



Software Quality Consulting

Putting Software Quality into Effect

Dr. Markus Pizka

Agenda

1 About us

2 Previous research on software quality

3 Practical reality

4 How to deal with it

5 Experiences

About us

- **15 Years research in Systems Architecture and Software Engineering**
- **information technology and strategy GmbH**
 - since 2003
 - 30% - 40% annual growth
- **Locations**
 - Munich, Germany
 - Tallinn, Estonia
- **Expertise**
 - Automotive, energy, financial, pharmaceutical, IT
 - Research & development, production, controlling, sales
- **Services**
 - Solution Engineering and Reengineering
 - Software Governance Consulting



Customers and Projects

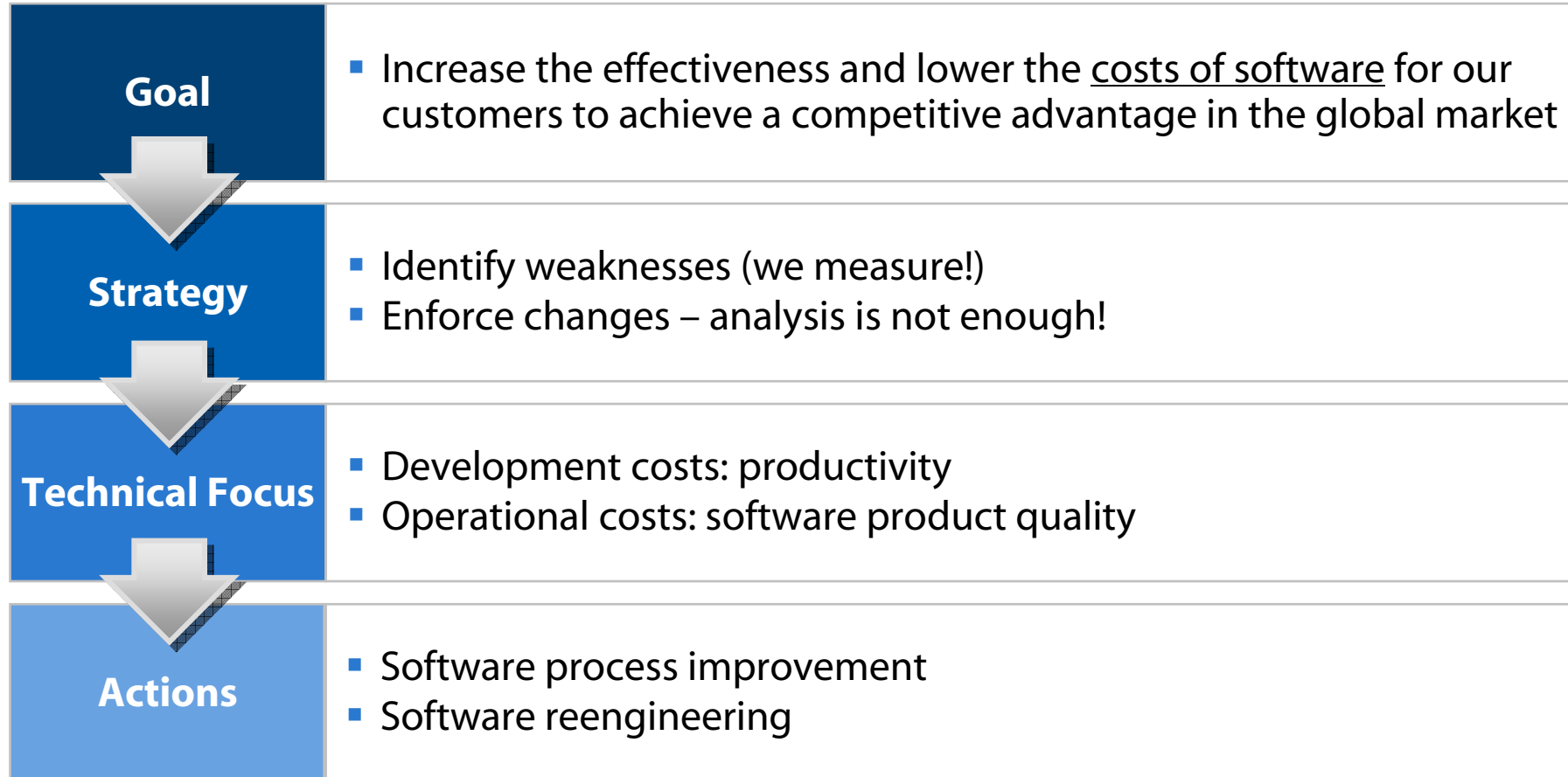


Sample Projects

- **Governance**
 - Lean Management Principles
 - Software Health Check (Assessment of Applications)
 - Software Tuning
- **Engineering**
 - Limit and risk portfolio management
 - World-wide online service (car)
 - Stock market pricing (energy)
 - Product information management

Software Governance

What we do ...



History

Software Quality Research

- **2001:** **Technische Universität München – Systems Engineering**
- **2002:** **Project VSEK – www.software-kompetenz.de**
- **2003:** **Center of Competence „Software Maintenance“**
- **2004:** **Productivity and Quality in Long-Lived Software-Systems**
 - With Siemens Business Services
 - Design of a new software quality model
 - Initial tool support
- **2004:** **First Lecture incl. Labs on „Software Maintenance“**
- **Today:** **Center of Competence „Software Quality“**
 - 5+ permanent researchers
 - numerous projects with several industrial partners
 - 20+ research papers on quality models and assessment

Results

Measuring complexity is useless

- **Halstead Volume**

- (different) operators and operands

- **McCabe's Cyclomatic Complexity**

- Control flow
- Counterexamples easy to construct

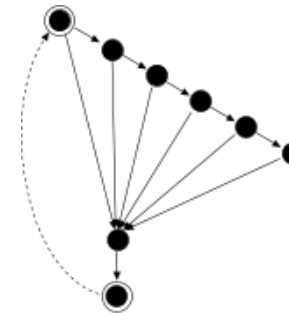
- **SEI Maintainability Index =**

$$71 - 5.2 * \ln(\text{avgHV}) - 0.23 * \text{avgCC}(g') - 16.2 * \ln(\text{avgLOC}) + 50 * \sin(\text{sqrt}(2.4 * \text{perCM}))$$

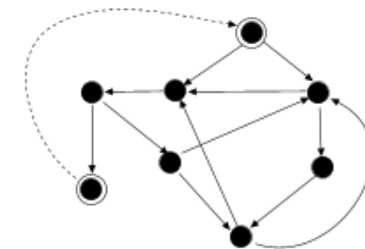
HV: Halstead Volume perCM: % comments

- **Given a system of "complexity" C, what is the effort of**

- changing an error message
- changing the domain of dates to [0;9999]



$$C(G)=13-8+2=7$$



$$C(G)=13-8+2=7$$

so what?

Results

Maintainability is a myth

- **„Maintainability“ is a misleading term**

- verb “to maintain” transformed into attribute “maintainable” of system
- e.g. “readability” – ability to read depends mostly on reader!

- **C.S. Hartzmann 93. „Maintenance Productivity“**

- | | |
|------------------------------|-----------------------|
| - T1 – Problem understanding | T5 – Specification |
| - T2 – Administrative time | T6 – Coding |
| - T3 – Tool setup | T7 – Unit test |
| - T4 – Analysis | T8 – Integration test |
| | T9 – Review |
- ~ 50% comprehension

- **Cross cutting aspect: organization, tools, software, ...**

- Psychology, skills, team, communication, processes
- Documentation, rules, modeling, languages, generation, tools, ...

Results

Quality criteria must be both justified and testable

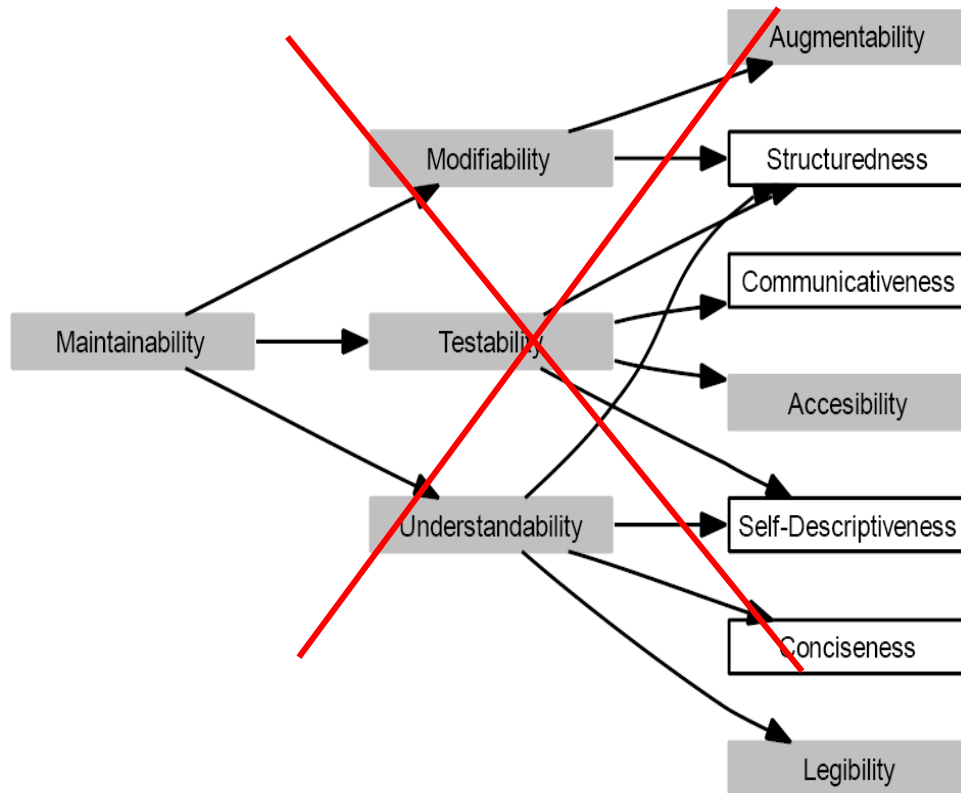
- **OO metrics: CC, NOC, CBO, LCOM, ...**
 - can be measured but neither sufficient nor necessary
- **“average effort for the removal of bugs ...” (variance?)**
- **„... minor changes ... minor costs ...“**
 - tautology, counterexamples: Y2k, 32→64 Bit
- **Quality guidelines**
 - „Simplicity of interface and modules“
 - „Readable code“
- **ISO 9126 – internal metrics**
 - (1) Analyse: „can ... identify specific operations“ ~ size of activity log
 - (2) Change: „...easily change the software“ ~ inline-comments per change

not testable
⇒ no effect

not justified
⇒ ignored

Results

Tree-like quality models are troublesome

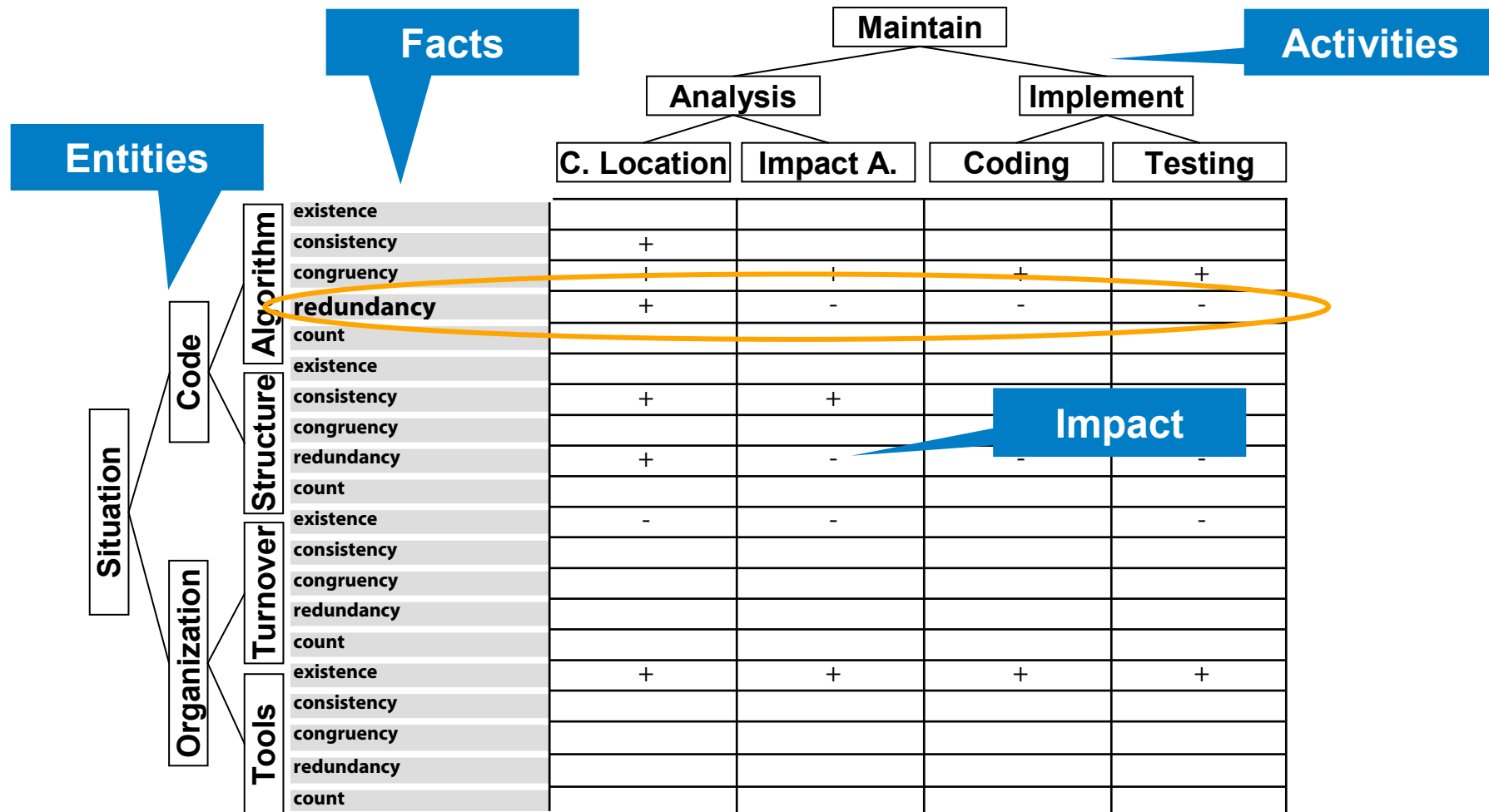


Improvement Needed

1. Separate goals from influences - Activities and Situation
2. Do not limit criteria selection to empiricism and "easy to measure"
3. Do not expect simple answers (few criteria)

A 2-Dimensional Quality Model

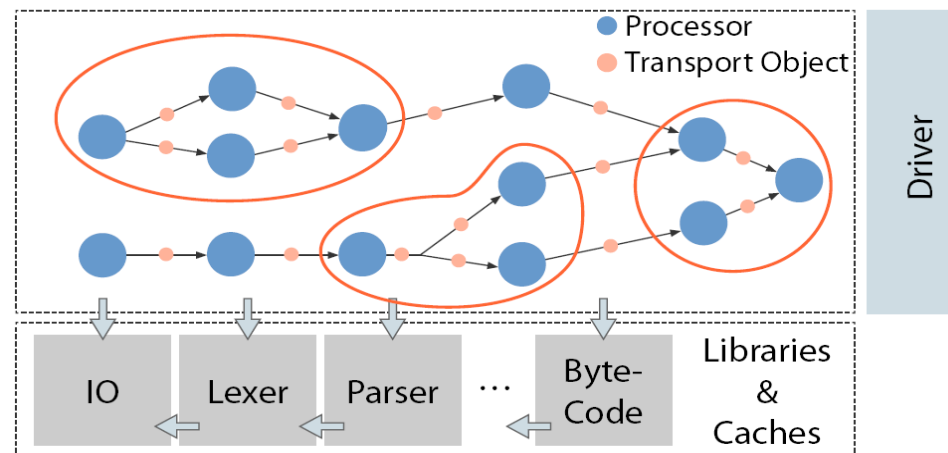
Activities versus Facts



Tool-Supported Assessment

Semi-automated is the key

- **Many important facts are semantic aspects** \Rightarrow **review by expert (e.g. naming)**
 - Tools can speed-up the „manual“ assessment process (e.g. sorted-list of all occurring names)
- **Some facts can be measured automatically**
- **Tool ConQAT (<http://www.conqat.org>)**
 - Highly configurable
 - Fast - no generic fact database but specialized light-weight processors (e.g. Clone-Identification)



Summary of Research Findings



Reasonable understanding of what software quality is



Strategy for quality assessments



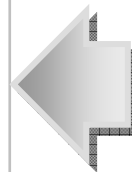
Initial tools

Practical Reality

Reality is driven by ...

Money

- The universal unit of measurement
- Everything costs
- The costs are always & quickly visible
 - Invoices
 - "Wallet"
- The value (potential income) is always hard to see
- Money is limited



Personal Benefits

aka „politics“

- Safety
- Carreer
- Effort
- Money

„Person“

- Single individual
- Group

Truths From the Far (Practical) Side

Quality defects are not hard to discover

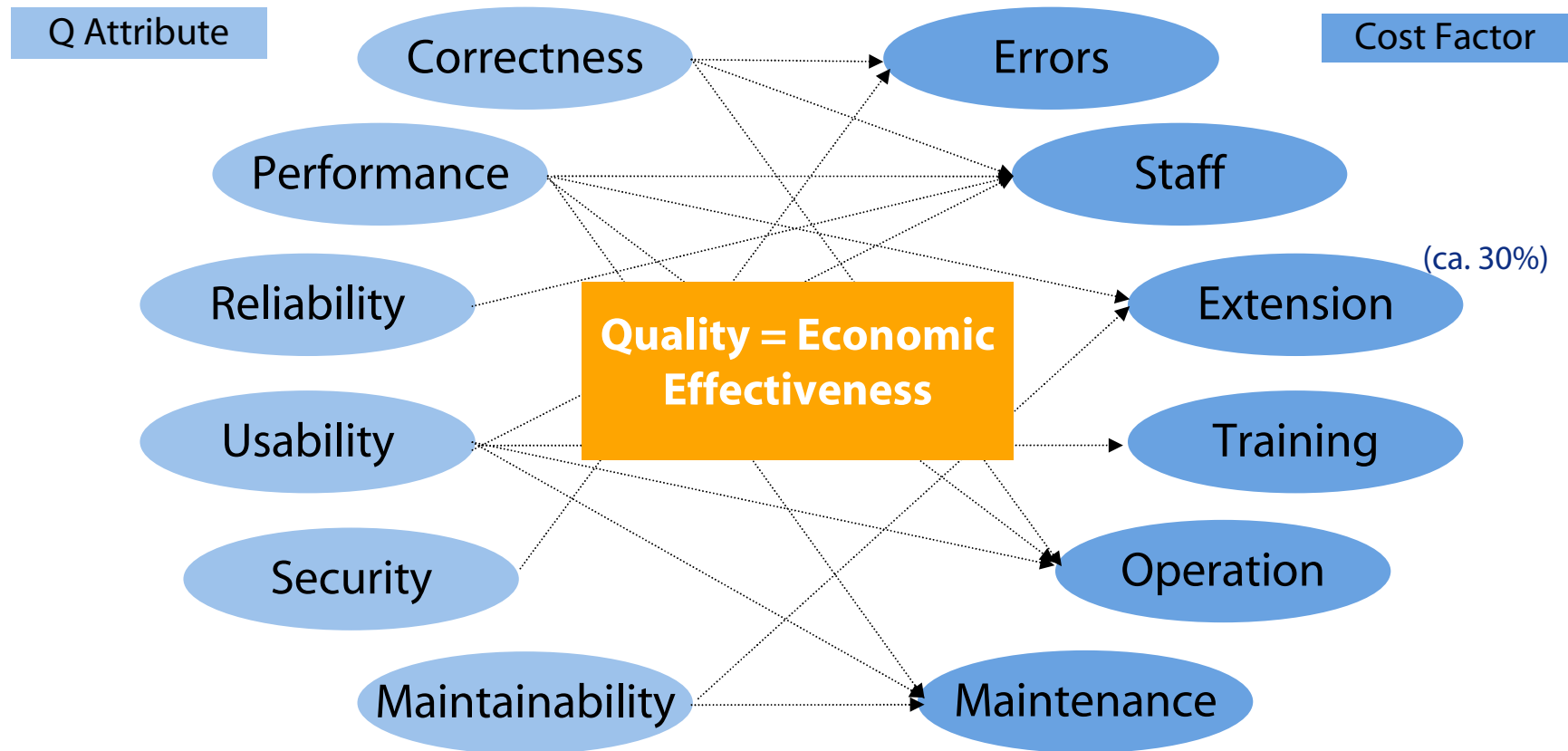
- **Excessive development and operational costs for trivial systems**
 - e.g. 0.5 Mio EUR for a simple technical interface
- **Excessive variance in productivity (up to 60 times !)**
- **60% of all software costs are operational costs (w/o failures)**
- **Unacceptable technical shortcomings**
 - up to 90% redundancy (cloning), 50% unused code (but Y2K migrated)
 - single procedures with >5.000 LOC and lots of GOTOs
 - bubblesorts, ...
- **No accepted or mandatory quality standards**
 - ISO9126, CMMi and others generally unknown or at least misunderstood
 - no quality assurance besides functional testing (and maybe security analysis)
- **Quality is regarded a cost factor; i.e. the „magic triangle“ – quality costs**
- **Internal costs are regarded differently than external costs**

How to deal with it

- **Communicate, educate**
- **Provide plausible estimates for the economic benefit**
 - Answer the question „How much is quality worth?“
- **Demonstrate credibility**
 - e.g. customer only pays in case of proven success
- **Build trust - Go step-by-step from small to large (aka „iterative“)**
- **Perform systematic change management**
- **Create win-win or even win-win-win situations**
 - embrace opponents?
- **Document – publish or perish!**

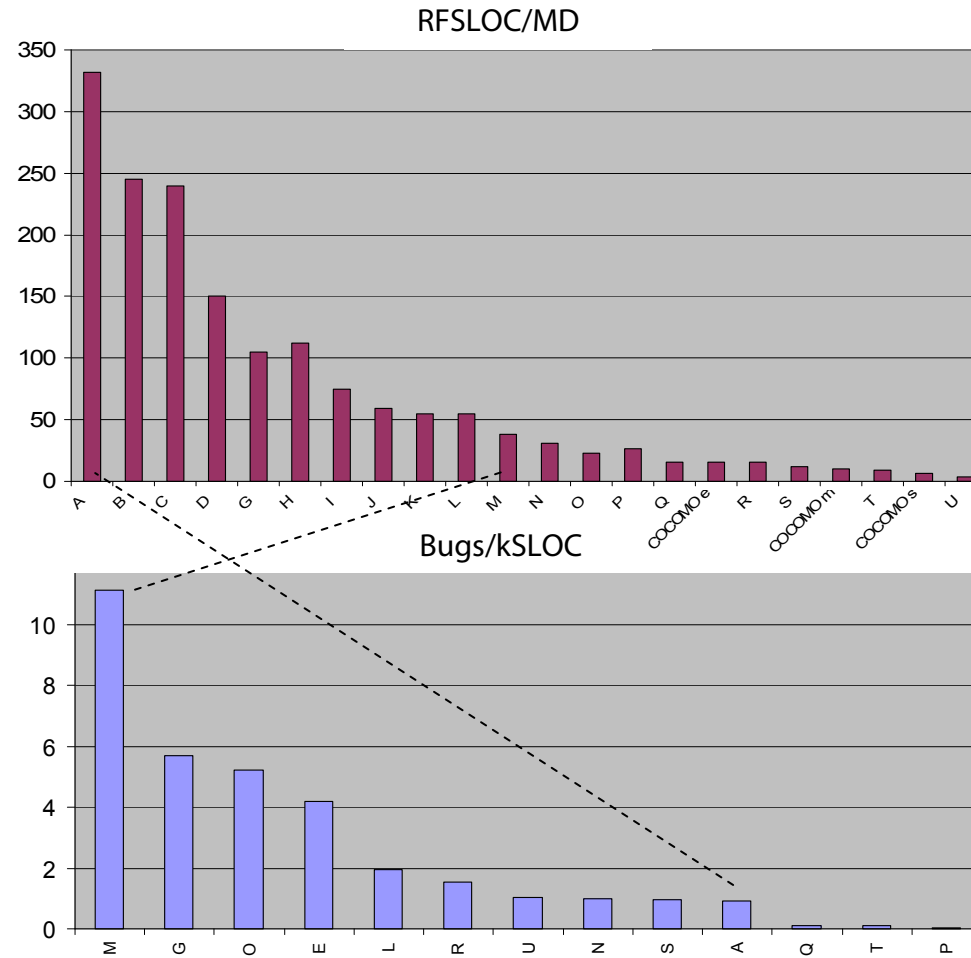
Educate

What's quality?

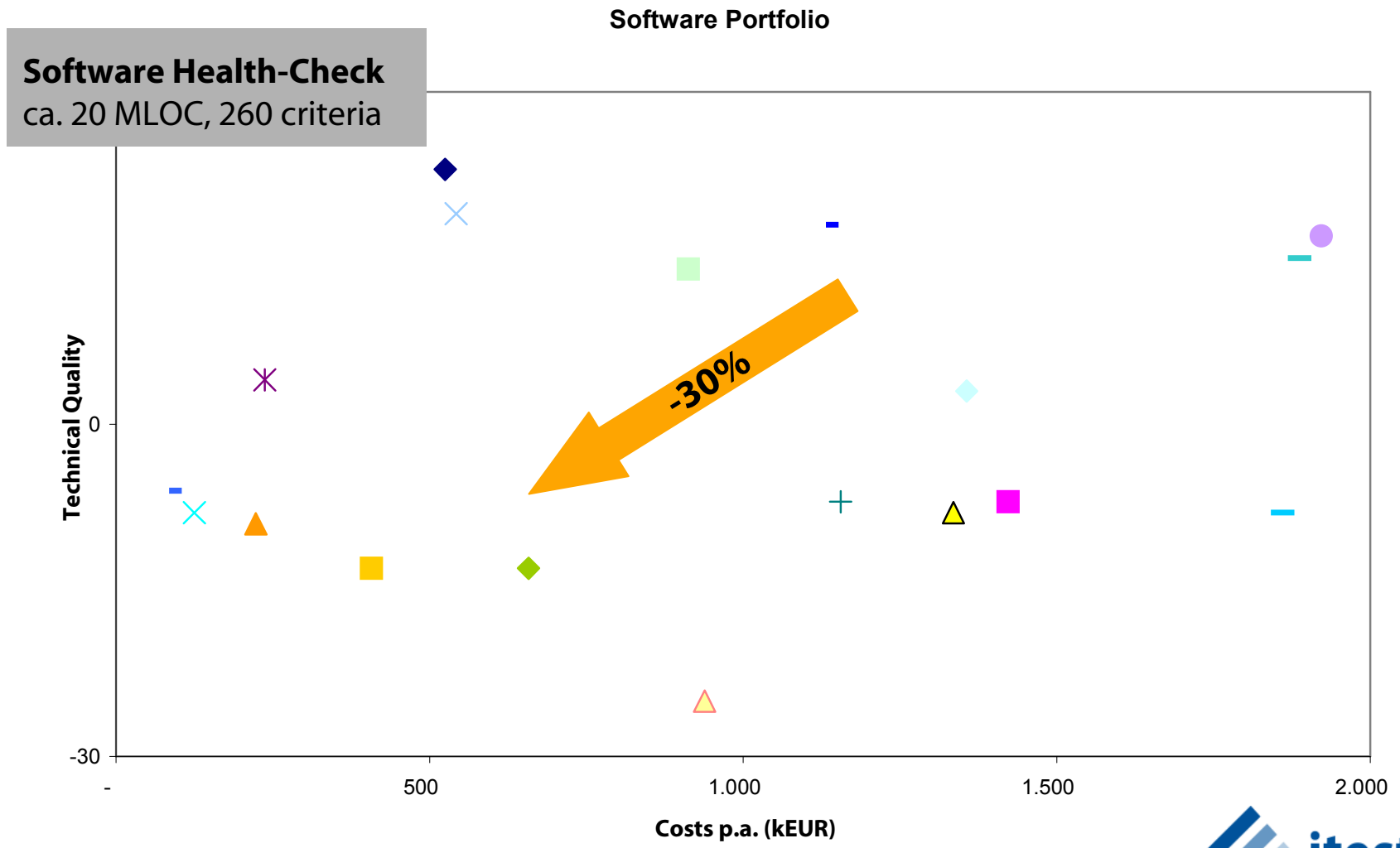


The Magic Triangle is Untrue

Quality doesn't cost, it saves – early!



Economic Perspective



Plausible Estimates

Cost factor	Amount p.y.	Chance	Explanation	Amount p.a.
Development	250.000 EUR	- 30%	50% cloning, structure	75.000 EUR
Maintenance	50.000 EUR	- 30%	see above	15.000 EUR
Operations (Staff)	150.000 EUR	- 50%	80% management of user accounts	75.000 EUR
Operations (Ressources)	490.000 EUR	- 40%	loop-depth 8, bubblesort	196.000 EUR
Use (Staff)	? EUR	- 25%	50 open requirements	? EUR
Failures	> 1.000.000 EUR	- 20%	known risks	200.000 EUR
	1.940.000 EUR			561.000 EUR

Build Trust

Savings guaranteed by contract

Reduction MipsH		Savings EUR (at 1,46€/MipsH)		discount	discount	total
from	to	from	to			
1	30.000	1,46 €	43.800,00 €	90%	130.239,90 €	14.471,10 €
30.001	60.000	43.801,46 €	87.600,00 €	80%	115.768,80 €	28.942,20 €
60.001	90.000	87.601,46 €	131.400,00 €	70%	101.297,70 €	43.413,30 €
90.001	120.000	131.401,46 €	175.200,00 €	60%	86.826,60 €	57.884,40 €
120.001	150.000	175.201,46 €	219.000,00 €	50%	72.355,50 €	72.355,50 €
150.001	180.000	219.001,46 €	262.800,00 €	40%	57.884,40 €	86.826,60 €
180.001	210.000	262.801,46 €	306.600,00 €	30%	43.413,30 €	101.297,70 €
210.001	240.000	306.601,46 €	350.400,00 €	20%	28.942,20 €	115.768,80 €
240.001	270.000	350.401,46 €	394.200,00 €	10%	14.471,10 €	130.239,90 €
270.001	360.000	394.201,46 €	525.600,00 €	0%	0,00 €	144.711,00 €

Experiences (1)

The Responsibility Deadlock

■ **Situation**

- Savings of 1 Mio EUR guaranteed
- Cost to implement: 250.000 EUR
- Break-Even: 6 months
- Nobody signs the contract and start the project

■ **Explanation**

- IT organizations of many large enterprises are structured into Plan, Build, and Run
- Build would have to pay
- Run would have the benefit
- Departments are controlled as individual cost centers
- Top-level management has no budgets besides what's distributed to P, B & R

Experiences (2)

Successful optimizations

Example 1

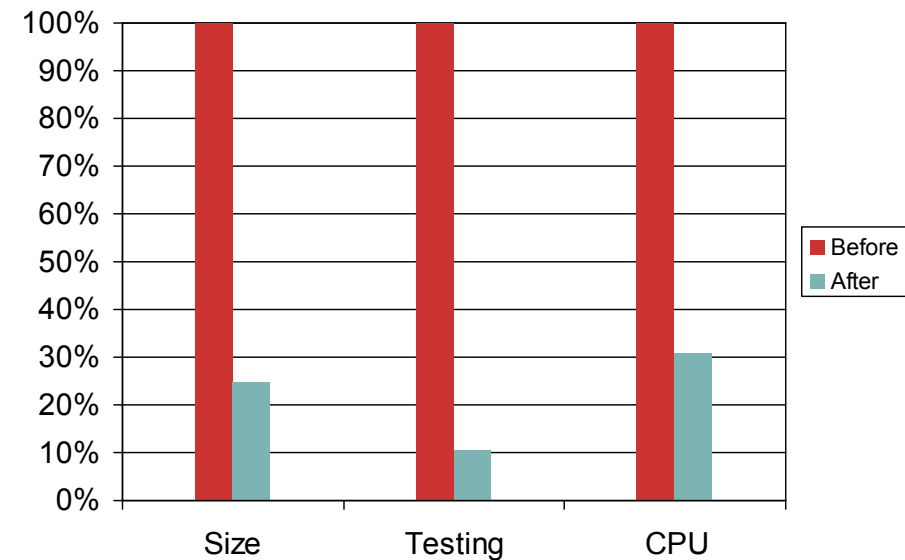
Goal: -30% Operations (Ressources)

Order	Measure	Result
1	MR Fetch	-55%
2	MR Fetch	-32%
3&5	MR Fetch, Bin Search, Quicksort	- 85%
4	MR Fetch	-19%
6	Bin. Search	- 57%
7	Cursor Insert	- 20%
8	DB Index	- 40%
9	Caching	- 30%

- 120 kEUR p.y. i.e. - 42%

Example 1

Goal: -50% Maintenance Costs



Experiences (3)

```
CUR = FIRST;
DO Z = 1 TO ANZ
WHILE (CUR ^= NULL)
UNTIL (LAST);
    IF CUR->SP.NEXT ^= NULL
        THEN CUR = CUR->SP.NEXT;
    ELSE
        LAST = '1';
    END;
```

$O(n^2)$ instead $O(n)$ → 500.000 statt 1.000 steps

QuaMoCo

Software-Quality: Flexible modeling and integrated controlling

- **Research project**

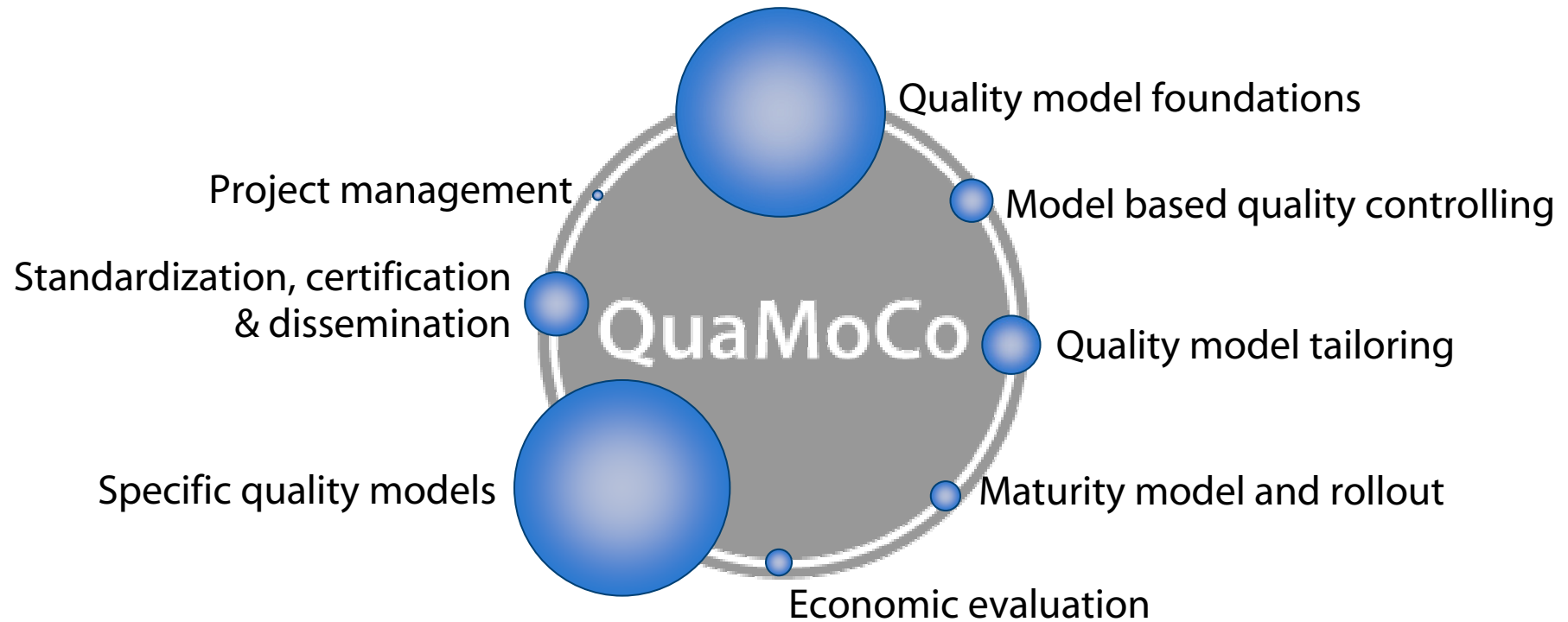
- Three years (2009 – 2011), 5.9 Mio EUR
- Three iterations
- 8 work packages
- co-funded by the German Federal Ministry of Education and Research

- **Goal: A Software Quality standard**

- domain independent,
- customizable,
- justified,
- economically evaluated.



QuaMoCo Work Structure





**Do You Have
Questions?**