Munich Agile Deep Dive Conference


Speaker
Eike Gehler
Agile Coach
Visual Design Group GmbH

Friday, 15 June, 2018
WHO AM I
Eike D. Gehler

- Enterprise Agile Coach
- Scrum Master
- Product Owner
- Architect
- Family man
- Explorer
- Traveler
- Golf addicted
WHY?

SCRUM  XP
WATERFALL VS. AGILE METHODOLOGY

Where we come from

WATERFALL
Development of the software flows sequentially from the start point to the end point.
UPSTREAM/DOWNSTREAM EFFECT

the earlier the more expensive
The MCO Mishap Investigation Board has determined that the root cause for the loss of the MCO spacecraft was the failure to use metric units in the coding of a ground software file used in trajectory models.
REQUIREMENT MANAGEMENT
Using waterfall methodology
FEATURE USAGE

in Software Systems

ONLY 1/5th OF THE FEATURES ARE USED ON A REGULAR BASIS

2/3rd OF THE FEATURES ARE USED ALMOST NEVER

Source: Standish Group
THE AGILE MANIFESTO
AGILE MANIFESTO

2001

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

That is, while there is value in the items on the right, we value the items on the left more.

Source: http://agilemanifesto.org/
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.

Source: http://agilemanifesto.org/
THE PAST

SCRUM

XP
TOYOTA PRODUCTION SYSTEM

Automation of machine tools and having one person be in charge of several machines
TOYOTA PRODUCTION SYSTEM

"Start from need!"

Taiichi Ōno

https://www.logisticshalloffame.net/de/presse/pressebilder/mitglieder/taiichi-ohno#louno_300dpi
TOYOTA PRODUCTION SYSTEM
Invented from 1947 on

- **Objective:**
  High productivity and quality, punctual delivery

- **Just-in-time principle:**
  - Produce only for customer needs (build-to-order)
  - Pull system (Kanban)
  - Continuous flow of material
  - Versatile employees

- **Continuous improvement -** **Kaizen**
KAIZEN

- "Change for the better"
- Awareness and involvement of employees
- Gradual improvement
- Continuous learning
- Transparency
- Eliminate Muda, Muri and Mura
ELIMINATE

- Muda - Waste
- Muri - Overload
- Mura - Imbalance
Toyota consistently achieved four times the productivity and twelve times the quality of General Motors by the early 1990’s.

Jeff Sutherland
SCRUM → XP
A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Ken Schwaber / Jeff Sutherland
SCRUM – DEFINITION

LIGHTWEIGHT

SIMPLE TO UNDERSTAND

DIFFICULT TO MASTER

Source: ©Scrum.org and Scrum Inc., Ken Schwaber and Jeff Sutherland: Scrum Guide
Scrum is an agile framework designed to add energy, focus, clarity, and transparency to project planning and implementation. Today, Scrum is used in small, mid-sized and large software corporations all over the world.

**PROPERLY IMPLEMENTED, SCRUN WILL:**

- Increase speed of development
- Align individual and corporate objectives
- Create a culture driven by performance
- Support shareholder value creation
- Achieve stable and consistent communication of performance at all levels
- Enhance individual development and quality of life
**SCRUM — THEORY**
Based on empirical process control

<table>
<thead>
<tr>
<th>TRANSPARENCY</th>
<th>INSPECTION</th>
<th>ADAPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Main aspects of the process must be defined by a common standard.</td>
<td>✓ Scrum users must frequently inspect Scrum artifacts and progress towards a Sprint Goal.</td>
<td>✓ Process or the material being processed must be adjusted if an inspection determines that one or more aspects of a process deviate outside acceptable limits and the resulting product will be unacceptable.</td>
</tr>
<tr>
<td>✓ A common language that refers to the process has to be shared by all participants.</td>
<td>✓ Inspections should not be so frequent that they get in the way of the work.</td>
<td>✓ Sprint Planning</td>
</tr>
<tr>
<td>✓ A common “Definition of Done” for those performing work and accepting the work product.</td>
<td>✓ Inspections are most effective when diligently performed by skilled inspectors at the point of work.</td>
<td>✓ Daily Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ Sprint Review</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓ Sprint Retrospective</td>
</tr>
</tbody>
</table>

Source: ©Scrum.org and Scrum Inc., Ken Schwaber and Jeff Sutherland: Scrum Guide
Not all involved people are part of a Scrum Team – and therefore can directly influence the Development Team’s work.

This will help to reach focused the Sprint’s goal.
**SCRUM – TEAM**

Members on a Scrum Team

**PRODUCT OWNER**
A person responsible for maximizing the value of the product and the work of the development team (Product Backlog and managing priorities)

**DEVELOPMENT TEAM**
Consists of professionals who do the work for delivering a potentially releasable Increment of „Done“ product at the end of each Sprint, self-organizing, cross-functional

**SCRUM MASTER**
A person ensuring that Scrum is understood and enacted, sets up meetings and monitors everything

Source: ©Scrum.org and Scrum Inc., Ken Schwaber and Jeff Sutherland: Scrum Guide
SCRUM – ARTEFACTS

- Product Backlog
- Sprint Backlog
- Sprint 4 Weeks
- Daily Scrum Meetings
- 24 Hours
- Potentially Shippable Product Increment
- Product
SCRUM – EVENTS

- **Release Planning**
- **Sprint Planning**
- **Daily Scrum** (24h)
- **Sprint**
- **Shippable Increment**

**Burndown Velocity**
SCRUM – THE 5 VALUES OF SCRUM

Five Values of the Scrum Framework

Teams pay close attention to the results.

Teams hold each other accountable.

Team members ensure that the whole team is committed.

Team members engage each other in healthy conflicts.

Team members earn respect by building trust among the teams.
EXTREME PROGRAMMING (XP)
Ask 'why' five times about every matter.

Taiichi Ōno
The biggest cause of failure in software-intensive systems is not technical failure; it’s building the wrong thing.

Mary Poppendieck, Leading Lean Software Development: Results Are not the Point
I'm not a great programmer; I'm just a good programmer with great habits.

Kent Beck, Extreme Programming Explained: Embrace Change
EXTREME PROGRAMMING
The history

- Methodology for agile solution of programming tasks
- Developed in 1995 by Kent Beck, Ward Cunningham, Ron Jeffries
- Influenced by Lean Thinking and the design patterns of the "Gang of Four"
- Focus on techniques, values and principles first described by Kent Beck: "Extreme Programming Explained" (2001)
XP – PRACTICES, PRINCIPLES, VALUES
Similar to Scrum XP breaks a project into tiny phases, and developers cannot continue on to the next phase until the first phase is completed.
XP – PRINCIPLES

Similar to Scrum, XP breaks a project into tiny phases, and developers cannot continue on to the next phase until the first phase is completed.
XP –VALUES

Communication

Source: Kent Beck
XP – VALUES

Simplicity

Source: Kent Beck
XP – VALUES

Feedback

Source: Kent Beck
XP – VALUES

Courage

Source: Kent Beck
XP – PRACTICES

The core of Software development
XP – PRACTICES

Test Driven Development

1. Write a test that fails
2. Make the code work
3. Eliminate redundancy
XP – PRACTICES

Why we need integration tests...
XP – PRACTICES

Test Driven Development

- All tests are automated and will be executed with every build:
  - First **Unit tests** are written, only then the actual functionality is implemented
  - **Integration tests** examine the interaction of classes and subsystems
  - **Regression tests** are added once an error has been fixed to prevent a reoccurrence of the same error

- Advantages:
  - High safety and quality standards throughout the project
  - Reduced fear of changes and refactoring

Based on work of M. Cramer
Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?

Brian Kernighan

https://cdh.princeton.edu/people/brian-kernighan/
Refactoring

– Continuous improvement and change in architecture, design and code
– Therefore continuous optimization
– Changes are perceived as positive
– Advantages:
  – Better readability, testability and modularity of the code
  – By continuous improvement, large redesigns are rarely necessary

Based on work of M. Cramer
Refactoring is a disciplined technique for altering the internal structure of an existing body of code, without changing its external behavior. Each refactoring does little, but a sequence of transformations can produce a significant restructuring.

Martin Fowler
Source Code is a liability, not an asset.

Eric Lee

https://twitter.com/SaintGimp
Please do not turn off the light switch.

It also operates the elevator.

Thank You
Keep It Simple, Stupid.

Kelly Johnson

"You Aren’t Gonna Need It."

Ron Jeffries
"Code is inventory. It is stuff lying around and it has substantial cost of ownership. It might do us good to consider what we can do to minimize it.

Michael Feathers
**XP – PRACTICES**

Pair programming

- Two programmers work together on one machine
- One developer writes code ("**driver**")
- The other developer steers development ("**navigator**")
- No developer develops by himself
- The pairs of developers change regularly
- Advantages:
  - Permanent code review improves quality and avoids mistakes
  - Continuous transfer of knowledge
  - Project becomes less vulnerable to failure of individual developers
XP – PRACTICES

Continuous integration

- Code is continually checked in
- Automated build & smoke by check-in on main branch
- **Including automated tests**
- Minimal occurrence: Nightly Build
- Advantages:
  - Greater project security
  - Higher quality, integration errors are detected early
  - Educate the developers

Based on work of M. Cramer
Integrate often
XP – PRACTICES

Incremental delivery
TAKEAWAY

Extreme Programming

- Writing **unit tests** before programming and keeping all of the tests **running** at all times. The unit tests are automated and eliminates defects early, thus reducing the costs.
- Starting with a simple design just enough to code the features at hand and **redesigning** when required.
- Programming in **pairs** (called pair programming), with two programmers at one screen, **taking turns** to use the keyboard. While one of them is at the keyboard, the other constantly reviews and provides inputs.
- **Integrating** and **testing** the whole system several times a day.
- Putting a minimal working system into the production quickly and upgrading it whenever required.
- Keeping the customer **involved** all the time and obtaining constant feedback.

Includes work of M. Cramer
ABSTRACT

SCRUM

XP
ABSTRACT

KAIZEN

AGILE

SCRUM

XP

PHILOSOPHY

PRINCIPLES & VALUES

MANAGEMENT FRAMEWORK

DEVELOPMENT PRACTICES
SCRUM

Information from end-users, customers, team and stakeholders

PRODUCT OWNER

THE TEAM

SCRUM MASTER

DEFINITION OF DONE

AGREED MINIMUM SET OF COMMON ACCEPTANCE CRITERIA

Sprint end date, goal and team deliverables do not change

SPRINT 1-4 WEEKS

SPRINT PLANNING MEETING

SPRINT BACKLOG

TEAM AGREES ON HOW MUCH TO COMPLETE BY SPRINT'S END

TASK BREAKOUT

PRIORITIZED FEATURE LIST

PRODUCT BACKLOG

SPRINT REVIEW & RETROSPECTIVE

DAILY STAND-UP MEETING

REVIEW

DONE PRODUCT

RETROSPECTIVE
THANK YOU!