

Introducing Agility into an Organization or: How to become Agile

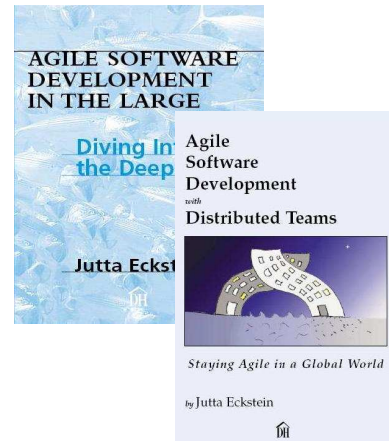
Jutta Eckstein
Agile Portugal 2010

Disclaimer

- **This talk is not about**
 - Introducing / explaining agile development
- **This talk is about**
 - Introducing agility to a project or to an organization

Jutta Eckstein

- **Independent coach and consultant**
- **Bachelor of education and teaching**
 - In „former life“
- **As engineer**
 - C++, Smalltalk, Java
 - Design, Patterns
 - Agile Processes
- **Project / process coach, project manager**
 - Finance, automobile, manufacturing, telecommunication



Two Typical Approaches

- **Guerilla tactic**
 - Self-organization
 - Establishing agile development practices
- **Supreme command**
 - Roles and responsibilities
 - E.g. steering and feedback by customer
 - Incorporation of whole organization
 - E.g. infrastructure, quality assurance

Successful Transition Impossible without Management

- **Customer involvement**
 - Steering through product owner
 - Highest business value first
- **Frequent delivery**
 - Small releases
 - Frequent valuable feedback

War Story: From Guerilla Tactic to Supreme Command

- **Small team started by itself**
 - Several problems occurred → later addressed
 - No separation between technical and business decisions
 - No self-organizing / self-responsible team
 - No clear business value
 - Daily wasting time
 - Acceptance within the team vanished
- **Acquired by new company**
 - Impressed about discipline, morale, transparency and delivery
 - Decision for overall application
 - Even in non-software areas

Command and Control Transitioning Unfeasible

- **Contradicts agility**
 - Trust is already lost right from the beginning
 - Self organizing team?
 - Development practices can hardly be dictated
 - E.g. pair programming, even unit testing

Commitment from Everyone Necessary

- **Management and customer awareness, by:**
 - Project situation, such as
 - Market demand, new domain, failure, frequent changes, ...
 - Reports, such as
 - Forrester research, Standish report, ...
 - Articles, talks
- **Developer awareness, by:**
 - Conferences, trainings
 - Cutting-edge
- **Everyone, by:**
 - Retrospective

Transitioning to Agility implies Change

- **Understanding change**

- Elisabeth Kübler-Ross:

- Denial
- Anger
- Bargaining
- Depression
- Acceptance

- Virginia Satir:

- Old status quo
- Foreign element
- Chaos
- Transforming idea
- Practice and integration
- New status quo

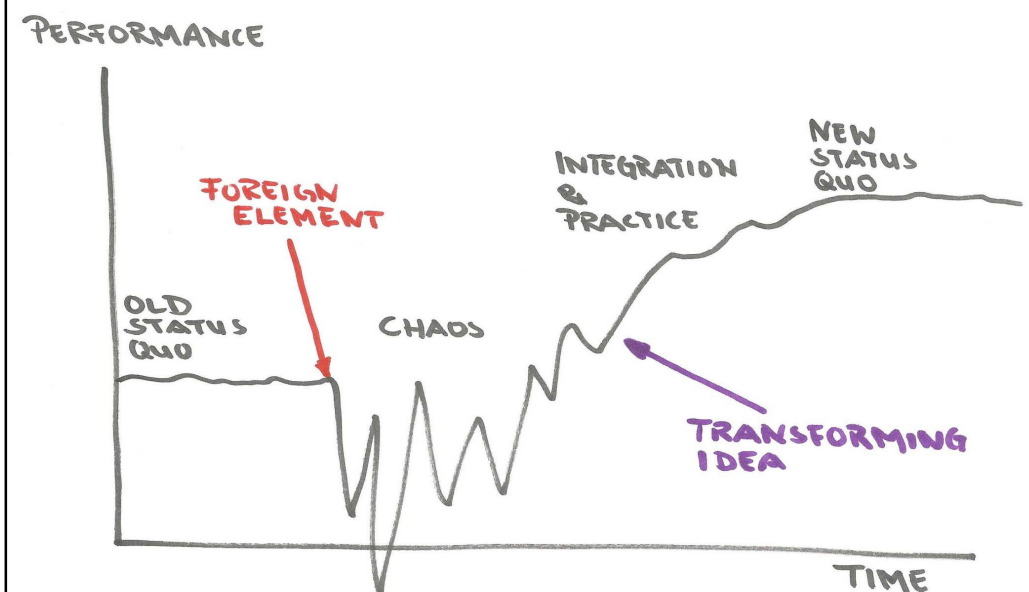
- **In general**

- Recommended reading:

- Linda Rising, Mary Lynn Manns: Fearless Change



Satir Change Model



Often: Pain Threshold supports Change

- **Typical quotes that will support the change:**
 - „Everything that we will do differently will be better..“
 - „We can do anything, except the things we're doing now..“
- **Pain threshold is typically reached jointly**
 - But often expressed / experienced differently:
 - Giving up
 - Cynicism
 - ...
- **But: Not everyone has the power and believe to start over**

War Story: Change by Pain Threshold

- **Project failure**
 - Large team
 - Long development time
 - No delivery
 - Customers upset
- **New start**
 - PM, process, technology
 - Planned measured iteration
 - Some *agile* practices
 - „Feedback circles“
 - One team / project / process

Yet: Pain Threshold often Eases Change only Superficial

- **Typical quotes that make it hard:**
 - „*This new idea is just old wine in new skin..*“
 - „*Sounds cool, but nothing will change anyway..*“
- **Generally:**
 - Skepticism is healthy and natural
 - Highly enthusiastic people can be difficult as well
 - **It's important that everyone is getting heard**

War Story: Pain Threshold created Superficial Support

- **Large team started new project (after failure)**
- **Team and management convinced of agile**
 - Eager to learn and apply
 - Few people convinced won't work (here)
 - In general claimed it's a good idea
 - Dogmatic
 - Special mentoring required

Pilot Project

- **Often: Wishful thinking**
- **Moreover: Cream of the Crop often not helpful**
 - Would have been successful anyway
 - No role model
 - No realistic (acceptable) conclusions
- **Start with a regular team**
 - Helpful if they are at least a bit keen on trying it
- **Start with a regular project**

War Story: Pilot Project

- **Large failed project**
- **New start with small team**
 - First two iterations whole team worked on toy project
 - Familiarized with technology, process and practices
 - Smooth start in real project on third iteration
 - Still supported by coach / mentor

Rough Schedule

- **Preparation**
 - Talk, book reading, ...
- **Retrospective**
- **„Readiness / Enabling“ Workshop**
- **Training**
 - Ideally: Customized for project
 - Result: The team's own process
- **Mentoring / coaching**
- **Sustaining change**

Supporting Activities

- **Offering**
 - Talks, articles, conferences, ...
- **For all roles**
 - Management
 - Customer
 - Developer
 - QA
 - ...
- **Three key roles:**
 - Project leader
 - Passionate change agent
 - Architect / technical leader

Retrospective

■ Reflect on:

- Either past project
- Or project so far

■ Advantage:

- No presentation of solutions the team is not aware of
- Team is developing their own solutions
 - Taking ownership
 - First step to self-organization
- Insight about necessary change
- Awareness of good practices

Readiness / Enabling Workshop

■ Workshop objective

- E.g. Basis for decision for introducing agility

■ Applicability in the organization

- What's done already?
- What's easy to apply?
- What's really difficult to apply?
- What's impossible to do?

■ Adjustments and process improvements

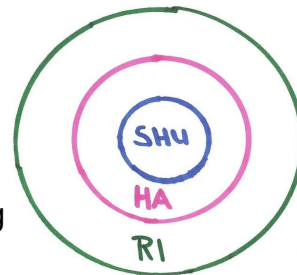
- What makes it difficult?
- How could we benefit from the solution?

■ Next steps

- Precise necessary actions

Steps in Learning

- **It is easier to start with a process out-of-the-box**
 - But: it is still important to understand the agile value system
 - Adjustments via regular retrospectives
- **Respect the learning process**
 - Shu: Copying techniques without modification
 - Ha: Reflection on meaning and purpose
 - Ri: Originally thinking and developing from background knowledge



[Brought into the agile field by Alistair Cockburn]

(Customized) Training

- **Education about agile development**
- **Setting up a concrete process**
 - When and where will the daily synchronization take place?
 - Develop your first product backlog (list of user stories)
 - Agree on roles and responsibilities
 - Estimate and prioritize first set of user stories
 - Agree on iteration length
 - Plan your first iteration
 - ...

War Story: Customized Training

- **Small highly motivated team**
 - Everyone had read / heard about agile
 - Everyone was convinced about it
- **Quick start with customized training**
 - Including first iteration plan
- **Passionate team member**
 - Helped whole team going through change

Mentoring / Coaching

- **Passionate change agent, who**
 - Sees the difficulties
 - States problems
 - Insists on finding solutions
 - Helps implementing solutions
 - Is trustworthy, competent and respected
- **It is helpful to empower someone inside the organization**
 - Someone who has been there, done that
 - You might need some support from outside at first

War Story: According to the Schedule

■ Maintenance „project“

- One week for process kick-off
 - Simulation, leveraging experiences
 - Shaping process by customized training
- A few months later review
 - Impressive improvements
 - Communication, transparency, whole team
 - Some struggles
 - Quality (legacy code...), estimates
- People who missed the kick-off were left behind
 - No buy-in for the process
 - Resolution: participated in kick-start of different team later

Sustaining Change

- **A few passionate people who care**
 - Till the new status quo is reached
- **Mentoring for newbies**
 - Newbies work with seasoned team members
 - Foster parents support newbies
- **Celebrate achievements**
 - Share experiences
- **Keep on learning**

Lessons Learned

- **Commitment from all sides necessary**
- **Change is never smooth**
 - You will have to go through chaos
- **Retrospective creates awareness**
 - Helps later on to continuously improve
- **Readiness / enabling workshop gives everyone a voice**
- **Customized training provides a jump-start**
- **Change agent provides overall support**

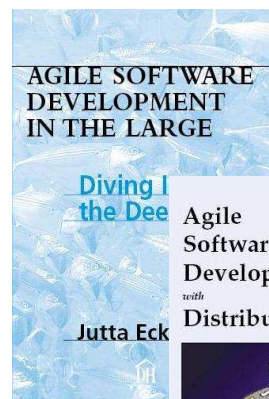
Many Thanks!

Contact information:

Jutta Eckstein

je@it-communication.com

www.it-communication.com



Agile
Software
Development
with
Distributed Teams



Staying Agile in a Global World

by Jutta Eckstein

