Skalierung von agilen Prozessen

Ein Erfahrungsbericht

OOP 2003

Jutta Eckstein

Nicolai Josuttis

This Talk is About

- **Agility**

- **Large**

- **Experience**

- **Success**

What Does "Large" Mean?

- **Large in ...**
  - scope
  - time
  - people
  - money
  - risks
- **We focus on „Large Teams"**
  - which implies everything else
- **Large is relative**
  - 1, 2, 10, 100, 2000 people

3

Why is "Large" an Issue?

- **Aspects do not „scale linearly"**
  - new kinds of problems occur

- **For example: Communication**
  - with 2 people you have to communicate
  - with 10 people you can't discuss all aspects in one group
  - with 100 people you can't have all people in one room
  - with 1000 people you can't know all people

4

„Large" and Agile Methods

- **XP**
  - typical team size < 12
- **Scrum**
  - Ken Schwaber reports from teams with 400 members
- **Crystal**
  - different colors for different team sizes:
    - clear: for teams < 10
    - orange: for teams < 40
    - red, blue, ... (not defined yet)
- **FDD**
  - teams are assembled for designing a feature

5

The Agile Manifesto

- **developed in February 2001**
- **by key people of the lightweight processes community:**
  - Kent Beck, Ward Cunningham, Martin Fowler
  - Alistair Cockburn
  - Jim Highsmith
  - Dave Thomas, Andrew Hunt
  - Ken Schwaber, Michael Beedle
  - Steve Mellor
  - …
- **see agilemanifesto.org**

6

Values of the Agile Manifesto

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

7

Principles behind the Agile Manifesto

- **Early and continuous delivery of valuable software**
- **Welcome changing requirements**
- **Deliver working software frequently**
- **Business people and developers work together**
- **Trust motivated individuals**
- **Face-to-face conversation**
- **Working software is the primary measure of progress**
- **Constant pace indefinitely**
- **Technical excellence and good design**
- **Simplicity is essential**
- **Self-organizing teams**
- **Team reflection and adjustment**

8

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Business people and developers work together
- **Trust motivated individuals**
- Face-to-face conversation
- Working software the primary measure of progress
- Constant pace
- Tec
- Sim
- Self
- Team reflection and adjustment

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

9

Transparency

- **Communication and transparency for**
  - developers
  - QA
  - controlling / revision
  - customers
- **Practices and values:**
  - shared ownership
  - shared knowledge
  - no head monopolies
  - honesty

10

Wiki-Web

Web Based Collaboration Platform

- **shared vivid documentation of the project**
  - ideas, concepts, problems, tasks, questions, …

- **(write) access for all stakeholders**
  - developers, managers, customers, QA, controlling

- **easy to use**
  - self-explaining, free, stable
  - just a web server and standard browsers
  - TWiki.org

11

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Business people and developers work together
- Trust motivated individuals
- **Face-to-face conversation**
- Working software is the primary measure of progress
- Constant pace
- Techn
- Simpli
- Self-o
- Team re

> **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**

12

Communication

- **Face-to-face communication is always preferred**
- **But:**
  - large teams have limitations
  - feedback might be a problem

Therefore:

- **Communication channels must be a subject of change**
  - use different communication channels and switch them over time
  - locate the project members as close together as possible
  - change the locations ("floating desks")
- **Establish a (virtual) communication team**

13

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- **Business people and developers work together**
- Trust motivated individuals
- Face-to-face communication
- Work
- Con

  Business people and developers must work together daily throughout the project.

- Technical excellence and good design
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

14

Customer Involvement

Defined single customer is rare, more typical are:

- **Large invisible customer base**
  - typical for standard software
- **Community of customers**
  - often not homogenous, but competitive
  - no accepted representative

Therefore:

- **„Customer on-site office"**
  - specifies and performs acceptance tests
- **Customer surrogate**
  - *"designing for a single customer is the most effective way to satisfy a broad audience"*

  [Alan Cooper in The Inmates Are Running the Asylum]

15

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- **Deliver working software frequently**
- Business people and developers work together
- Trust motivated individuals
- Face-to-face communication
- Working software
- Constant pace
- Technical excellence
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

16

Development Cycles

- **The larger the team,
the shorter the development cycles**

- **We promote:**
  - 1 week iterations
  - 1 month releases
  - 3 month external releases

17

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welco
- Delive
- Busin
- Trust

    Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

- Face-to-face co
- Working softw   is the primary measure of progress
- **Constant pace indefinitely**
- Technical excellence and good design
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

18

9

Time-Box Development

- **Development cycles are fixed time boxes**
  1. plan amount of work for amount of time
     - consider vacation and real time
  2. develop
  3. deliver
  4. retrospective

- **Fixed time limit**
  - result or reason for failure

- **Advantages:**
  - teams concentrate on results
  - teams learn to estimate their speed
    - about 80% is a typical stable value
  - allows early computing of remaining tasks

19

Principles behind the Agile Manifesto

- **Early and continuous delivery of valuable software**
- **Welcome changing requirements**
- **Deliver working software frequently**
- **Business people & developers work together**
- **Trust motivated**
- **Face**
- **Work**
- **Cons**
- **Technical excellence and good design**
- **Simplicity is essential**
- **Self-organizing teams**
- **Team reflection and adjustment**

> **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**

20

Planning Development Cycles

- **Plan for accomplishing a feature**
  - often unusual for large teams
    - they are used to develop and plan against milestones.

- **Sometimes the iterations might be too short for a feature**
  - double check if the feature couldn't be further broke down
  - allow that feature development will exceptionally cover more than one iteration

- **Detailed planning only for the next steps**

21

NICOLAI M. JOSUTTIS
solutions in time

Jutta Eckstein
Objects in Action

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- **Welcome changing requirements**
- Deliver working software frequently
- Business people and developers work together
- Trust motivated individuals
- Face-to-face communication
- Working software
- Constant pace
- Technical excellence
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

22

Architecture

Architecture

- **is always subject of change**
  - progress means change
  - don't try to finalize the architecture before growing the team
    - Domain teams will formulate the requirements
    - the development level of the retrospectives enable to speak with one voice
- **is influenced by team size**
  - avoid any kind of bottleneck (technical, structural, organizational)

But:

- **You might have to define the architecture as finalized**

23

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Busi...
- Trust me...
- Face-to-face co...
- Working softw... primary measure of progress
- Constant pac... finitely
- Technical excellence and good design
- **Simplicity is essential**
- Self-organizing teams
- Team reflection and adjustment

> **Simplicity — the art of maximizing the amount of work not done — is essential.**

24

Simplicity

- **KISS**
  - *„I built a lot of large systems,
    but I never built a complex system"*

    [Kerth, Meszaros, Doble, OOPSLA2000]

- **Because simple systems**
  - are better maintainable
  - don't require smart programmers

- *„Simplicity comes from conceptual integrity"*

  [David Parnas, XP2002]

- **A system architect is the main step towards conceptual integrity**

25

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Delive
- Busin
- Trust motivated
- Face-to-face co
- Working softw the primary measure of progress
- Constant pace indefinitely
- **Technical excellence and good design**
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

> **Continuous attention to technical excellence and good design enhances agility.**

26

Refactoring

- **Missing skills**
  - they don't know how, why, when and where to refactor
  - unit tests are the safety net

- **Code size and compile time matters**
  - for example:
    - change an interface specification in a fundamental class

- **Refactoring with global impact has to be done outside business hours**
  - weekend
  - holidays

27

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welco...

  Working software is the primary measure of progress.

- Delive...
- Business peopl... ...pers work together
- Trust motivate... ...duals
- Face-to-face c...versation
- **Working software is the primary measure of progress**
- Constant pace indefinitely
- Technical excellence and good design
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

28

Force Discipline

- **Guidelines**
  - good practices (essence of experience)
  - „no code without test code"
- **Deprecated interfaces**
  - and getting rid of them
- **Reviews and code inspection**
  - external
  - internal (by developers and scripts)
  - active marketing of concepts and interfaces

29

Integration Concepts

- **Beware:**
  - **integration is a bottleneck you can't avoid**

- **Two approaches:**
  - **individual integrations**
    - one change after the other
    - often: continuous integration
  - **collective integration**
    - multiple changes at once
    - often: nightly builds

30

Integration Issues

- **Problems of individual integrations:**
  - time limitations for integrating one team after the other
  - for example:
    - framework build: 2.5 hours
    - domain build: 6 hours
    - unit tests: 4 hours

- **Problems of collective integration:**
  - error detection is more difficult
  - it takes time to have nightly builds running (1-2 years)

31

Addressing Integration Problems

Therefore:
- **Plan resources**
  - at least 1/10 of developers for integration/build
- **Use integration tools**
  - e.g. Cruise Control
- **Establish integration & build team**
  - collects changes
  - maintains progress
  - measures project status
  - provides scripts for build and integration

32

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Business people and developers work together
- Tru
- Fac
- Wo
- Constant pac
- Technical ex   d good design
- Simplicity is   ial
- Self-organiz   g teams
- **Team reflection and adjustment**

> **At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.**

33

Retrospectives

- **Retrospectives cover two levels:**
  - development level: evolution of the software
  - meta level: inspection of the process

- **Attendance:**
  - team leads
  - floating

34

Force Courage

- **Errors are Part of the Process**

- **"Do It Right the First Time" sends the wrong message**
  - we can't be uncertain
  - we can't experiment
  - we can't learn from mistakes
  - we can't deviate from plan
  - *"Don't worry about getting it right the first time, get it right the last time"*

    [Jim Highsmith, OOPSLA 2000]

- **Change people and roles**
  - many projects fail due to people in wrong positions

- **Honesty**
  - bad news are good news

35

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Busi...
- Trus...
- Face-to-face c...
- Working softw... ...mary measure of progress
- Constant pac... ...tely
- Technical ex... ...nce and good design
- Simplicity is essential
- **Self-organizing teams**
- Team reflection and adjustment

> **The best architectures, requirements, and designs emerge from self-organizing teams.**

36

Responsibility

- **Members of large teams are often not used to take up responsibility**
  - show confidence in their ability to bear responsibility
  - because:
    - „Telling people what to do doesn't guarantee that they will learn enough to think for themselves in the future. Instead it may mean that they'll depend on you or their superiors even more and that they will stop taking chances, stop innovating, stop learning."

      [The Fast Company (http://www.fastcompany.com)]

37

People are different

- **People and teams are different**

- **Let subteams decide**
  - each subteam defines its own process

- **Don't over specify and overrule**
  - for example: no need to require pair programming

- **Remember:**
  - "Individuals and interactions over tools and processes"

38

The Essence

- **Communication is the key**
  - Force feedback
- **Stability means death**
  - Be agile
- **Break rules**
  - Force courage
- **Common sense**
  - Use your brain (and feel/smell the problems)

39

Links

agilemanifesto.org

crystalmethodologies.org

xprogramming.com

adaptivesd.com

www.controlchaos.com

junit.org

twiki.org

cruisecontrol.sourceforge.net

40

NICOLAI M. JOSUTTIS
solutions in time

JUTTA ECKSTEIN
Objects in Action

Many Thanks!

Nicolai M. Josuttis
solutions@josuttis.com
www.josuttis.com

Jutta Eckstein
jutta@jeckstein.com
www.jeckstein.com

41