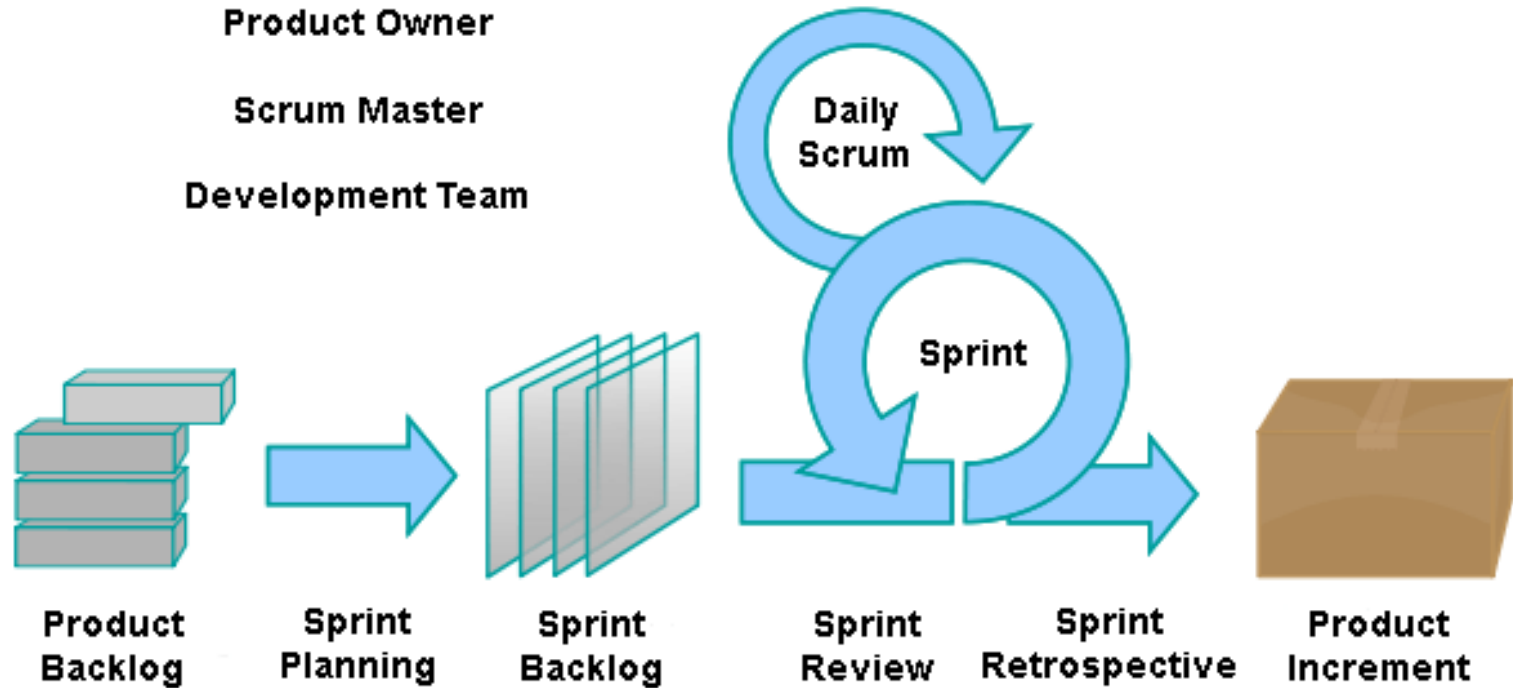
XP Overview

"XP is a lightweight methodology for small to medium sized teams developing Software in the face of vague or rapidly changing requirements"
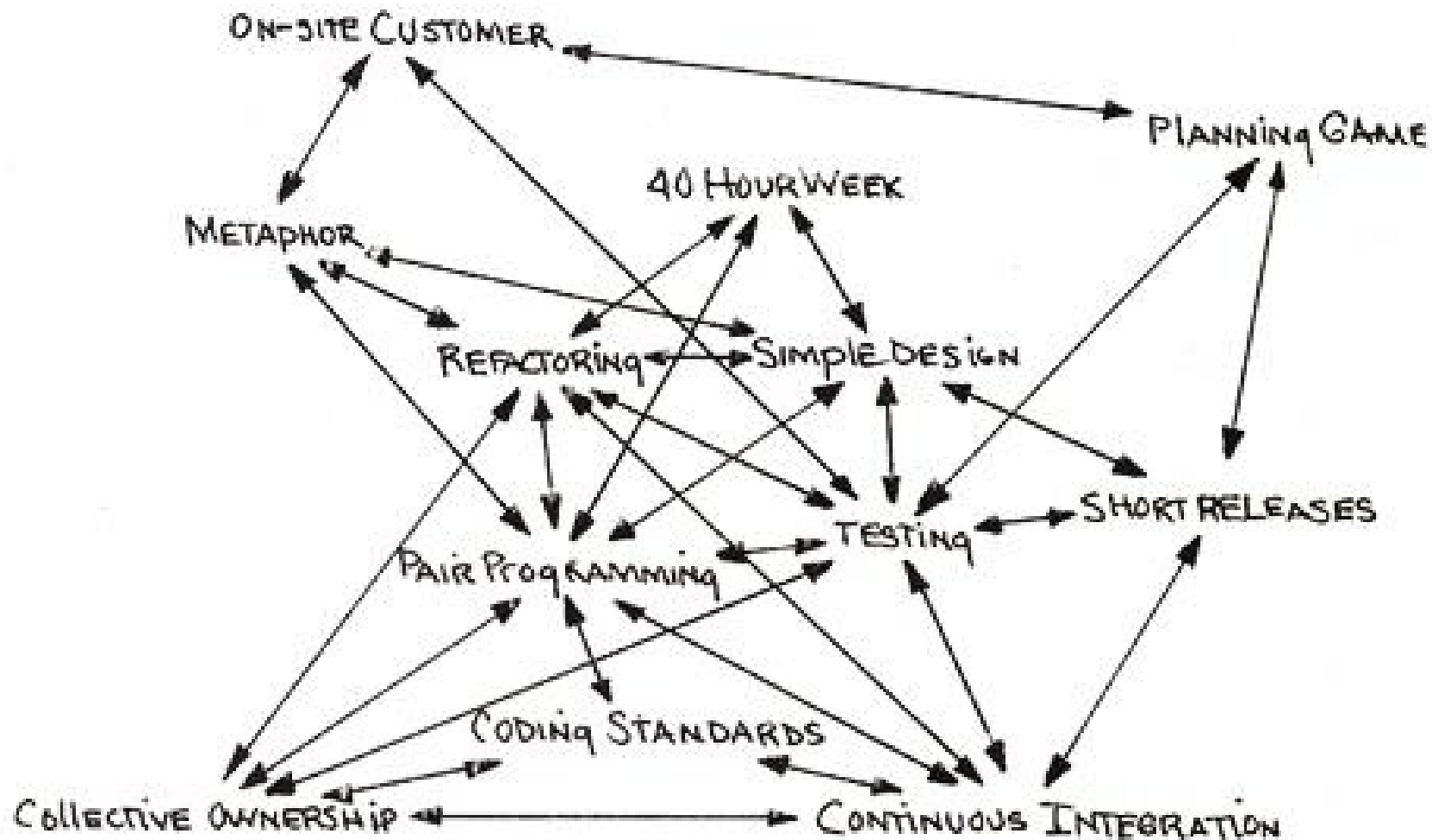
Kent BECK

The XP cycle (says it all)

Working Software - How?

- **XP practices**
  - *Whole Team*
  - Planning Game
  - Small Releases
  - **Customer Tests**
  - **Coding Standards**
  - Sustainable Pace
  - Metaphor
  - **Continuous Integration**
  - **Collective Code Ownership**
  - **TDD**
  - **Refactoring**
  - **Simple Design**
  - **Pair Programming**

On-Site Customer • Planning Game • Metaphor • 40 Hour Week • Refactoring • Simple Design • Short Releases • Testing • Pair Programming • Collective Ownership • Coding Standards • Continuous Integration

source: Kent Beck - Extreme Programming Explained

Differences between Scrum and XP - 1

- Iteration
  - Scrum Sprints: 1-4 (usually 2-3 weeks)
  - XP iterations: 1-2 weeks
- Change
  - Scrum teams usually do NOT allow changes during the sprint (has changed ;-)
  - XP: Embrace change
    When not in progress it is OK to swap features of equivalent size
- Prioritization
  - XP teams work in strict priority order
  - Scrum teams - not necessarily they have freedoms (although rarely do)
- Scrum doesn't prescribe ANY engineering practices
  - XP does

Conclusion

Best of all Worst of None
Master Ken

- Use scrum as a framework
- Implement the XP practices
  - Don't enforce them, coach them

source: https://blackbeltmag.com/wp-content/uploads/black-belt-photo-master-ken-150.jpg

References & Sources

agile42.com

brainconcert.com

mountaingoatsoftware.com

agilepainrelief.com

visual-paradigm.com

quickscrum.com