

SSSE Swiss Society of Systems Engineering

Implementing Continuous Improvement in product development

Blaise Rey-Mermet, EVOCEAN GmbH

Bio

Blaise Rey-Mermet

Dipl. Natw. ETHZ, Master MTE IMD
blaise.rey-mermet@evocean.com
www.twitter.com/blaiseoip

Product Development / Software Engineering

Lean & Agile Coach

Certified Scrum Master CSM

Kanban coach

Scaled Agile Framework Program Consultant SPC

Requirements Engineering Coach

Coaching:

Continuous Improvement
Lean & Agile Product Development
Requirements Management
Model Driven Development

Tools:



Workshops & Training:

Agile Requirements Management
Model Driven Development
UML and SysML

Three Key Take-Aways

- Add systematic and continuous improvement as part of each product development project
- Measure your improvements based on real facts
- Visualize your improvements to get fast feedback

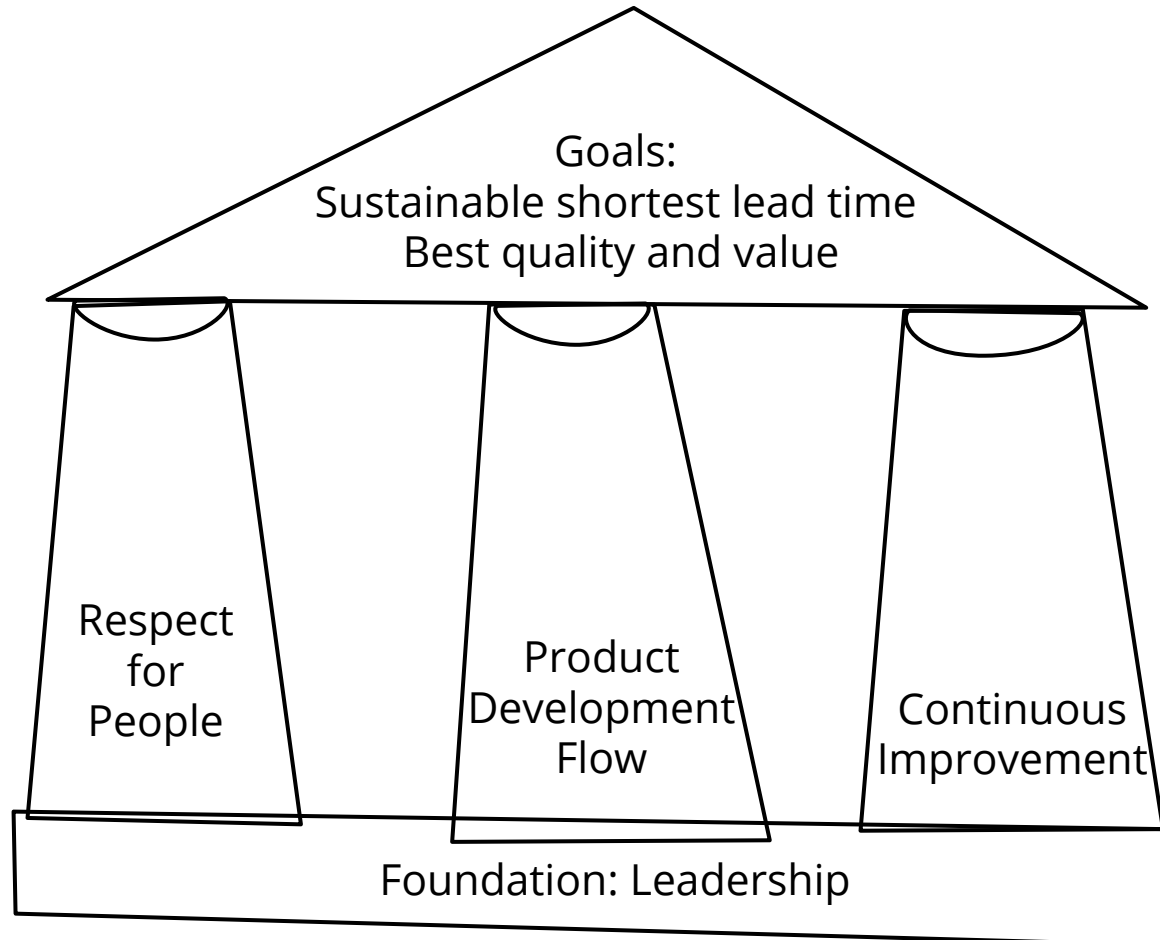
Take it
away

Goals:
Sustainable shortest lead time
Best product quality and value

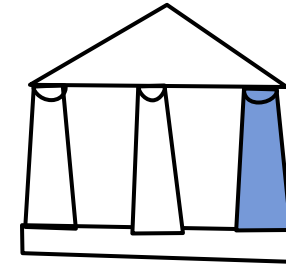
For that, we want to **establish** better practices,
not just define new ones.

Lean Development Core Values

Ref. **LEAN PRIMER** by Craig Larman and Bas Vodde Version 1.5



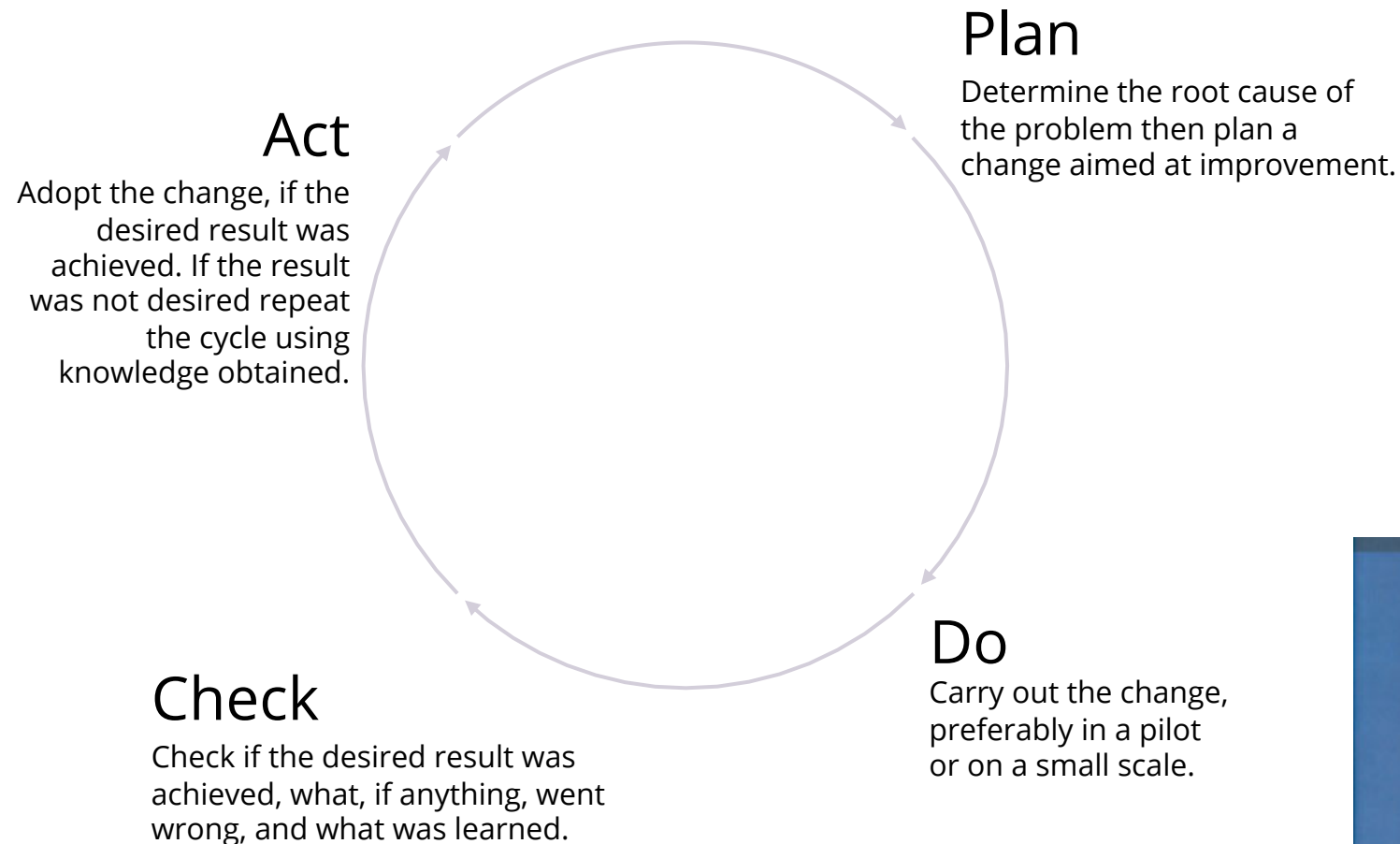
Establish a “Kaizen” culture



Kai	Zen
	
Change	Good

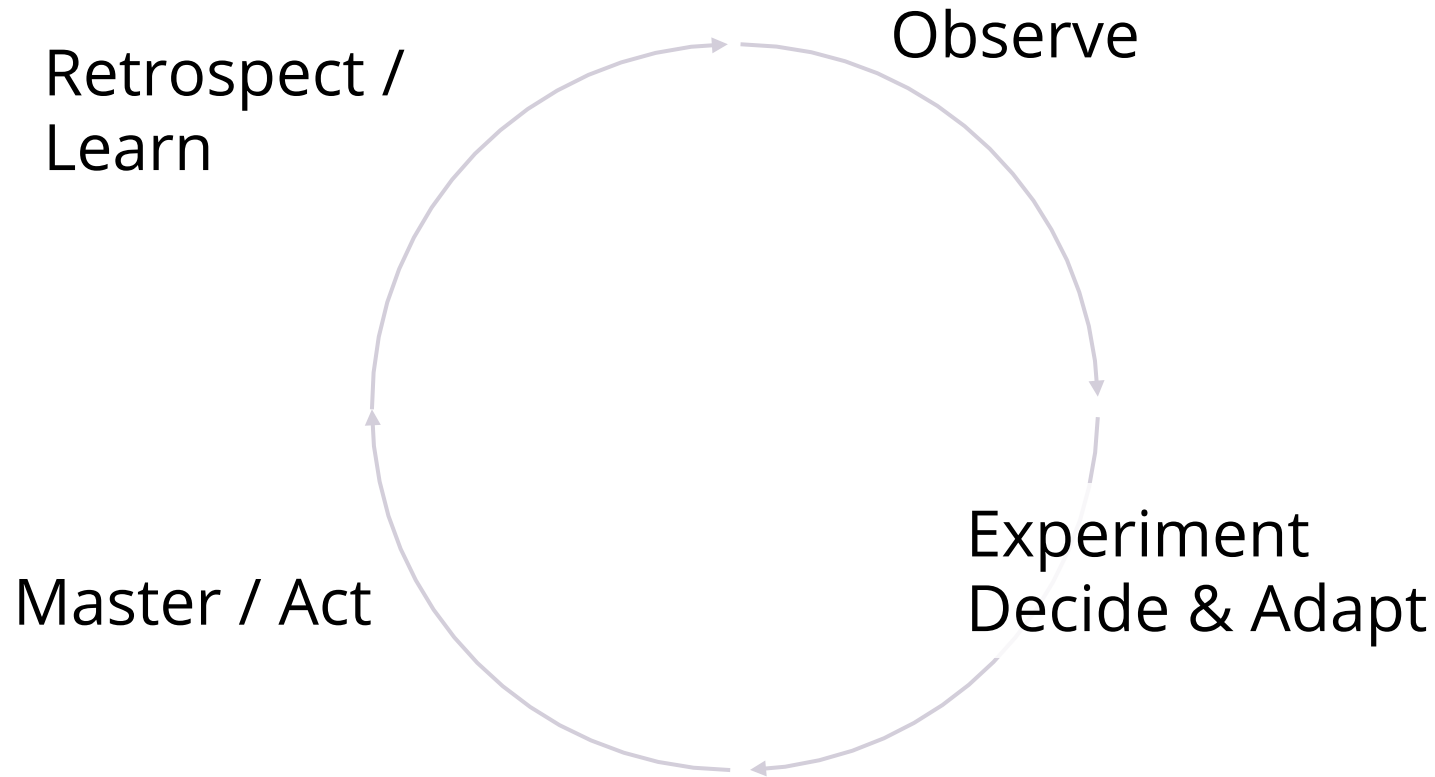
1. Choose and practice techniques the team has agreed to try, until they are well understood
2. Experiment until you find a better way
3. Repeat forever

PDCA Quality Cycle (E. Deming)



W. Edwards Deming

Another Model: Apply decentralized control principles,
let your people find the path



See also: Stephen Bungay - Back to
The Future, Business and
Organisation in the 21st Century

Continuous Improvement Phases: let your people find the path



Observe

- *Observe*, collect needs and motivations
- *Imagine* new solutions, collect good practices
- Formulate objectives

Experiment & Adapt

- Decide, choose a solution
- Draft a set of solutions, adapt it
- Pilot in your own projects.

Master

- Anchor the change in everyday life
- Tailor the practice to new projects
- Involve participants of the pilot to deploy
- Communicate new successes

Retrospect

- Learn
- Measure
- Visualize
- Compare the picture of where you are vs. where you want to be
- Find a better way

Observe

Visualize the team improvements

Kanban Practices (Lean Product Development)

Visualize Workflow

Limit Work-In-Progress (WIP)

Manage Flow

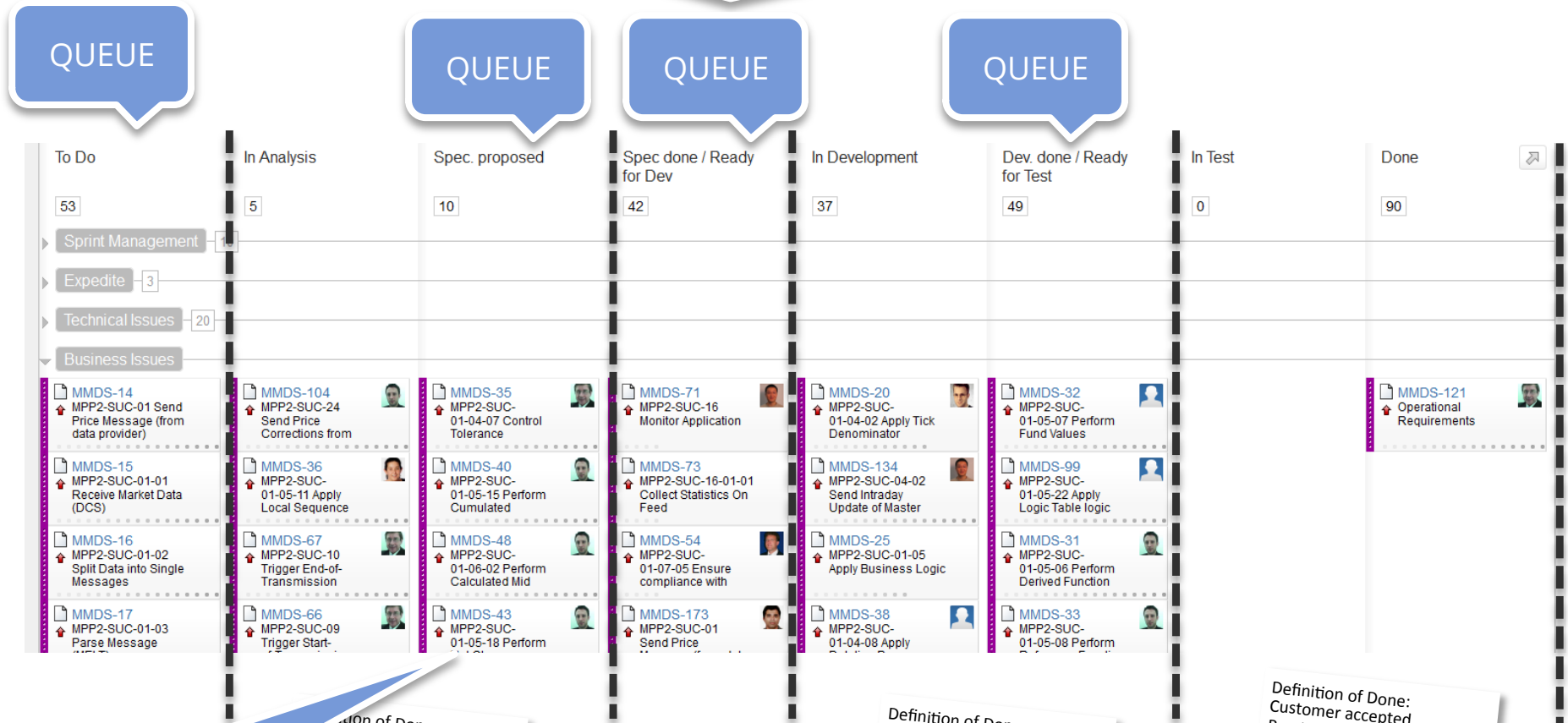
Make Process Policies Explicit

Improve Collaboratively (using models & scientific method)

Ref: Kanban – Successful Evolutionary Change for Your Technology Business
by David J. Anderson (Blue Hole Press 2010)

Visualize the process flow as Kanban board

Visualize the development process as a flow →



Visualize who is working on which task

Definition of Done:
Code clean & checked in on trunk
Integrated & regression tested
Running on UAT environment

Definition of Done:
Customer accepted
Ready for production

Visualize the process flow as Kanban board

Limit Work-In-Progress

Stream 1 - Kanban Board ... ?

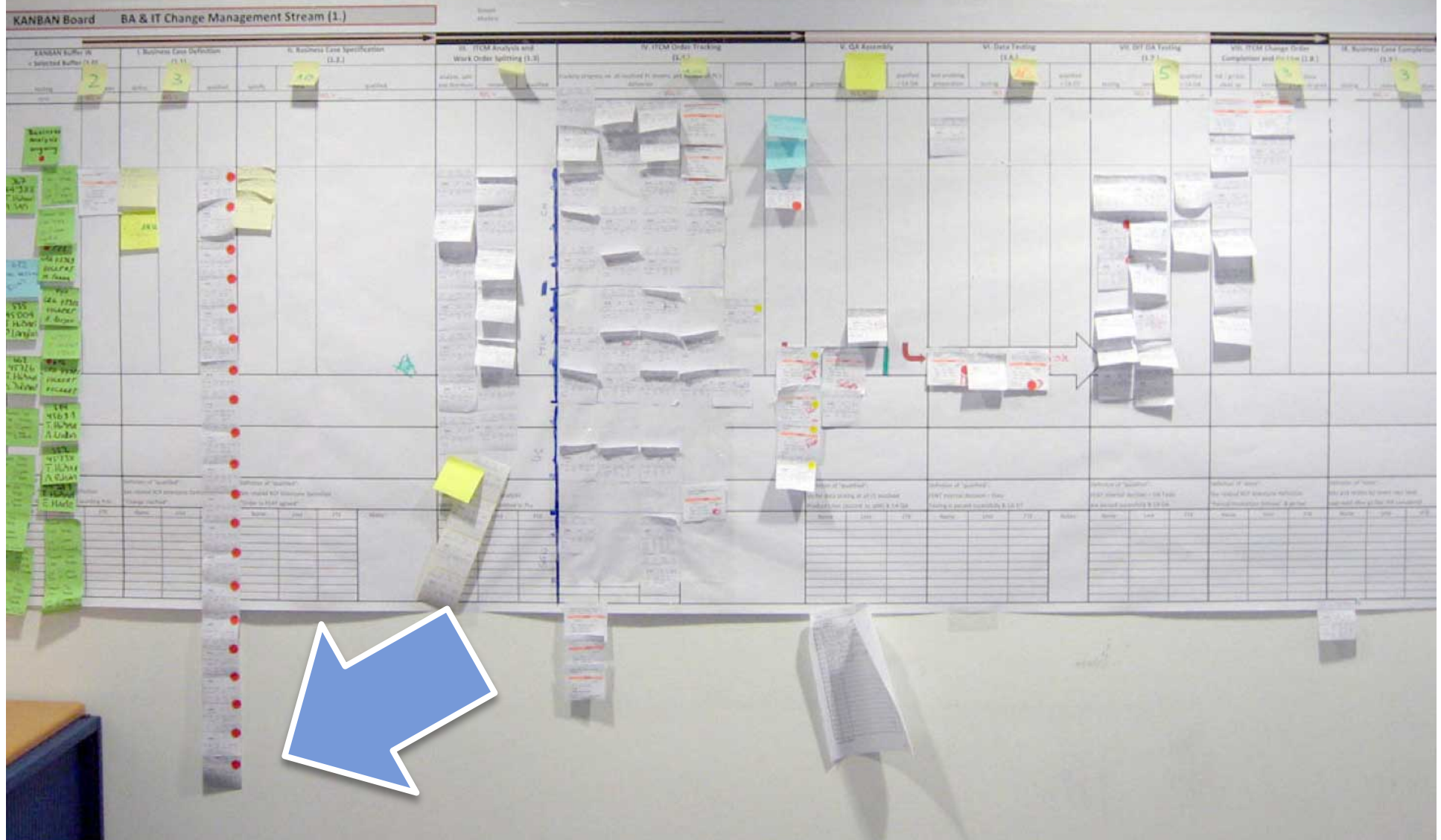
focus board view timeline

Tags / State < To Do (Open) (4) < Planned (In Analysis, Plann(5) > Spec. p(6) < Ready for Dev. (7) < In Development (6 of 5) > Ready (2) > In Testing

Business Issue (30)

<p>220 Business Issue RD-268 Review the understanding and feasibility of 0h</p>	<p>227 Business Issue SD-UC-013 Process a code erase request 0h T 1/1</p>	<p>240 Business Issue SD-UC-004-001 Initial assignment 0h T 1/1</p>	<p>247 Business Issue SD-UC-602 Initialize the snapshot save 0h T 2/2</p>
<p>223 Business Issue SD-UC-612 Automatic switch to another SD instance 0h</p>	<p>228 Business Issue SD-324 Write UC-013 process code erase request 0h</p>	<p>241 Business Issue SD-140 update static data maintainer 0h</p>	<p>248 Business Issue SD-239 Implement the snapshot save initialization 0h</p>
<p>221 Business Issue SD-UC-001-002 Enable high-availability on Inbound 0h</p>	<p>229 Business Issue SD-UC-010-002 Prevent publication of old prices in the 0h</p>	<p>242 Business Issue SD-UC-603 Update the snapshot with intradays 0h</p>	<p>249 Business Issue SD-UC-004-002 Initial Assignment of Listing per save 0h T 1/1</p>
<p>226 Business Issue SD-SB-001-003 Additional administration screens 0h</p>	<p>219 Business Issue SD-UC-001-001 Improve Inbound Stream Performance 0h T 2/2</p>	<p>243 Business Issue SD-UC-605-001 Calculate codes with manual marks 0h</p>	<p>250 Business Issue SD-266 Update maintainer - processing messages 0h</p>
	<p>230 Business Issue SD-UC-302-001 Inprove performance of HPS streams 0h T 2/2</p>	<p>244 Business Issue SD-UC-004-003 Process the manual marks 0h</p>	<p>251 Business Issue SD-UC-305-001 Check for non-existing listings 0h</p>
		<p>245 Business Issue SD-UC-004-004 Process the value multiplier 0h</p>	<p>252 Business Issue SD-UC-005-002 Intraday data snapshots 0h</p>

Visualize the process flow as Kanban board



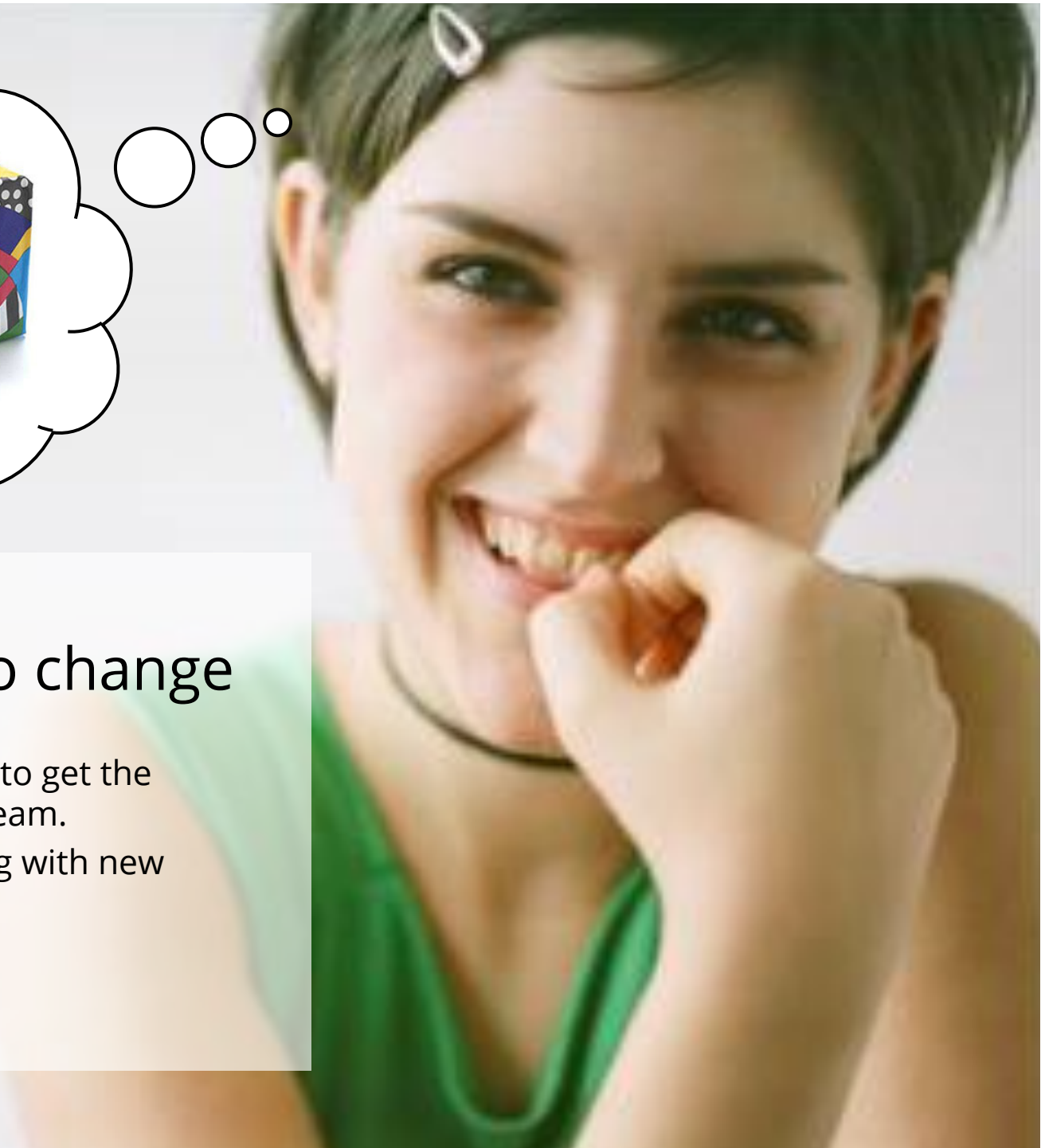
Desire,
not urgency.



Replace urgency with the **desire** to change

Urgency will not be sufficient to get the attention and energy of the team.

- **Learning** and experimenting with new techniques are fun
- **Focus on what is important** (not only what is urgent)



A model of (10 intrinsic) Desires (Jürgen Appelo)

- Curiosity
- Honor
- Acceptance
- Mastery / Competence
- Power
- Freedom / Independence / Autonomy
- Relatedness / Social Contact
- Order
- Goal / Idealism / Purpose
- Status

The need to think
Being loyal to a group
The need for approval
The need to feel capable
The need for influence of will
Being an individual
The need for friends
Or stable environments
The need for purpose
The need for social standing



Ref. Jürgen Appelo – How to Change the World

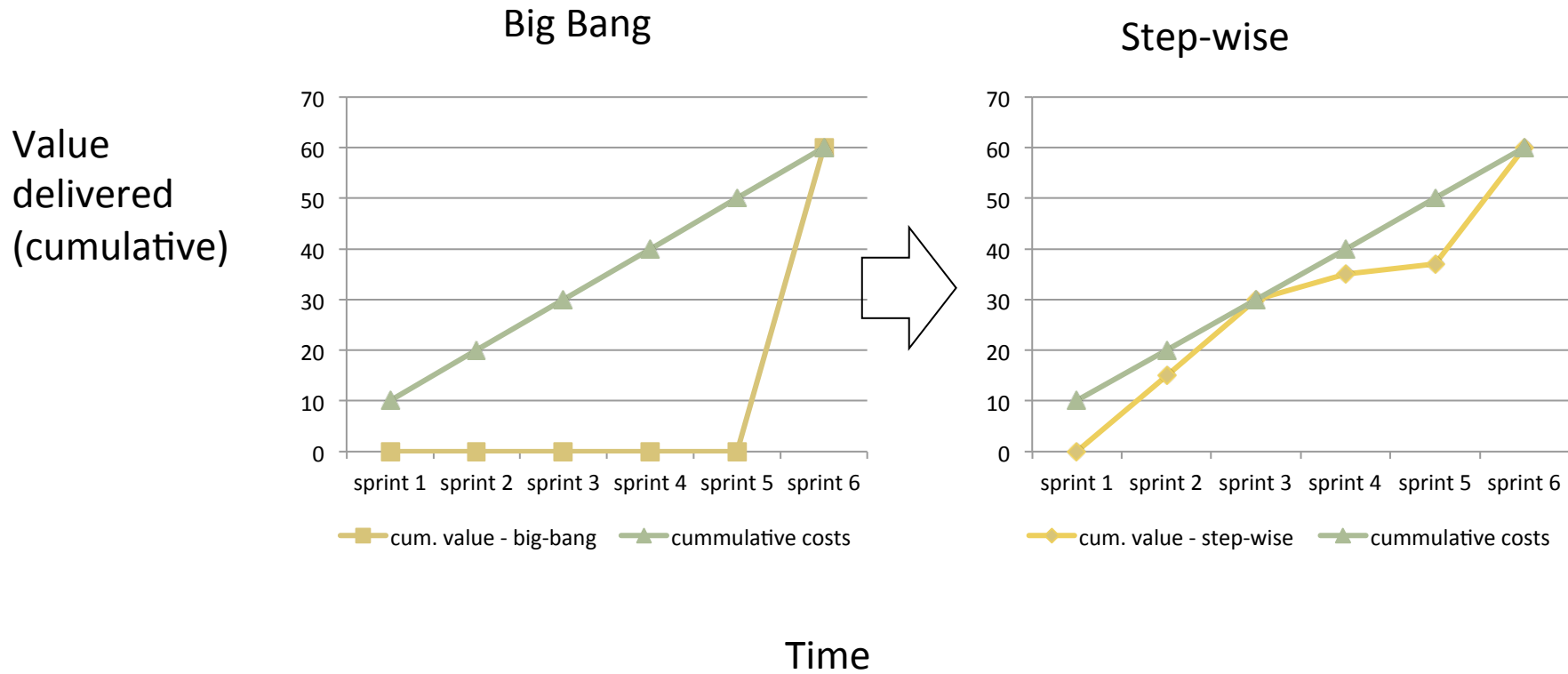
Improve in small steps

Define realistic objectives

Too ambitious goals will not be reached and will end in frustration.

Changes always result in a temporary drop of performance. Big bang changes are either rejected or will kill the work.

Big Bang = Higher Costs



Group improvements in a series of small steps.

Improvements (Example)

21. ...

22. Each project names a responsible for the requirements (either a requirement engineer or a product owner).

23. The requirements are documented in a single list or repository for the project.

24. Each requirement is uniquely identifiable, e.g. with a unique ID.

25. Requirements are discussed with the stakeholders and the team to achieve common understanding.

26. ...

Improvement Backlog (Example):

Improvement 01:
Requirements Development

Improvement 02:
Change Management

Improvement 03:
Configuration Management

**Improvement 04:
Requirements Management I**

Improvement 05: Monitor / Measure
progresses progresses

Improvement 06:
Requirement Mgmt. II

Improvement 07:
Retrospectives

Implement the most valuable/desirable improvements first

Improvement Backlog (Example):

Improvement 01: Clarify process, work products, and roles	<p>Higher Value, Most desirable</p> <p>Lowest Value, Less desirable</p>
Improvement 02: Scrum: clarify planning and estimation	
Improvement 03: Formulate Business Requirements	
Improvement 04: Formulate Specifications	
Improvement 05: Manage Business Requirements	
Improvement 06: Model the req. types and adapt ALM tool	
Improvement 07: Establish Requirement Traceability	
Improvement 08: Scrum: establish Scrum systematically	
Improvement 09: Visualize the workflow	

Address Improvements during the development sprints.

Legend: ● Observe ● Experiment ● Master ● Retrospect

Improvement Backlog (Example):



To Do	In Analysis	Spec. proposed	Spec done / Ready for Dev	In Development	Dev. done / Ready for Test	In Test	Done
57	5	10	36	37	53	0	94
Sprint Management MMDS-287 Review of Requirements for next Sprint MMDS-282 Prepare Sprint Planning for Sprint 20 MMDS-283 ALM Defect Management MMDS-284 Sprint Management MMDS-285 Scrum Management MMDS-286 Sprint Retrospect Meeting							
Technical Issues MMDS-185 Performance Analysis III MMDS-279 Performance MMDS-280 Support for BC 600 MMDS-281 Enable MMDS-298 Implementation of Hot-Hot for Scoach MMDS-300 Design for restart in hot-hot of 1 instance and sync MMDS-165 MMDS-276 Performance MMDS-277 Performance MMDS-278 Performance MMDS-332 Performance							
Business Issues MMDS-35 MPP2-SUC-01-04-07 Control Tolerance MMDS-71 MPP2-SUC-16 Monitor Application MMDS-20 MPP2-SUC-01-04-02 Apply Tick Denominator MMDS-134 MPP2-SUC-04-02 Send Intraday Update of Master MMDS-127 MPP2-SUC-01 Send Price Message - Alternate Flow 6 MMDS-54 MPP2-SUC-01-07-05 Ensure compliance with MDF2 MMDS-25 MPP2-SUC-329 Apply MMDS-330 Apply MMDS-37 MPP2-SUC-01-05-12 Apply Location Number MMDS-133 MMDS-145 *Need to MMDS-71 MMDS-173 MPP2-SUC-01 Send Price Message (from data provider) - MMDS-329 Apply MMDS-25 MMDS-331 Apply MMDS-25 MMDS-71 MMDS-291 2 BUS MMDS-127 MMDS-334 Handling MMDS-71 MMDS-128 MPP2-SUC-17 Request Start-of-Day Price Information MMDS-128 MMDS-246 Adapt MMDS-71 MMDS-288 1.1 MPP MMDS-289 1.2 MPP MMDS-290 1.3 MPP MMDS-174 MPP2-SUC-01 Send Price Message (from data provider) - MMDS-38 MPP2-SUC-01-04-08 Apply Dablation Process Flag							
Continuous Improvement MMDS-301 (T17 QA) Improve Defect Management Process MMDS-302 (T17) MMDS-303 (T17) MMDS-305 (T17) MMDS-307 (T04 Formulate sw specs) Coaching MMDS-308 (T04) MMDS-309 (T04) MMDS-310 (T18 Improve Config Mgmt.) MMDS-301 MMDS-306 (T17) MMDS-310 MMDS-311 (T18 Imp...) MMDS-312 (T18 Imp...) MMDS-324 (T18 Imp...) MMDS-301 MMDS-304 (T17) MMDS-315 MMDS-319 (T08) MMDS-320 MMDS-322 (T12 Lim...) MMDS-323 (T12 Lim...)							



Sprint Management Tasks

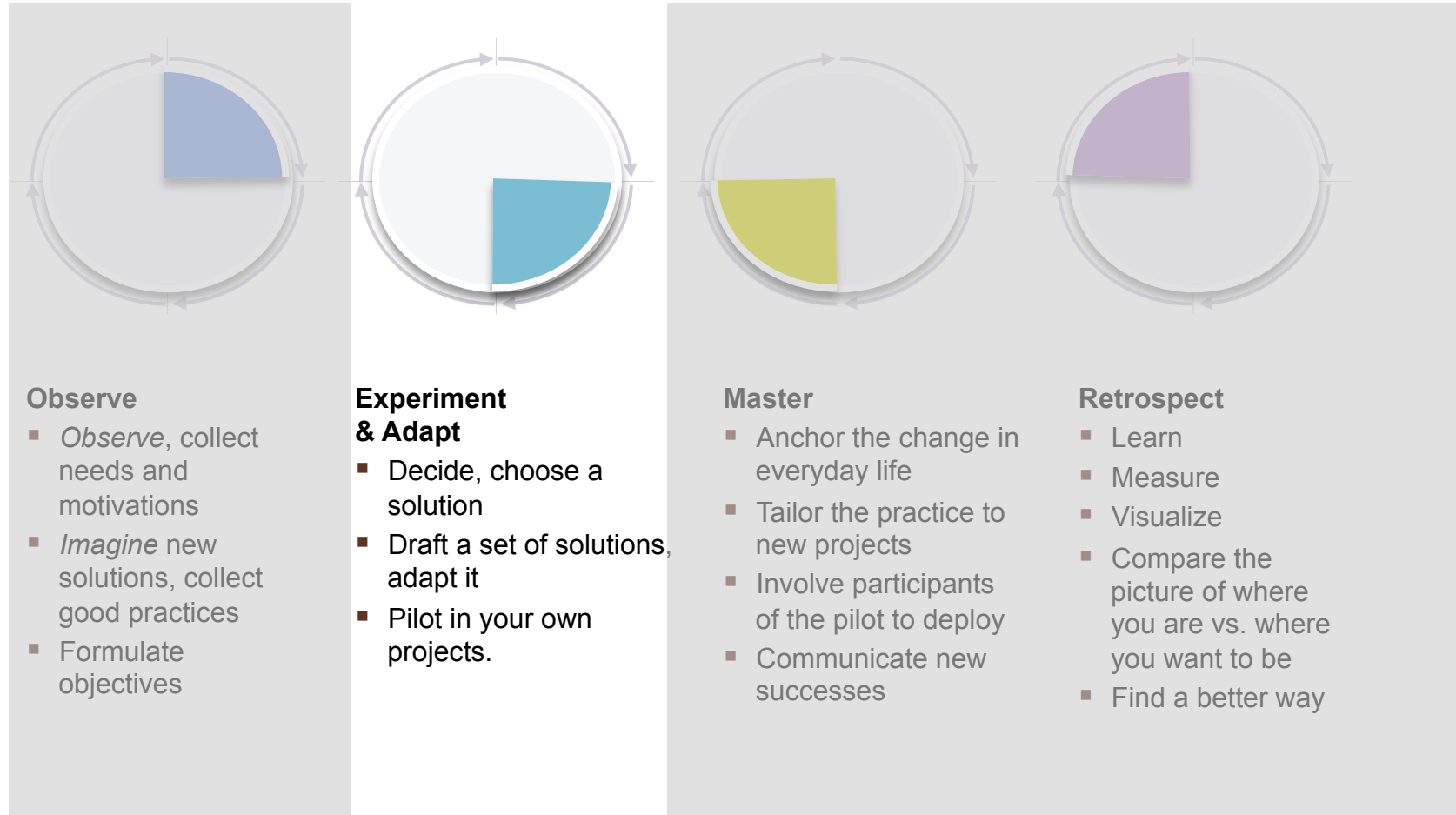
Architecture & Technical Tasks

Implementing Product Requirements

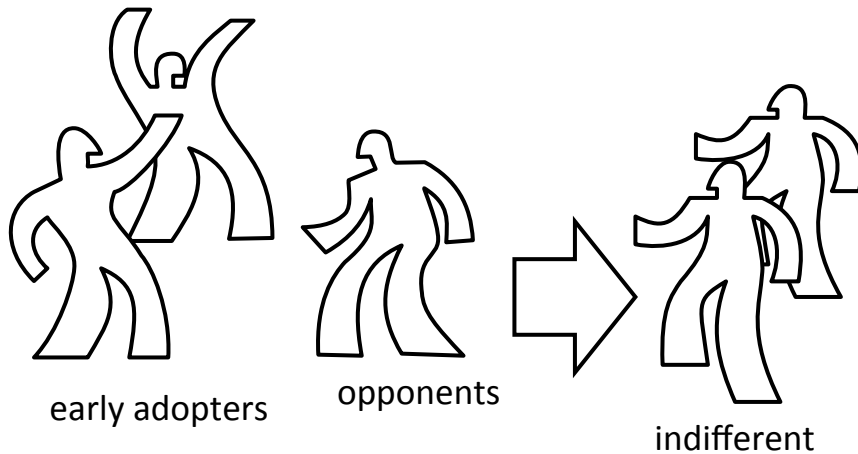
Continuous Improvement Tasks

Kanban board with continuous improvement tasks

Continuous Improvement Phases: Let your people find the path

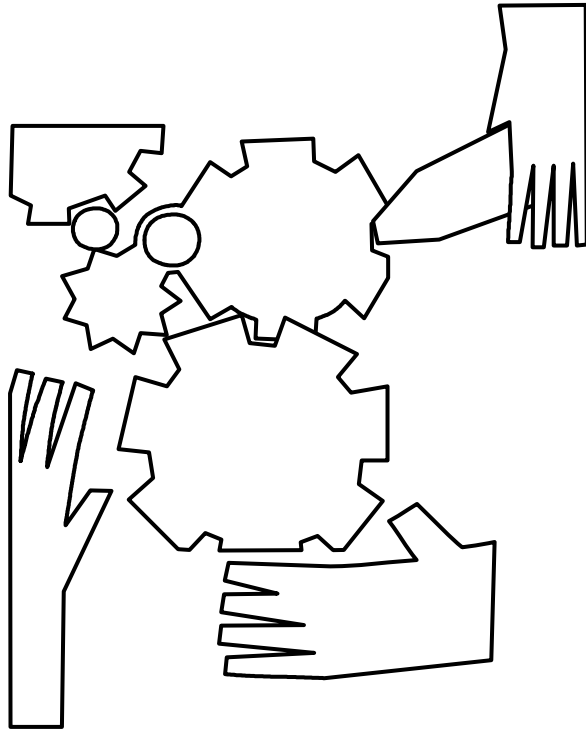


Paradoxical strategy: Rely on opponents



- Change creates 3 types of attitudes:
 - early adopters,
 - the indifferent that do not care, and
 - opponents.
- Find out about the personal motivation of the opponents and turn them into change leaders.

Engage yourself personally

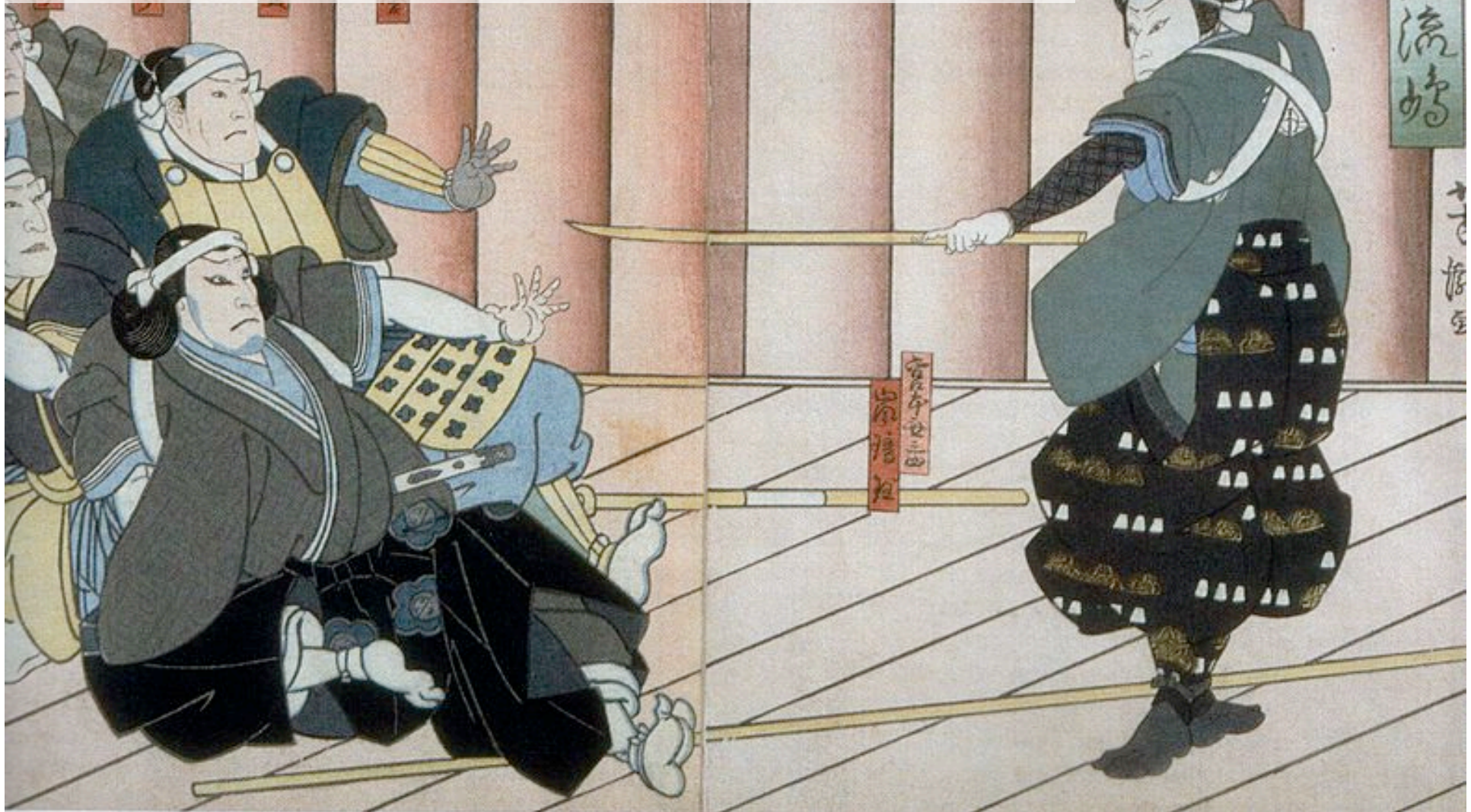


- **Be personal.**
State what your personal motivation is.
- **Be a first mover.**
Promote and lead the change.
- **Be curious** to get feedback
(go and see by yourself)
- Help to solve problems.

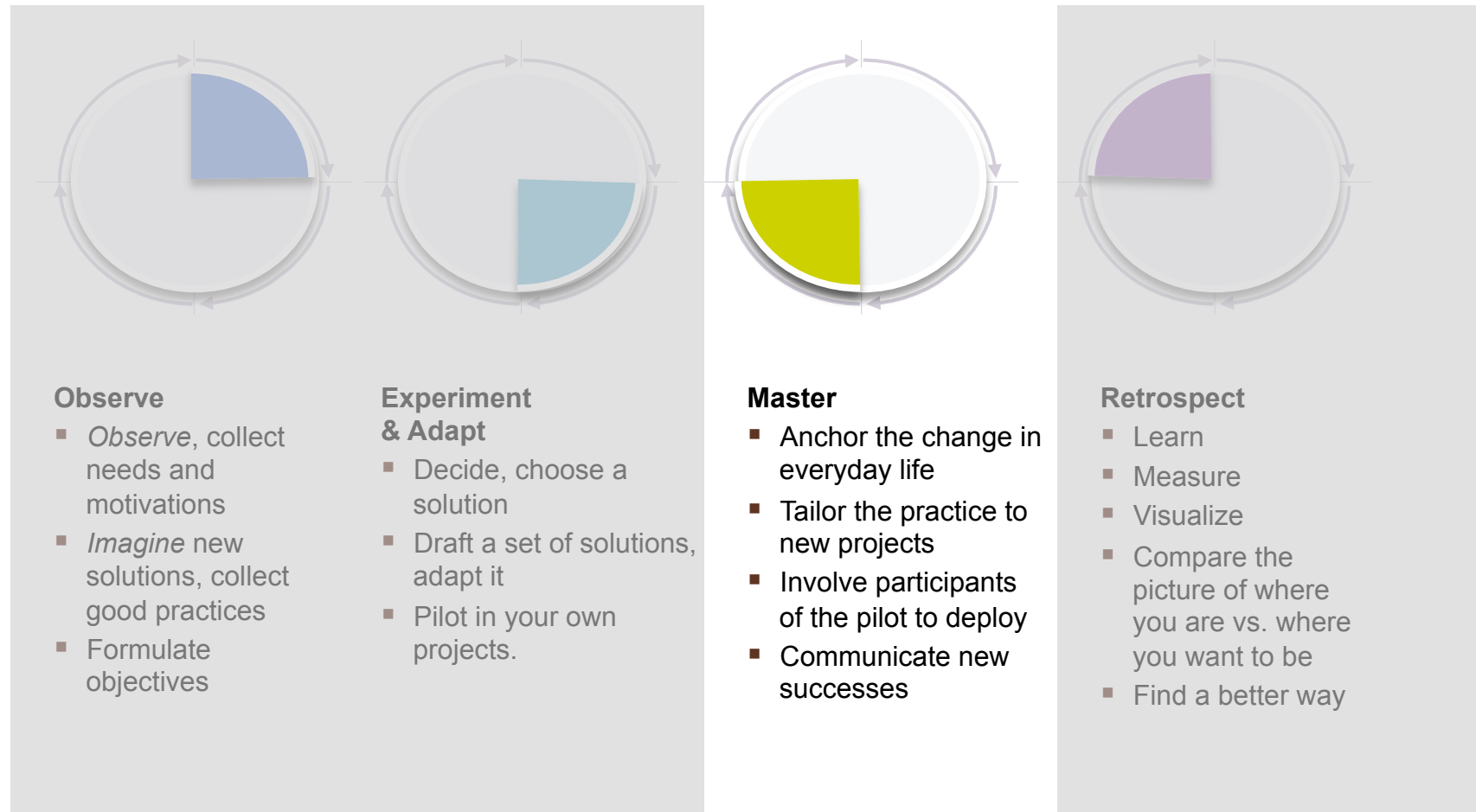
Do not develop an attachment to any one weapon or any one school of fighting

The Book Of Five Rings - Miyamoto Musashi (c. 1584 – June 13, 1645),
a Japanese swordsman and rōnin.

Source: Kanban and Scrum - making the most of both H.Kniberg & M.Skarin



Continuous Improvement Phases: Let your people find the path



Start simple... adapt continuously as needed

Definition of Done (Sprint 1)

- No one breaks the build!

Definition of Done (Sprint 21)

Code

- Project builds (Hudson)
- Unit tests provided + succeeded
- Code is documented (Doxygen)
- Code is checked into the configuration management system
- Component Integration Test passed
- Release Notes are documented

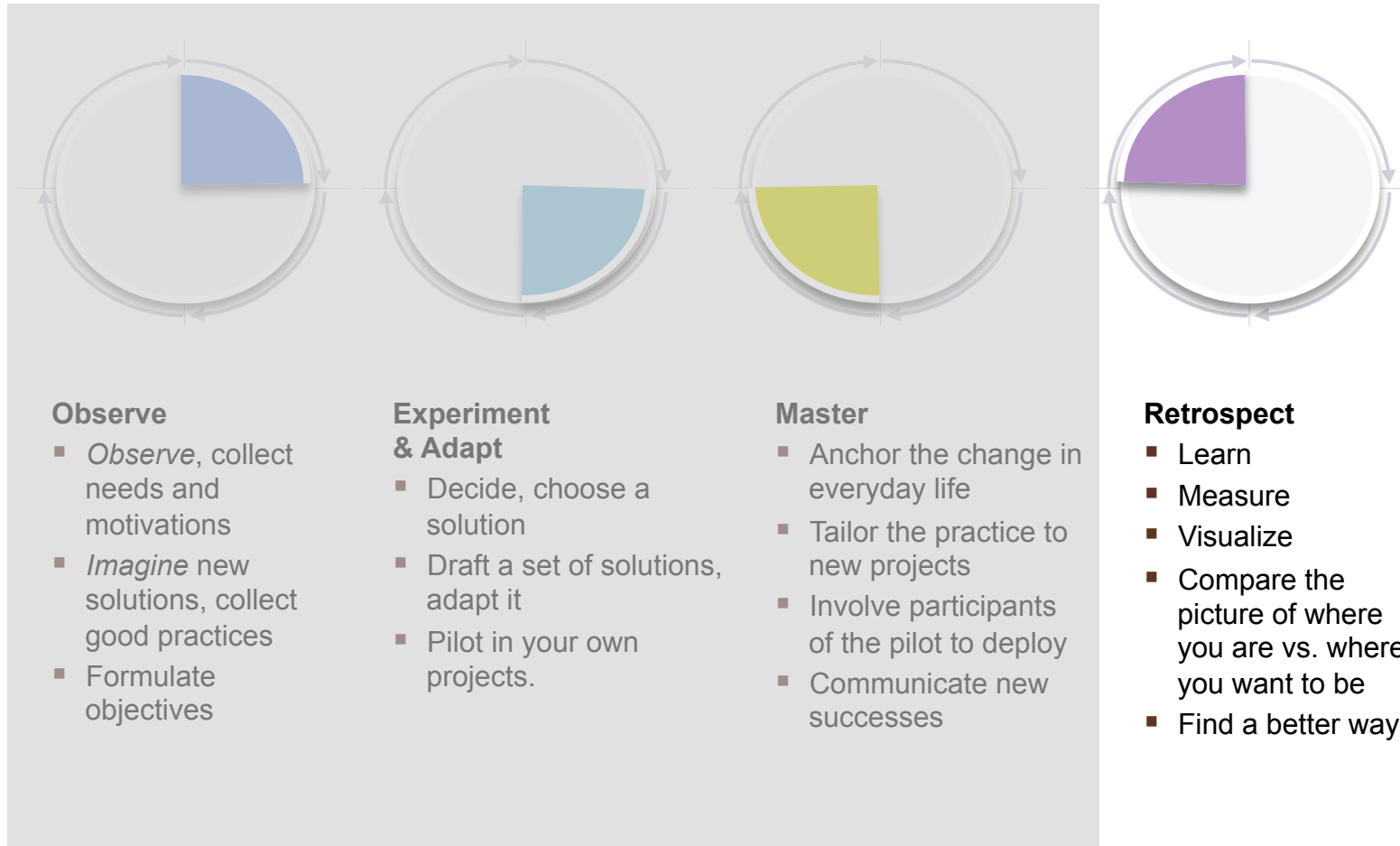
Documentation

- Reviewed, updated and approved
- Product Requirements reviewed and approved
- Doc. managed in the RE tool and traced to Business Reqs.

Test Artifacts

- Test Cases are documented, traced to the reqs. and reviewed.
- Test Results are documented in the QM Tool

Continuous Improvement Phases: Let your people find the path

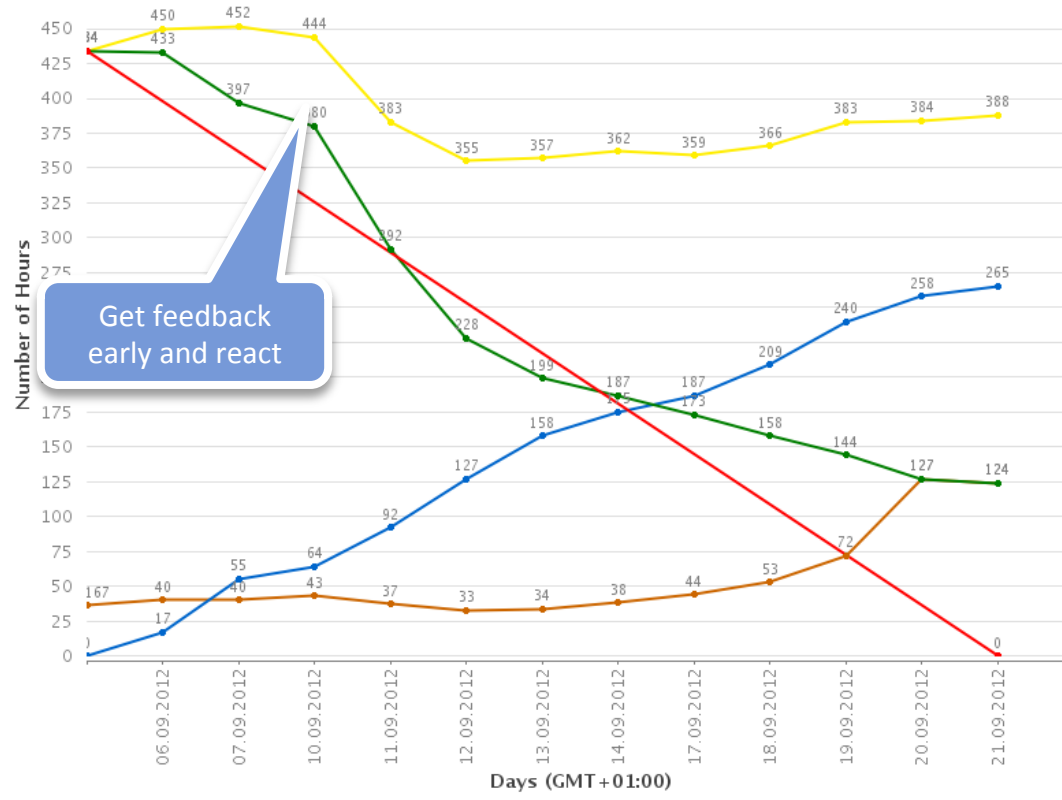


Get fast feedback

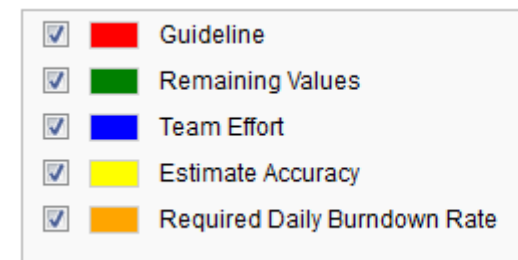
Measure the team improvements

Measuring Improvements

Example: Typical Burn Down Chart

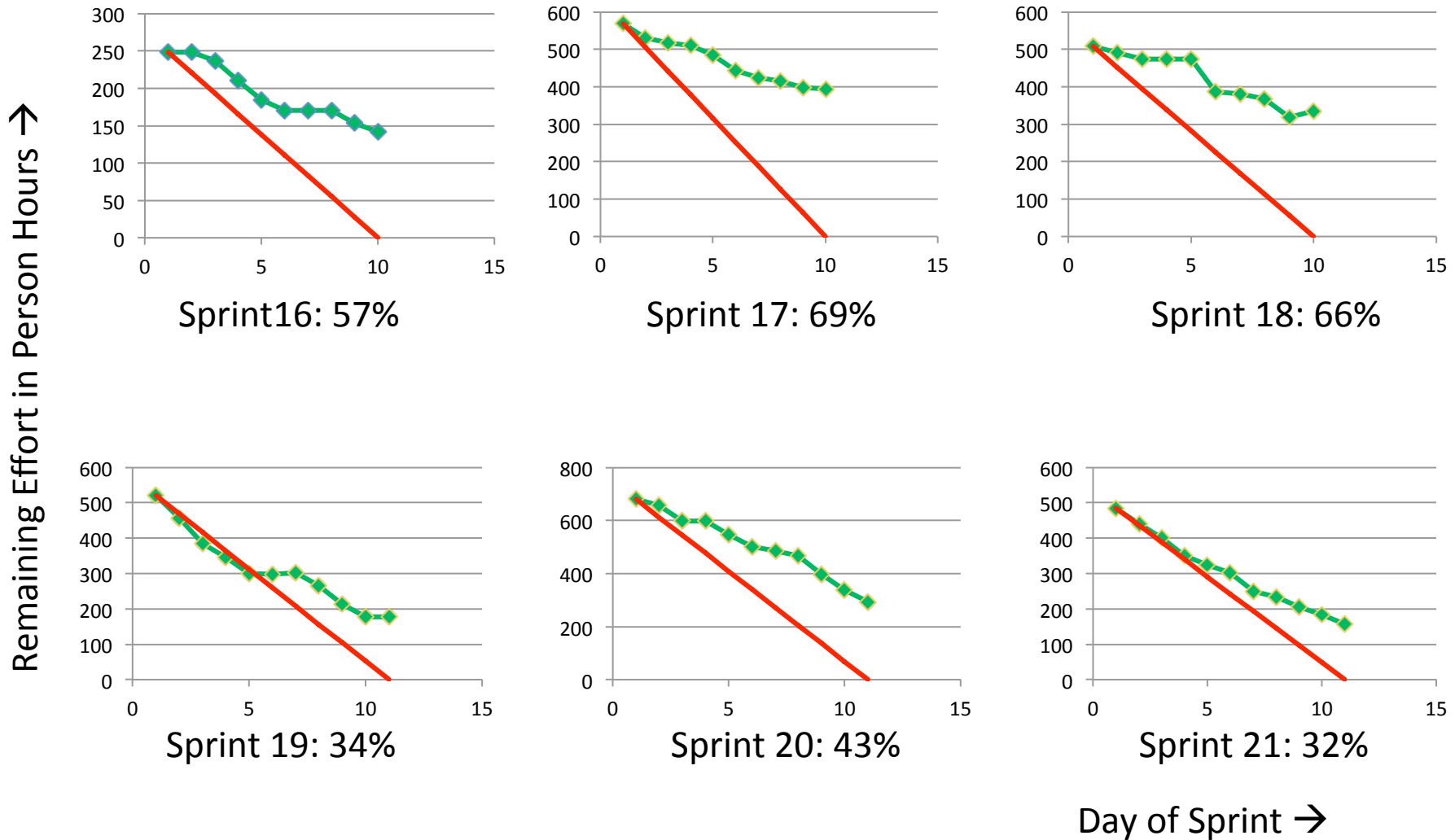


Get feedback early and react



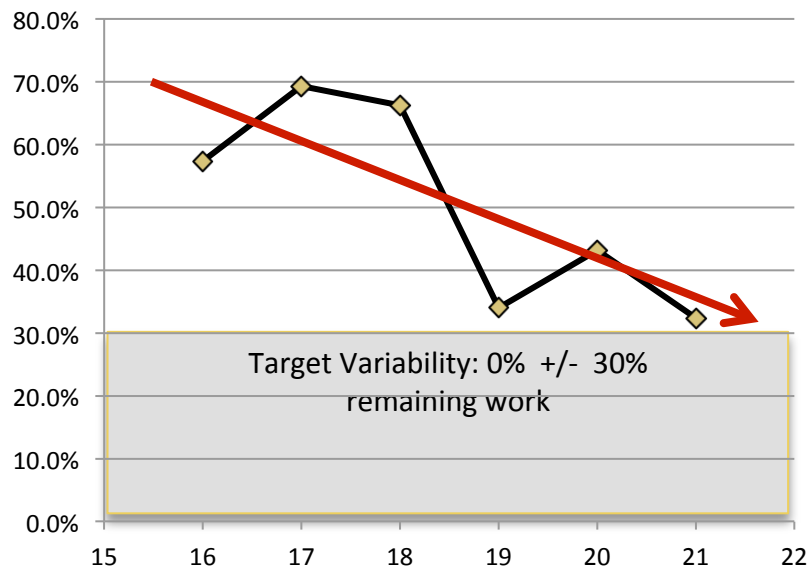
Measuring Improvements

Example: Manage Variability (of Burn Down Charts)



Measuring Improvements

Example: Manage Variability (of Burn Down Charts)



Huge step forward:

- The team can now **systematically** plan and track issues.
- The team **constantly** makes progresses

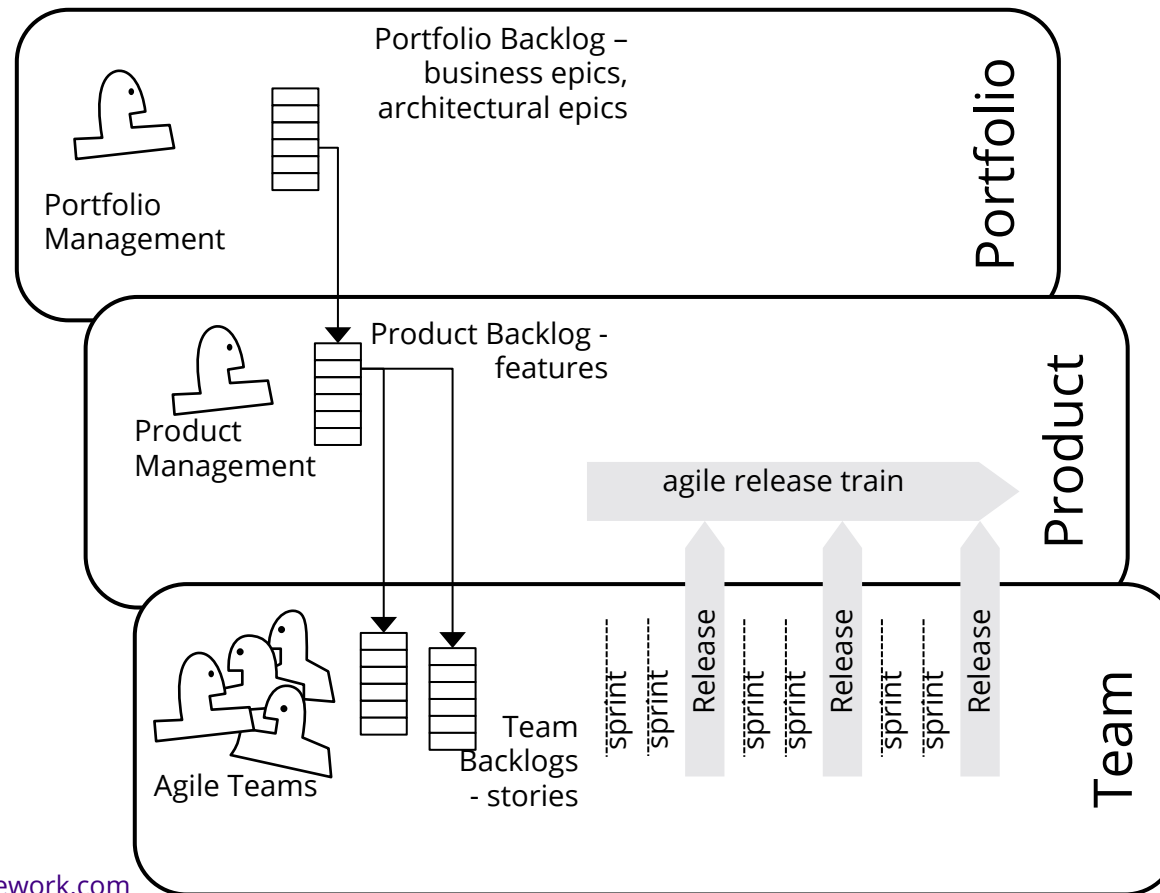
Challenges:

- The team velocity remains **too slow**. The root cause is the **complexity**.
- We still need to **improve maturity** in engineering.

Synchronize the team improvements
using a scaled approach

An example of Framework: The Scaled Agile Framework®

The Scaled Agile Framework is an online Framework for applying Lean and Agile practices at enterprise scale.

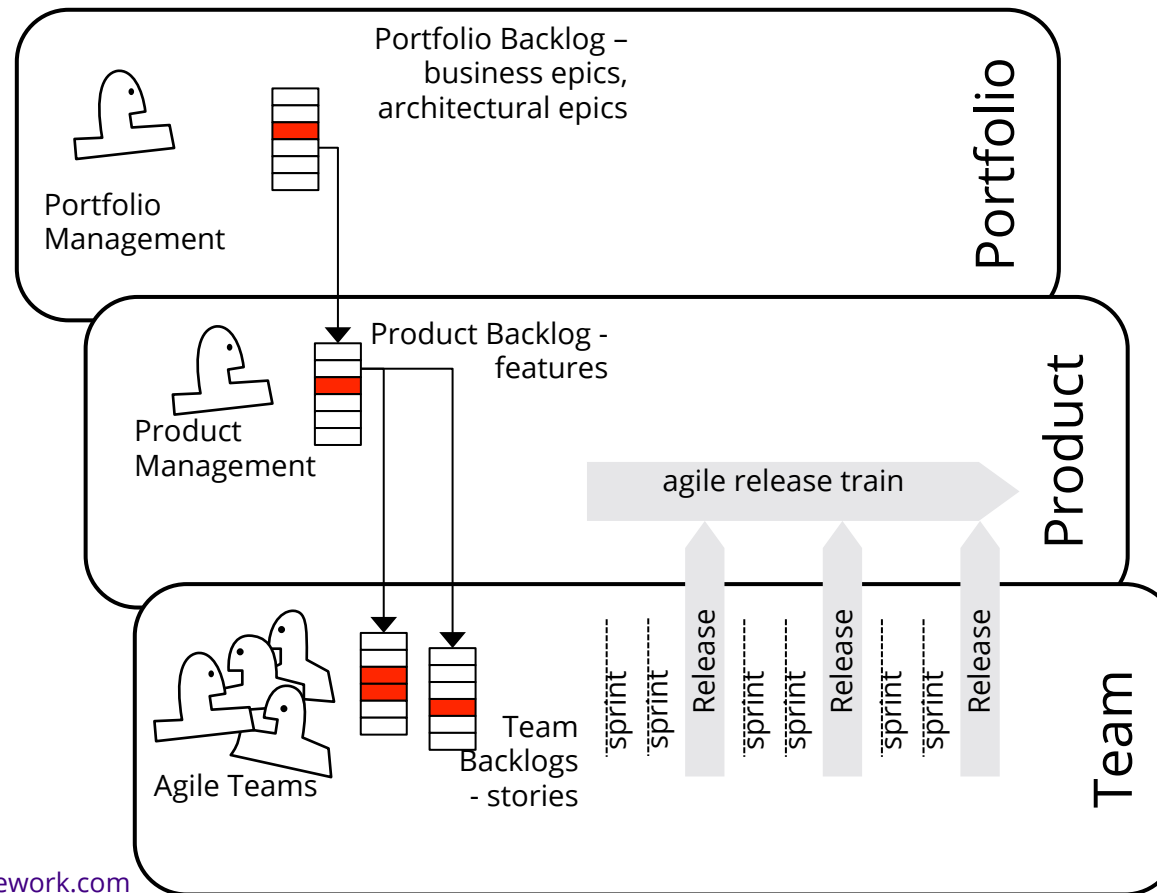


<http://ScaledAgileFramework.com>

Scaled Agile Framework® is a trademark of Leffingwell LLC

Synchronize Improvements at all levels.

The Scaled Agile Framework is an online Framework for applying Lean and Agile practices at enterprise scale.



<http://ScaledAgileFramework.com>

Scaled Agile Framework® is a trademark of Leffingwell LLC

Three Key Take-Aways

- Add systematic and continuous improvement as part of **each product development project**
- **Measure** your improvements based on real facts
- **Visualize** your improvements to get fast feedback

Take it
away

Change Management Reading List

- **The Heart of Change: Real-Life Stories of How People Change Their Organizations**
by John P. Kotter and Dan S. Cohen (Nov 6, 2012)
- **Our Iceberg Is Melting: Changing and Succeeding Under Any Conditions**
by John Kotter, Holger Rathgeber, Peter Mueller and Spencer Johnson (Sep 5, 2006)
- **Du Désir au plaisir de changer : Le coaching du changement**
de Françoise Kourilsky et Paul Watzlawick (5 mars 2008)
- **Fearless Change: Patterns for Introducing New Ideas**
by Mary Lynn Manns Ph.D. and Linda Rising Ph.D. (Sep 4, 2004)
- **How to Change the World** by Jurgen Appelo Apr.2012
<http://www.management30.com/product/how-to-change-the-world/>
- **Lean Change Management - A feedback-driven approach to change management**
by Jason Little <https://leanpub.com/leanchange>
- **Succeeding with Agile: Software Development Using Scrum**
by Mike Cohn (Nov 5, 2009)

Lean / Kanban Reading List

- **Lean Software Development: An Agile Toolkit**
by Mary and Tom Poppendieck (Addison Wesley 2006)
- **Kanban – Successful Evolutionary Change for Your Technology Business**
by David J. Anderson (Blue Hole Press 2010)
- **Lean Primer**
by Craig Larman and Bas Vodde. <http://www.leanprimer.com>
- **The principle of Product Development Flow. - Second Generation Lean Product Development**
by Donald. G. Reinertsen. Celeritas Publishing 2009
- **The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses...**
by Eric Ries (Sep 13, 2011)
- **Toyota Production System: Beyond Large-Scale Production**
by Taiichi Ohno and Norman Bodek (Mar 1, 1988)

Thank you!



Blaise Rey-Mermet
Principal Consultant
Dipl. Natw. ETHZ, Master MTE IMD
Scaled Agile Framework Program Consultant (SPC)
Mobile: +41 79 276 64 19
Email: blaise.rey-mermet@evocean.ch

Keep in touch: <http://www.evocean.com/improvement>
 <http://www.twitter.com/blaiseoip>

EVOCEAN GmbH | Grundstrasse 8 | CH-6343 Rotkreuz | Tel + 41 41 790 78 88 | [info \(at\) evocean.ch](mailto:info@evocean.ch)
EVOCEAN GmbH | Fischhof 3/6 | A-1010 Wien | Tel + 43 1 740 40 8035 | [info \(at\) evocean.at](mailto:info@evocean.at)
EVOCEAN Deutschland GmbH | Karlstrasse 35 | D-80333 München | Tel + 49 89 452 352 191 | [info \(at\) evocean.de](mailto:info@evocean.de)