How to Improve the Improvement Process

Karol Frühauf
INFOGEM AG
CH-5401 Baden
Karol.Fruehauf@infogem.ch

Chair of the ICT Process Improvement and Assessment Track
Who I am?

1975  MSc in Electrical Engineering / Technical Informatics, RWTH in Aachen, Germany
1975  Brown Boveri & Cie, Power system control programmer, project leader, quality manager, manager
1987  Co-founder of INFOGEM AG, consultant quality management, project management, reviews and testing, configuration management, metrics

sports  table tennis, tennis, skiing
arts     literature, theatre

Co-author of two books
    'Software-Projektmanagement und –Qualitätssicherung' and
    'Software-Prüfung – eine Anleitung zum Test und zur Inspektion'

... a sporty software engineer interested in arts and in all facets of quality
Involvement in professional activities

Swiss Association for Quality Software Engineering Group

European Organisation for Quality Software Group

World Congress for Software Quality

1995 San Francisco
2000 Yokohama
2005 München
2008 Bethesda
2011 Shanghai

American Society for Quality

Czech and Slovak Testing Board

International Requirements Engineering Board

International Certified Configuration Manager
In 2001 the Mária Valéria bridge between Štúrovo (Slovakia) and Esztergom (Hungary) was reopened. This bridge built in 1895 was in its history destroyed for a longer time than it was actually connecting the two towns.

The aim is to support artists, scientists and personalities from other professions who work on projects that emphasise uniting, connecting, and bridging.

The post of Bridge Guard requires a person in whose work boundaries of countries or eras are bridged, mental, social, religious or political boundaries are crossed, different scientific fields are connected, or various artistic media are utilised.
ICT Process Improvement and Assessment

- Process
- Assessment
- Process Metrics
- Improvement
- Process Monitoring
- PDCA revisited
To be able to improve ...

... you need to have something that is not good enough

if you want to improve a process ...

... you need to have one

that's easy
You can't have no process ...

... unless you don't work –

   but even if you don't work you're engaged in the idle process

what's not easy is to know your proces(ses)
Make your processes visible
Make your processes visible
Techniques for making processes visible

1. describe or engineer & describe the processes
2. monitor the processes continuously

Once you see it, you can improve it.
Process descriptions

great success!
- million companies have a described and ISO 9001 certified QMS
- you can see on intranet or paper how the processes of these companies suppose to work.
Process descriptions

great success!
- million companies have a described and ISO 9001 certified QMS
- you can see on intranet or paper how the processes of these companies suppose to work.

great!
- but …
  we still don't know how these companies actually do work!!
We need a tool for observation

check whether the work is done how the players agreed to and did describe it
→ internal audits

check whether the work is done how SEI (or another institution) thinks it should be done
→ assessment

audits and assessments deliver a snapshot
⇒ we know how the people worked on the day X
⇒ how do they work today?

is it so great?
Be aware of ...

... that there is an underlying assumption that the reference – a standard or a maturity model – defines the state of the art adequately

→ audit or assessment provides a useful picture
  ⇒ hints what to keep
  ⇒ hints where to change what

the development of such references and of the underlying models is a job with great responsibility

! my deep respect to all who do participate in this effort
Improvement?

not every change is for better, you better know

- what does work: it would be a sin to change it
- what does not work: it may be beneficial to change it

but

! don't use painkillers without taking into account the adverse reactions

in audits / assessments

→ nail down actual process strengths too
→ don't focus on deviations only
Another tool for observation – process metrics

predictor metrics

process characteristics

volume quality

process

input metrics

result metrics
Analogy with a car ride

process goal
- to arrive in Porto at 14:00

result metric
- arrival time in Porto, hotel Bessa

input metrics
- volume of petrol in the petrol tank [l]
- oil volume [l]
- cooling water volume [l]
- water for windscreen washer [l]
- break fluid volume [l]

predictor metrics
- current time [hh:mm]
- current speed [km/h]
- current distance to target Porto [km]
- current volume of petrol in the petrol tank [l]
Example software world (1)

process goal for development (many products)
- $\bar{\Omega}$ number of reported defects / kLOC and month $\leq 0.5$ first three months after deployment
- $\bar{\Omega}$ number of reported defects / requirement and month $\leq 0.01$ first six months after deployment

result metric (single product)
- kLOC
- number of reported defects / kLOC and month
- number of reported defects / requirement and month

input metric (single product)
- number of requirements

predictor metrics (single product)
- number of defects found in reviews / requirement
- number of defects found in tests / requirement
- number of defects found in tests / kLOC
Example software world (2)

process goal for project management (many projects)
- $\bar{\theta}$ project duration deviation: 80% within ± 15%
- $\bar{\theta}$ project cost deviation: 70% within ± 25%

result metric (single project)
- project duration [month]
- project costs [kCHF]

input metric (single project)
- number of requirements

predictor metrics (single project)
- number of accepted requirements
- number of accepted requirements / month
- number of requirements ready for test
- number of requirements tested / month
- number of requirements in development
- number of requirements developed / month
Make your processes measurable
Improvement!

with reliable evaluation of the situation at hand we need 'only' to implement improvements in the areas identified to be in need of

the only problem is the 'only'!
**What happens with the evaluation results?**

80% of the companies

- is happy that they know where they are (it does not matter whether they have any reason to be happy)
- don't change anything and look forward to the next evaluation
What happens with the evaluation results?

80% of the companies
- is happy that they know where they are (it does not matter whether they have any reason to be happy)
- don't change anything and look forward to the next evaluation

15% of the companies
- is unhappy what they know about where they are (it does not matter whether they have any reason to be unhappy)
- they identify three measures for conscious change of the way they work and implement these changes
What happens with the evaluation results?

80% of the companies
- is happy that they know where they are (it does not matter whether they have any reason to be happy)
- don't change anything and look forward to the next evaluation

15% of the companies
- is unhappy what they know about where they are (it does not matter whether they have any reason to be unhappy)
- they identify three measures for conscious change of the way they work and implement these changes

5% of the companies
- is easy-going, they know there is no reason to be happy or unhappy
- they go on with business as usual and change here and there to improve their work (results)
Improvement!

with reliable evaluation of the situation at hand we need 'only' to implement improvements in the areas identified to be in need of

the only problem is the 'only'!

organisations have a great inertia

- development organisations have the biggest (individual) inertia

Armour's observation on software process:
What all software developers really want is a rigorous, ironclad, concrete, hidebound, absolute, total, definitive, and complex set of process rules

Phillip G. Armour: The Laws of Software Process
Communications of the ACM, Vol. 44, Number. 1, 2001, pp. 15-17
Improvement!

with reliable evaluation of the situation at hand we need 'only' to implement improvements in the areas identified to be in need of

the only problem is the 'only'!

organisations have a great inertia

- development organisations have the biggest (individual) inertia

Armour's observation on software process:
What all software developers really want is a rigorous, ironclad, concrete, hidebound, absolute, total, definitive, and complex set of process rules they can break.

*Phillip G. Armour: The Laws of Software Process
Communications of the ACM, Vol. 44, Number. 1, 2001, pp. 15-17*
**Improvement!**

with reliable evaluation of the situation at hand we need 'only' to implement improvements in the areas identified to be in need of

the only problem is the 'only'!

organisations have a great inertia

- development organisations have the biggest (individual) inertia
  
  Armour's observation on software process:
  
  What all software developers really want is a rigorous, ironclad, concrete, hidebound, absolute, total, definitive, and complex set of process rules they can break.

  *Phillip G. Armour: The Laws of Software Process
  Communications of the ACM, Vol. 44, Number. 1, 2001, pp. 15-17*

- diagnosis is easy, therapy is harder to define and adamant to apply
Make improvements instantly …
Make improvements instantly ...

... but only one at a time
# From snapshot to 'continuous' overview (agile)

<table>
<thead>
<tr>
<th>To Do</th>
<th>In Work</th>
<th>Tests Passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story R01</td>
<td>Story R06</td>
<td>Story R02</td>
</tr>
<tr>
<td>Story R04</td>
<td>Story R11</td>
<td>Story R03</td>
</tr>
<tr>
<td>3x</td>
<td>2x</td>
<td>2x</td>
</tr>
<tr>
<td>Story R05</td>
<td>Story R13</td>
<td>Story R08</td>
</tr>
<tr>
<td>1x</td>
<td>5x</td>
<td>3x</td>
</tr>
<tr>
<td>Story R07</td>
<td>Story R10</td>
<td>Story R12</td>
</tr>
<tr>
<td>3x</td>
<td>1x</td>
<td>4x</td>
</tr>
<tr>
<td>Story R09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To Do</td>
<td>In Work</td>
<td>Tests Passed</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>--------------</td>
</tr>
<tr>
<td>Story R01 3x</td>
<td>Story R06 2x</td>
<td>Story R02 5x</td>
</tr>
<tr>
<td>Story R04 2x</td>
<td>Story R11 3x</td>
<td>Story R03 2x</td>
</tr>
<tr>
<td>Story R05 1x</td>
<td>Story R13 5x</td>
<td>Story R08 3x</td>
</tr>
<tr>
<td>Story R07 3x</td>
<td></td>
<td>Story R10 1x</td>
</tr>
<tr>
<td>Story R09 4x</td>
<td></td>
<td>Story R12 4x</td>
</tr>
</tbody>
</table>

- **To Do**: 5 stories, 13 story points
- **In Work**: 3 stories, 10 story points
- **Tests Passed**: 5 stories, 15 story points
## From overall view to workflow (lean)

<table>
<thead>
<tr>
<th>backlog</th>
<th>design</th>
<th>ready</th>
<th>code</th>
<th>resolved</th>
<th>test</th>
<th>done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
</tr>
<tr>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
</tr>
<tr>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
</tr>
<tr>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
</tr>
<tr>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
</tr>
<tr>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
</tr>
<tr>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
<td>Story</td>
</tr>
</tbody>
</table>
# Product structure and the workflow, in figures

<table>
<thead>
<tr>
<th>System</th>
<th>backlog</th>
<th>design</th>
<th>ready</th>
<th>code</th>
<th>resolved</th>
<th>test</th>
<th>done</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuSy A</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SuSy B</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>SuSy C</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
# Process signals

<table>
<thead>
<tr>
<th>stage</th>
<th>req</th>
<th>specified</th>
<th>design</th>
<th>ready</th>
<th>code</th>
<th>resolved</th>
<th>test</th>
<th>done</th>
</tr>
</thead>
<tbody>
<tr>
<td>healthy</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>design?</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>design??</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>design!</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>paralysis</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

↓ time
Make process execution visible, continuously
Make process improvement to rule, not to exception
PDCA – surprising search results
PDCA – many shapes and letters
PDCA – many shapes and colours
PDCA – many shapes .. directions
PDCA – the cycle is not always round

Plan
- Sony Group Environmental Vision
- Green Management 2010
- Formulation of Sony Group environmental rules and annual business plans

Do
- Formulation of each business unit’s annual business plan
- Implementation of environmental management based on the annual business plan

Check
- Audit and review of annual business plan and Green Management 2010, as well as performance assessments and decisions regarding awards

Act
- Review by top management
PDCA – variations

- **Act**
  - Take appropriate action
  - Check the effects of implementation

- **Plan**
  - Determine goals and targets
  - Determine methods of reaching goals
  - Engage in education and training
  - Implement work

- **Check**
  - Standardise

- **Do**
  - Continuous improvement

Flowchart:
- Reflect → Plan → Observe → Do
- Act → Plan → Check → Do
PDCA – extension
PDCA career
4D + D cycle

Do your work

Define processes

Derive measures

Determine state

Deploy change

Do your work

IIP-378
4D + D cycle

- Do your work
- Determine state
4D + D cycle

Do your work

Deploy change

Determine state

Define processes

Derive measures
Out of the Crisis

W. Edwards Deming

My best greetings to Mr. K. Mikawa

W. Edwards Deming
27 June 1959
A kind of conclusion ...

Process improvement is the use of opportunities you have to discover robust processes are through to change under continuous observation there is a greater chance for timely improvement agile can be made even lean but never mean