

Manfred Broy, Florian Deißeböck, Markus Pizka

Demystifying Maintainability

May 21st 2006
WoSQ 06 Shanghai



Outline

Relevance

Maintainability

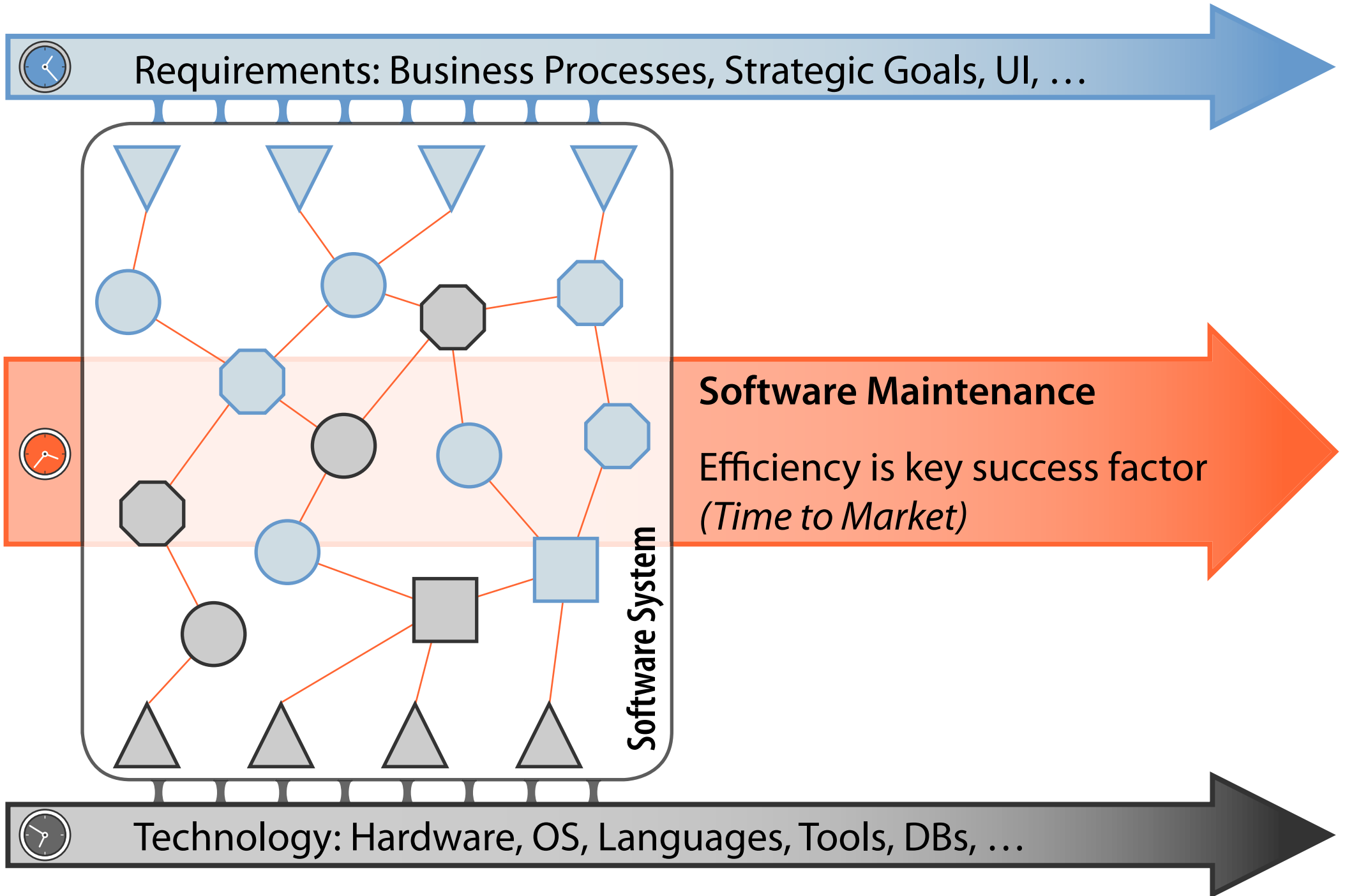
Metrics, Conventions, Models

Quality Matrix

Incremental development, principles, tools

Conclusion & Future Work

Relevance



What is »Maintainability«?


- *»The effort needed to make specified modifications to a component implementation.«* (SEI)
- *»The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment.«* (IEEE)
- *»correction of minor bugs requires only minor efforts«*
- Software-Metrics: McCabe, Halstead, CBO, LCOM, ...
- SEI Maintainability Index:


$$171 - 5.2 \cdot \ln(\text{avgHV}) - 0.23 \cdot \text{avgCC}(g) - 16.2 \cdot \ln(\text{avgLOC}) + 50 \cdot \sin(\sqrt{2.4 \cdot \text{perCM}})$$

HV:	Halstead Volume	CC:	Cyclomatic Complexity
LOC:	lines of code	perCM:	% Comment Lines



Conventions

- Companies:
 - Simplicity of interfaces and modules
 - Low structural complexity
 - Readable code
 - Changeable and complete documentation

Details?
Assessment?
- ISO 9126-3:
 - Analyzability: »effort to diagnose«
Size of Activity Log
 - Changeability: »effort to implement a modification«
Inline Comments per Change

Rationale?
Structure?
- not (justified \wedge assessable) → ignored / without effect

Real-World Problems

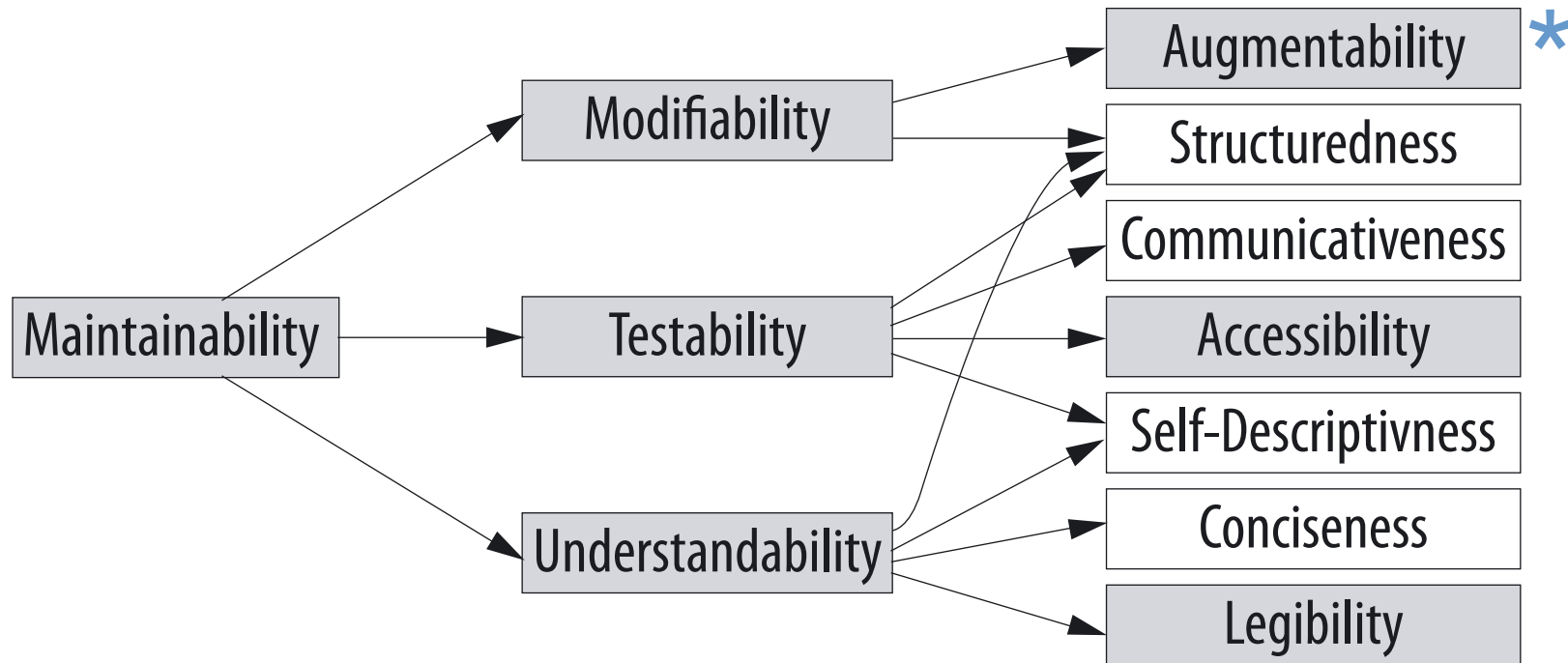
- Program comprehension ($\approx 50\%$ of maintenance activities):
 - Identifiers
 - Unnecessary code
 - Hidden dependencies
 - »Programming tricks«
 - Source code format
- Change:
 - Dependencies
 - Redundant code
- Lack of tool support
- Outdated documentation
- Compile times
- Turnover

```
function mr_mr_1(mr, mr_1)
  if Null(mr) or Null(mr_1) then
    exit function
  end if
  mr_mr_1 = (mr - mr_1)
end function
```

```
#if 0
Message-Id: <8705050653.AAarpa>
Subject: gnu emacs 18.41 on iris ...
Date: 04 May 87 23:53:11 PDT (Mon)
From: raible@orville.arpa

... if (utime (name, &utb) < 0)
    return; ...
#endif
```

Quality Models



Approach

- Break-down of quality criteria
- Tree-like structure

Problem

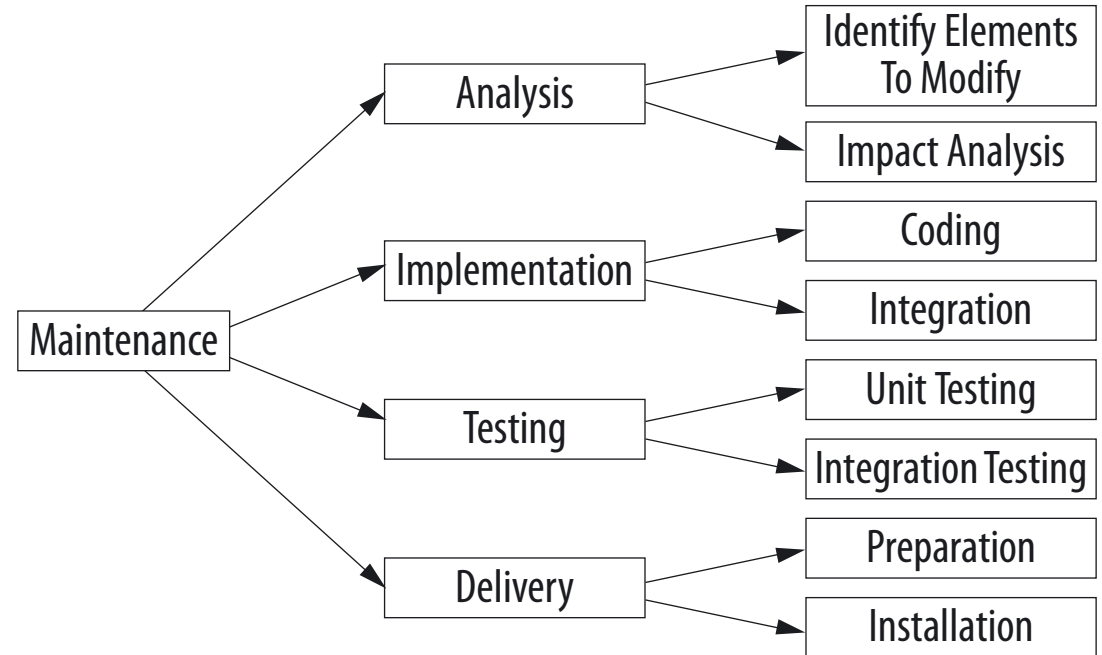
- Characteristics vs activities
- Ambiguous edge semantic

* B. W. Boehm et al., *Characteristics of Software Quality*, 1978

Separation of Activities & Facts

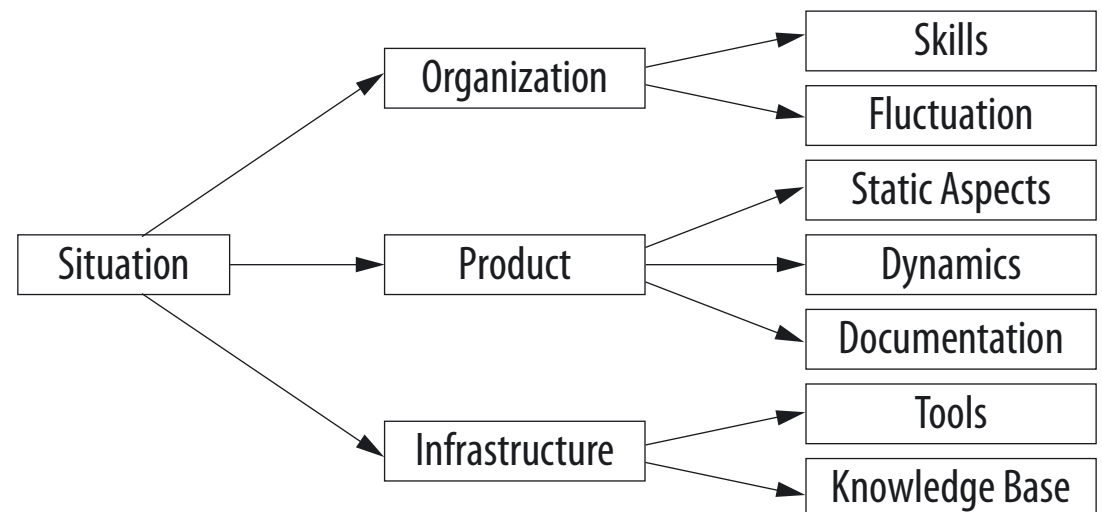
Activities

- Define costs
- Depend on maintenance task

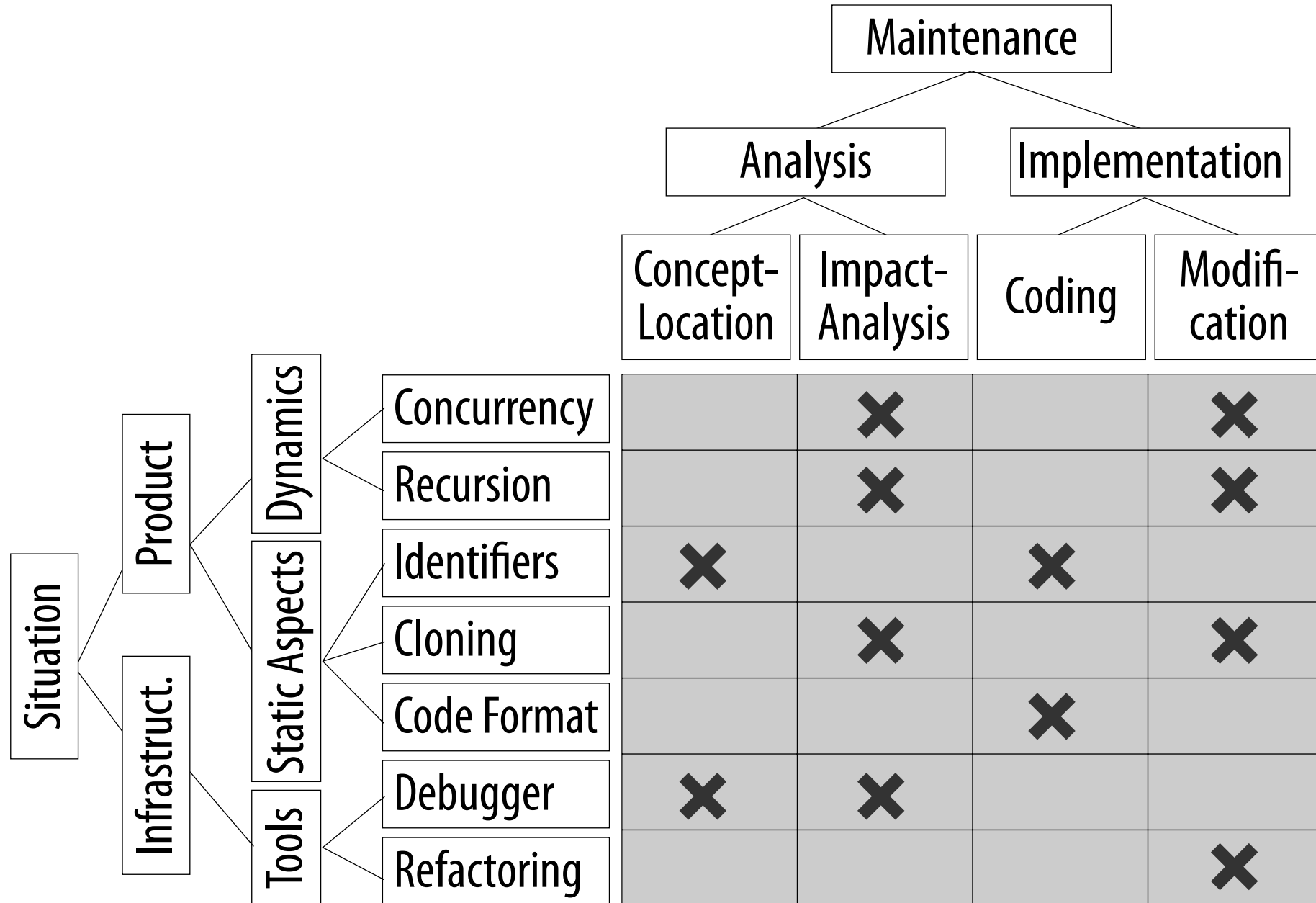


Facts

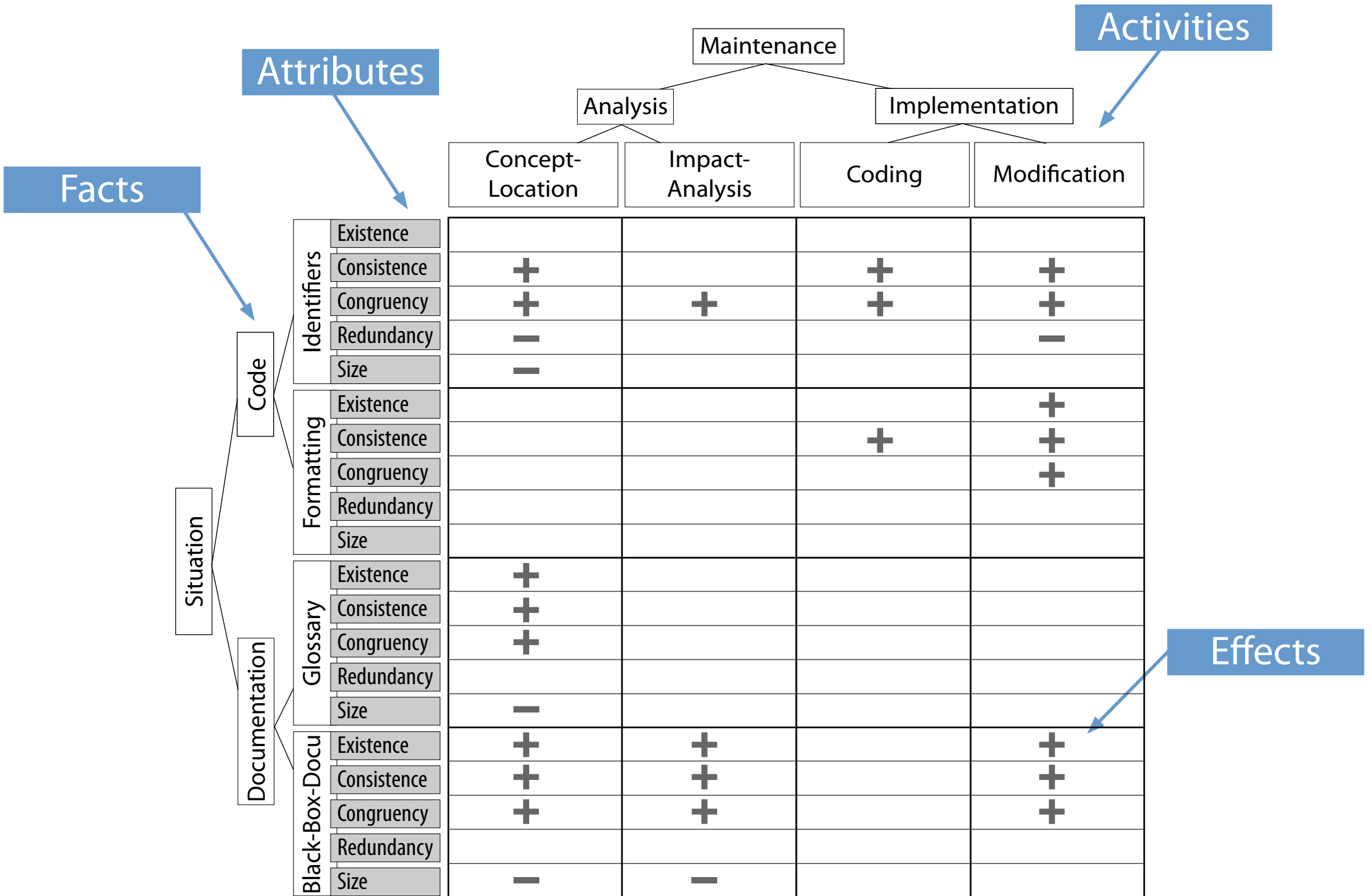
- FCM-like but without activities
- Includes organizational aspects



Quality Matrix



Extended Quality Matrix



Extended Quality Matrix

Assessment

- Subject of assessment: *attributed Facts*
- Types:
 - automatic: »Switch must have Default«
 - semi-automatic: Redundancy, Dead Code
 - manual: *wrong* Algorithm

Current Version

# Facts	145
# Activities	28
# Attributes	16
# Attributed Facts	178
# Effects	197
Σ Model Elements	564

Sources

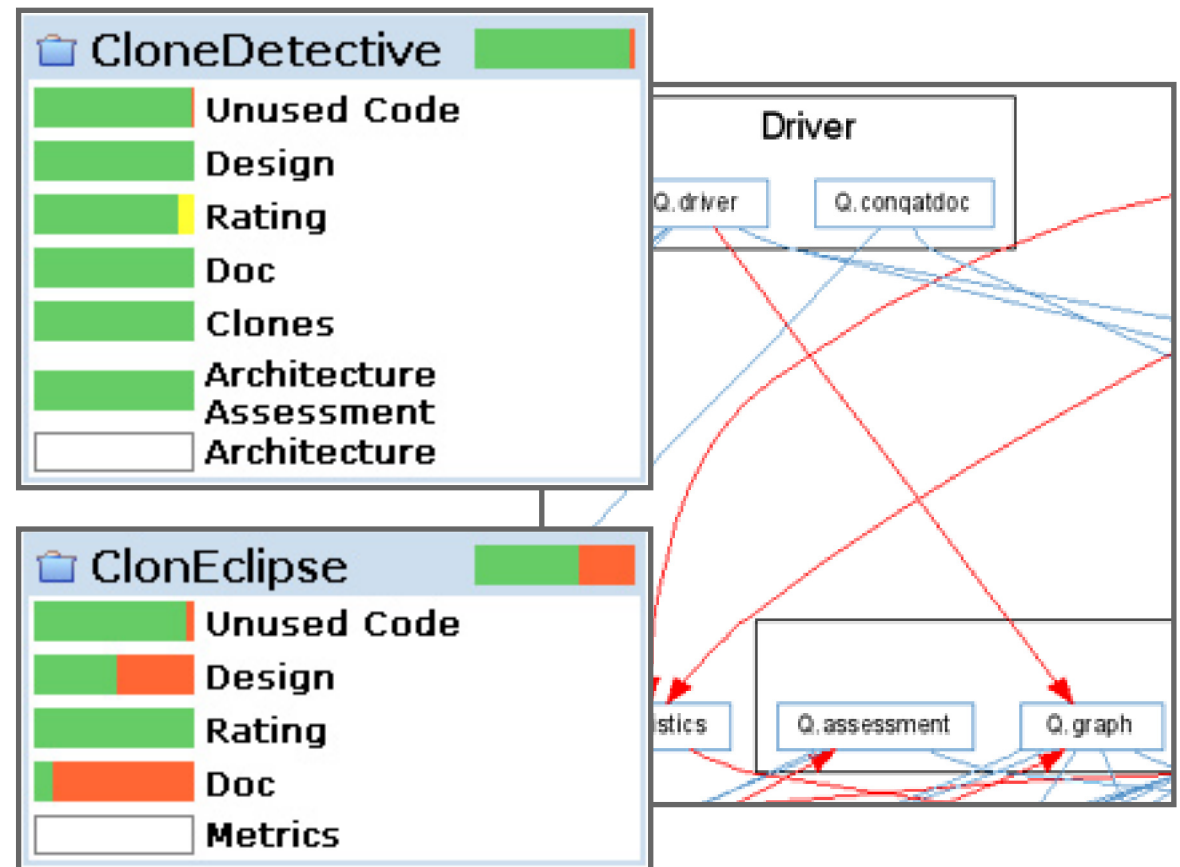
- Experiences of industrial partners
- Literature
- Standards, Conventions
- Plus a lot of work

Quality Controlling

- »By the time you figure out you have a quality problem, it is probably too late to fix it.« (Reel 92)
- Continuous, real-time quality controlling
- Improves product quality *and* developer education

Tool Support: ConQAT

- Integrated display of diverse quality characteristics
- Information aggregation
- Flexible, extensible architecture
- Batch mode
- High performance



Experiences

System

- Telecommunication domain
- 15 years, 150 change requests/year
- 3.5 MLOC C++, COBOL, Java

Lessons learned

- Way of presentation is crucial
- Well-foundedness helped to overcome scepticism
- Tool-supported quality controlling was perceived as great benefit
- Fostered lasting quality discussion
- Identification of previously unknown quality factors

Conclusion & Future Work

Conclusion

- Maintainability, a key success factor, lacks an accepted definition.
- There is no such thing as »the maintainability of a system«.
- Maintainability depends on the maintenance activities carried out.
- Maintainability does not depend on the product only.

Future Work

- Define model for maintainability of embedded systems (with MAN).
- In-depth analysis of a very small set of facts regarding the development infrastructure for mainframes (with BMW).
- Apply model to other quality attributes like usability.