

# Agile Software Development in the Large

GI-Vortrag Braunschweig

Jutta Eckstein  
Nicolai Josuttis

## 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

### 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

### 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](http://agilemanifesto.org)**

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

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**
- **Promote sustainable development**
- **Technical excellence and good design**
- **Simplicity is essential**
- **Self-organizing teams**
- **Team reflection and adjustment**

„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

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
- Promote sustainable development
- Technical excellence
- Simplicity
- Self-organizing teams
- 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.**

Trust Motivated Individuals

Trust is based on:

- Communication
- Transparency
- Honesty
  - also bad news are good news
- Empowerment for decisions

Trust regards all:

- Developers
- Customers

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
- Promote sustainable development
- Technical excellence
- Simplify and maximize the amount of work not done
- Self-organizing teams
- Team collaboration

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

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**

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 together daily
- Prioritize people over process
- Technical excellence and good design
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

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

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 Coach**
- **„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]

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 collaboration
- Working software is the primary measure of progress
- Technical excellence and the ability to maintain a sustainable pace
- 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.**

Development Cycles

- **The larger the team, the shorter the development cycles**
- **We promote:**
  - 1 week iterations
  - 1 month releases
  - 3 month external releases
- **Time-boxed development:**
  - teams concentrate on results
  - teams learn to estimate their speed
    - about 80% is a typical stable value
  - allows early computing of remaining tasks

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 team members
- Face-to-face conversation is the most effective and efficient communication
- Working software is the primary measure of progress
- 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.**

Planning Development Cycles

- **Plan for accomplishing a feature**
  - often unusual for large teams
    - they are used to develop and plan against components or activities.
- **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**
  - Planning and estimation is an ongoing activity
  - Quality of estimates and progress

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 together
- Progress is measured by working software
- 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.**

Welcome Changing Requirements

Success is defined by the customer, not by anybody else.

But:

- A change means:
  - Time delay
  - Elimination of another feature

Therefore:

- „Say No“
  - Trade new or changing requirements against time or another feature
  - Especially for fixed price projects
- **Make sure also the architecture is a subject of change**
  - Progress implies change

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Business and technical collaboration
- Trust in self-organizing teams
- Face-to-face communication
- Working software is the primary measure of progress
- Promote sustainable development
- 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.**

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**

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Business and technical people work together daily
- Trust motivated team members
- Face-to-face communication is the most effective method of conveying information
- Working software is the primary measure of progress
- Promote sustainable development
- **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.**

Force Discipline

- **Guidelines**
  - good practices (essence of experience)
  - „no code without test code“
  - Refactoring
    - Code size and compile time matters
- **Deprecated interfaces**
  - and getting rid of them

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcoming working software over comprehensive documentation
- Delivering frequently, releasing incrementally
- Business people and developers work together daily
- Trust motivated individuals
- Face-to-face conversation
- **Working software is the primary measure of progress**
- Promote sustainable development
- Technical excellence and good design
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

**Working software is the primary measure of progress.**

### Integration Concepts

- **Beware:**
  - integration is a bottleneck you can't avoid
- **Two approaches:**
  - **individual integrations**
    - one change after the other
      - time limitations for integrating one team after the other
    - known as: continuous integration
  - **collective integration**
    - multiple changes at once
      - error detection is more difficult
      - it takes time to have nightly builds running (1-2 years)
    - known as : nightly builds

### 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

### 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
- Face-to-face conversation
- Working software over documentation
- Promote sustainable development
- Technical excellence and good design
- Simplicity is essential
- Self-organizing teams
- **Team reflection and adjustment**

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

### Retrospectives and Adjustment

#### Retrospectives

- cover two levels:
  - development level: evolution of the software
  - meta level: inspection of the process
- Attendance:
  - team leads
  - floating

#### Change people and roles

- many projects fail due to people in wrong positions

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcome changing requirements
- Deliver working software frequently
- Business and technical collaboration
- Trust
- Face-to-face communication
- Working software as the primary measure of progress
- Promote sustainable development
- Technical excellence and good design
- Simplicity is essential
- **Self-organizing teams**
- Team reflection and adjustment

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

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"

Principles behind the Agile Manifesto

- Early and continuous delivery of valuable software
- Welcoming product back
- Delivering working software frequently
- Business people and developers must work closely together throughout the project
- Trusting team members
- Face-to-face conversation is the most effective method of communication
- Working software is the primary measure of progress
- **Promote sustainable development**
- Technical excellence and good design
- Simplicity is essential
- Self-organizing teams
- Team reflection and adjustment

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

Promote Sustainable Development

- **Amount of work must match amount of time**
  - consider vacation and real time
- **Reviews and code inspection**
  - external
  - internal (by developers and scripts)
  - *active* marketing of concepts and interfaces

The Essence

- **Communication is the key**
  - Force feedback
- **Stability means death**
  - Be agile
- **Common sense**
  - Use your brain (and feel/smell the problems)
  
- **Just in time planning:**

*“Plans are worthless. Planning is essential.”*

Dwight D. Eisenhower

Many Thanks!

If you want to know more:

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

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

