CHAIR IV
Software & Systems Engineering

TECHNISCHE UNIVERSITÄT MÜNCHEN

Software Quality Management

Manfred Broy
joint work with Florian Deißenböck, Stefan Wagner
Software & Systems Engineering
Institut für Informatik
Technische Universität München

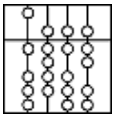
CHAIR IV
Software & Systems Engineering

TECHNISCHE UNIVERSITÄT MÜNCHEN

Outline

- Impact of software quality
- Misconceptions … what is software quality?
- Achievements
- Criteria for sound quality criteria
- The role of quality models
- Real-time quality controlling processes
- Tool support

Impact of Software and its Quality



TECHNISCHE UNIVERSITÄT MÜNCHEN

The role of software today:

Software landscapes

- 3000 applications and more in one company

Embedded systems

- more than 2500 software based functions in a car

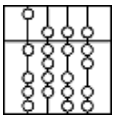
Software systems:

- Commodity
- Legacy
- Innovation driver

How good are companies in managing their software portfolios

- Management by metric values and numbers
- Management by expertise

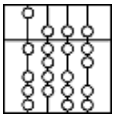
Economic evaluation



TECHNISCHE UNIVERSITÄT MÜNCHEN

- Quality defects increase the TCO, e.g.
  - Maintenance costs
  - Incidents costs
  - Cost of operation (e.g. CPU second)
  - Economical opportunity loss (Opportunity costs)
- Cost evaluation of quality defects
  - (Price per CPU second) * (wasted CPU second by quality defect) = cost of quality defect
  - E.g. replacing primitive SQL data access methods by advanced saves up to 55% CPU seconds worth tens of thousand € per year

What is Quality?

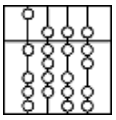


TECHNISCHE UNIVERSITÄT MÜNCHEN

- „Like beauty, everyone may have his or her idea of what quality is“ (ISO 9000:2000)
- Wordnet
  - a degree or grade of excellence or worth; "the quality of students has risen"; "an executive of low caliber"
  - choice: of superior grade; "choice wines"; "prime beef"; "prize carnations"; "quality paper"; "select peaches"
  - …
- „Quality is a complex and multifaceted concept“ (Garvin, 1984)

[Plato. Symposion. ca. 380 v. Chr.]
[R. M. Pirsig. Zen and the Art of Motorcycle Maintenance. Bantam, 1999.]

What Do We Mean by Quality?



TECHNISCHE UNIVERSITÄT MÜNCHEN

- *Totality of characteristics of an entity that bears on its ability to satisfy stated and implied needs.*

  ISO 8402 (withdrawn)

- *The degree to which a system, component, or process meets specified requirements.*

  IEEE SE Glossary

- *The discipline of software quality is a planned and systematic set of activities to ensure quality is built into the software.*

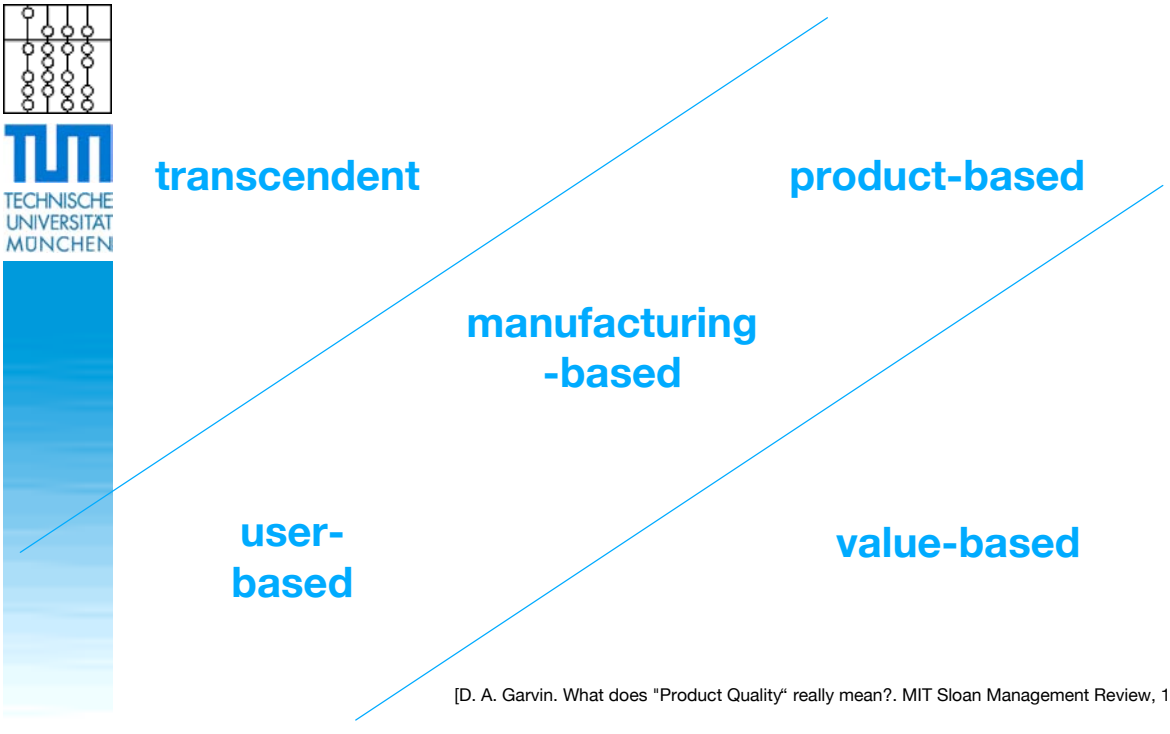
  NASA

?

Existing quality standards

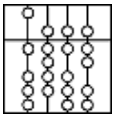
General standards	ISO/IEC 9126: SW engineering – SW Product quality ISO/IEC 15504: SPICE, *SW Process* ISO/IEC 25000: SW Product Quality requirements and evaluation ISO/IEC 12119: Software packages -- Quality requirements and testing ISO 55350: Concepts of quality management and statistics
Domain specific standards	ISO 15005 (Road vehicles - Ergonomic aspects of transport information and control systems - Dialogue management principles and compliance procedures) Automotive SPICE: Domain specific ISO/IEC 15504
Business specific quality models	Q-Index: SAP SW Cockpit: Capgemini sd&m

Quality Views



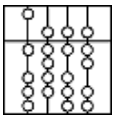
[D. A. Garvin. What does "Product Quality" really mean?. MIT Sloan Management Review, 1984]

Definitions of Software Quality



- Crosby (1979): Quality means conformance to requirements.
- Juran (1988):
  - Quality consists of those product features which meet the needs of customers and thereby provide product satisfaction.
  - Quality consists of freedom from deficiencies.
- Pressman (2000): Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.
- IEEE (1991): Software Quality is:
  - the degree to which a system, component, or process meets specified requirements.
  - the degree to which a system, component, or process meets customer or user needs or expectations.

Real Life Software Quality



Software quality has many different aspects:

- Safety
  - *An airbag killed a baby - although it was deactivated.*
- Functionality - meeting users' expectation
  - *In automotive software systems more than 50 % of reported bugs are formally not bugs but misconceptions between the users' expectations and the specified (and implemented) behavior.*

"it is not a bug, it is a feature ..."

This shows the significance of requirements engineering and verification!

Software Quality is Holistic

Misconceptions

- Quality can be measured automatically ?

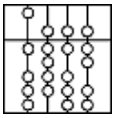
SEI-Maintainability-Index:

MI = 171 - 5.2 * ln(avgHV) - 0.23 * avgCC(g') –
6.2 * ln (avgLOC) + 50 * sin (sqrt(2.4 * perCM))

HV: Halstead Volume
LOC: lines of code
CC: extended Cyclomatic Complexity
perCM: % of comment lines

Can it?

Misconceptions II

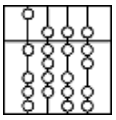


Good process ⇨ Good product ???

CMM Level	Min	Avg	Max
1	0,150	0,750	4,500
2	0,120	0,624	3,600
3	0,075	0,473	2,250
4	0,023	0,228	1,200
5	0,002	0,105	0,500

Defects per Function Point (C. Jones ´03)

The many facets of software quality
A real life example: embedded systems in cars



Consider following aspects of software quality

- Code size - memory demand
- Required execution time - processor exploitation

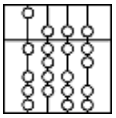
++++++++++++++++++++++++++++++++++

- Changeability
- Extendability
- Re-usability
- Portability

****

How do we find good weight functions.

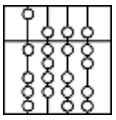
Achievements In Software Quality



TECHNISCHE UNIVERSITÄT MÜNCHEN

- Very valuable work in different disciplines:
  - formal verification
  - testing
  - security
  - performance engineering
  - software architecture
  - modelling techniques
  - usability engineering
  - …
- Interconnections are not well understood

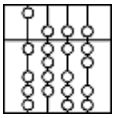
An example: Efficiency



TECHNISCHE UNIVERSITÄT MÜNCHEN

- How do we define efficiency?
  - Engineers: algorithmic complexity (O-Notation)?
  - Customers: response times?
- What about …?
  - space/time trade-off
  - throughput
  - latency
  - utilization
- Even the intuitive facet performance turns out complex
- What to expect from a performance requirement spec?

Quality Assessment



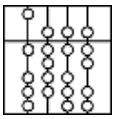
What do we need for quality assessment?

- Sound criteria
- Pro-active quality controlling process
- Tool support

We always need a profile of the application domain:

Requirements engineering

Sound Criteria



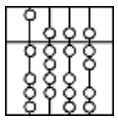
- Criteria need to be
  - justified
    - sound theoretical foundation (not only “seen elsewhere”)
    - interdependencies between criteria must be well understood
  - checkable

- Counter Examples
  - Cyclomatic Complexity < 20
  - Acyclic dependency graph

  } justification?

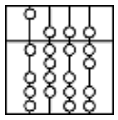
  - simplicity of interfaces
  - low complexity of structures
  - complete documentation

  } detailing / checking?



Part I: Quality Models

CHAIR IV
Software & Systems Engineering

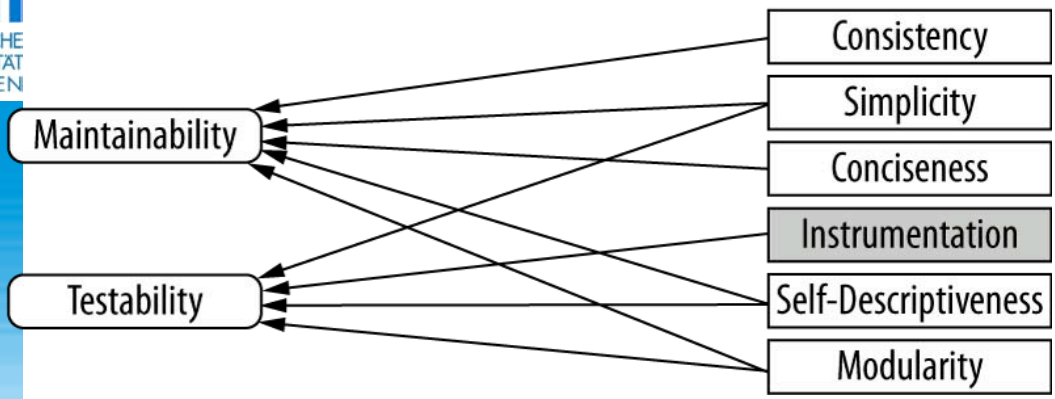


TECHNISCHE UNIVERSITÄT MÜNCHEN

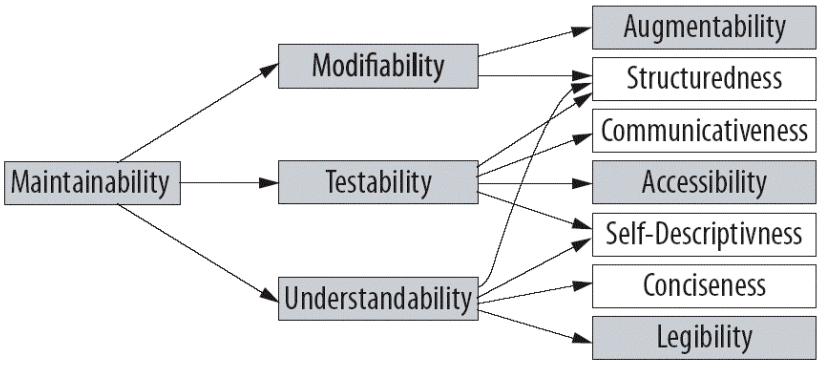
Quality Models

- Quality model: abstract definition of the important attributes for quality
- Basis for the definition of quality requirements
- Structured quality assessments
- Typically adapted to organisation, project, domain, …
- Standard: ISO 9126

The Model of McCall



The Model of Boehm et al.



Approach

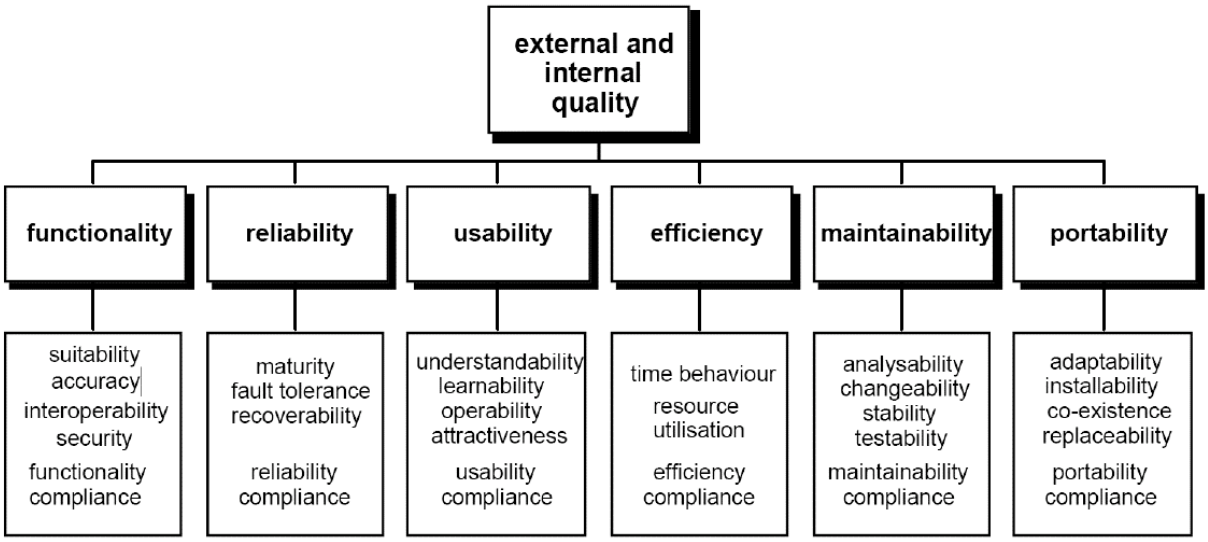
- Break-down of quality criteria
- Tree-like structure

Problem

- Characteristics vs activities
- Ambiguous edge semantic

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

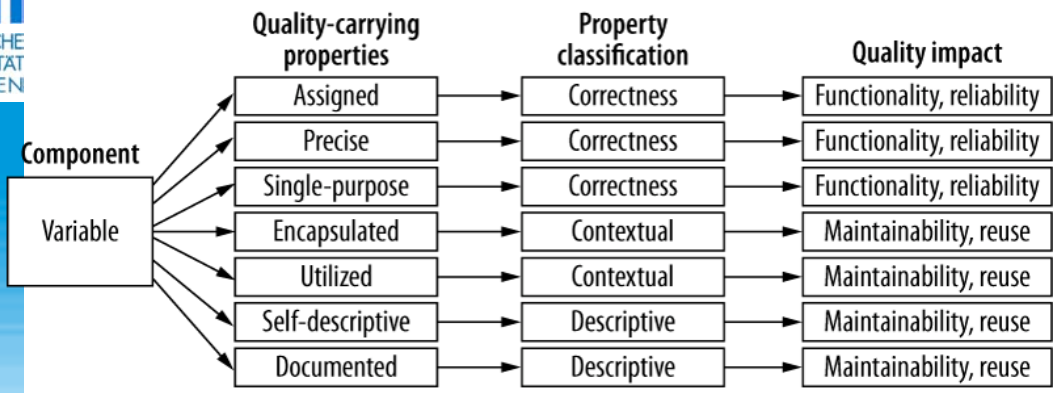
ISO 9126



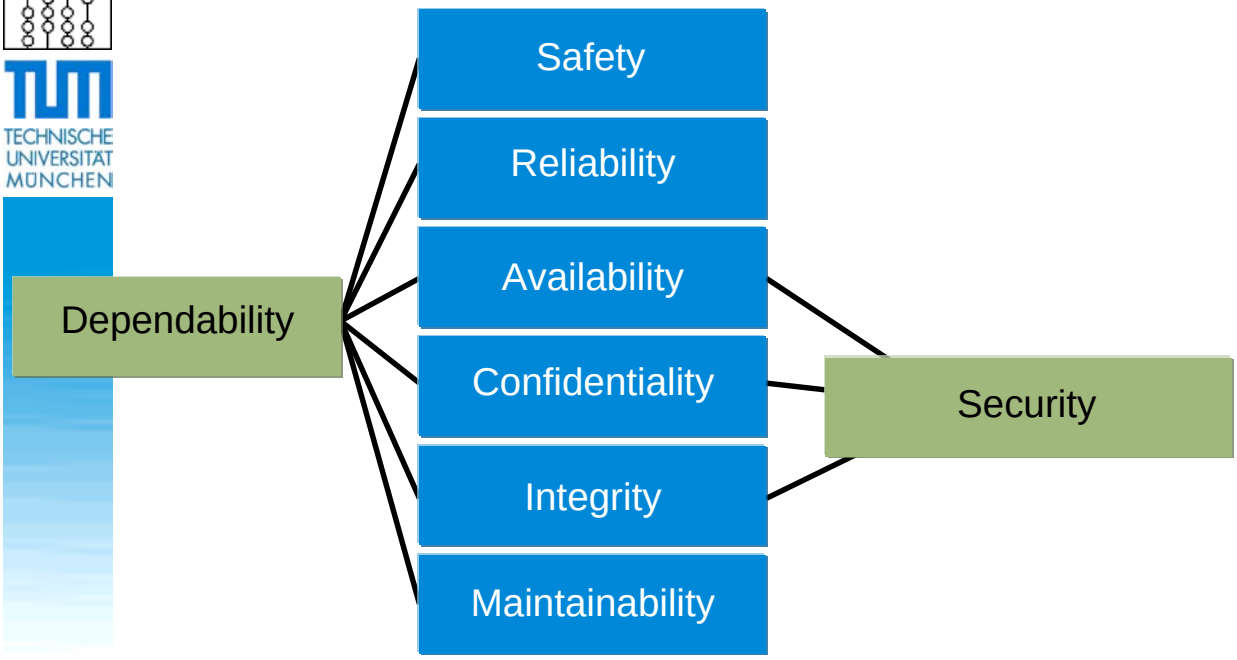
ISO 9126, 2003

The Model of Dromey

TECHNISCHE UNIVERSITÄT MÜNCHEN



Dependability Model [Avizienis et al.]

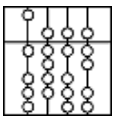


[A. Avizienis et al. Basic concepts and taxonomy of dependable and secure computing. TSD, 1994.]

Problems

- Unclear Semantics
- Assessability
- Justification
- Operationalisation
- Quality Economics

Quality Models as Living Artefacts

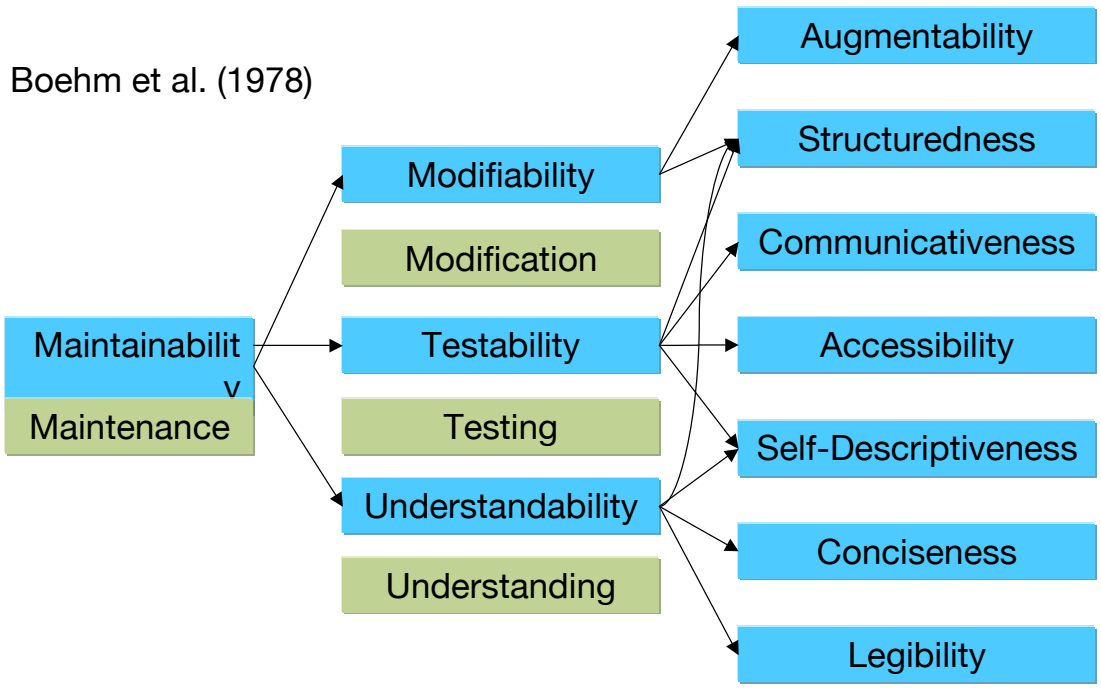


TECHNISCHE UNIVERSITÄT MÜNCHEN

- Models structure quality definitions
- Often based on ISO 9126
- Such standards are only structuring quality on a high level
- In practical project work, more detail is needed
- It needs to be adapted to
  - Domain
  - Project type
  - Size
  - Technology

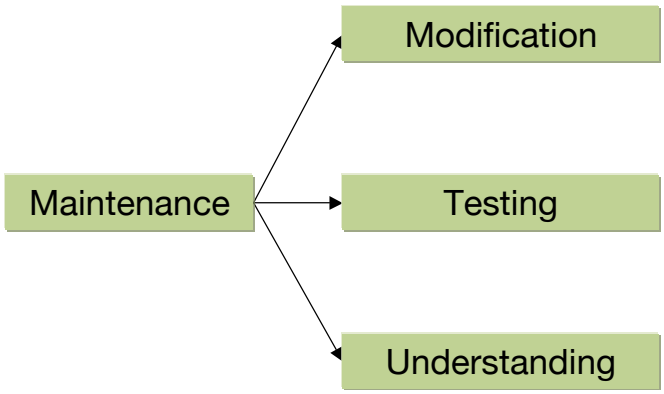
Activity-Based Quality Models

Boehm et al. (1978)



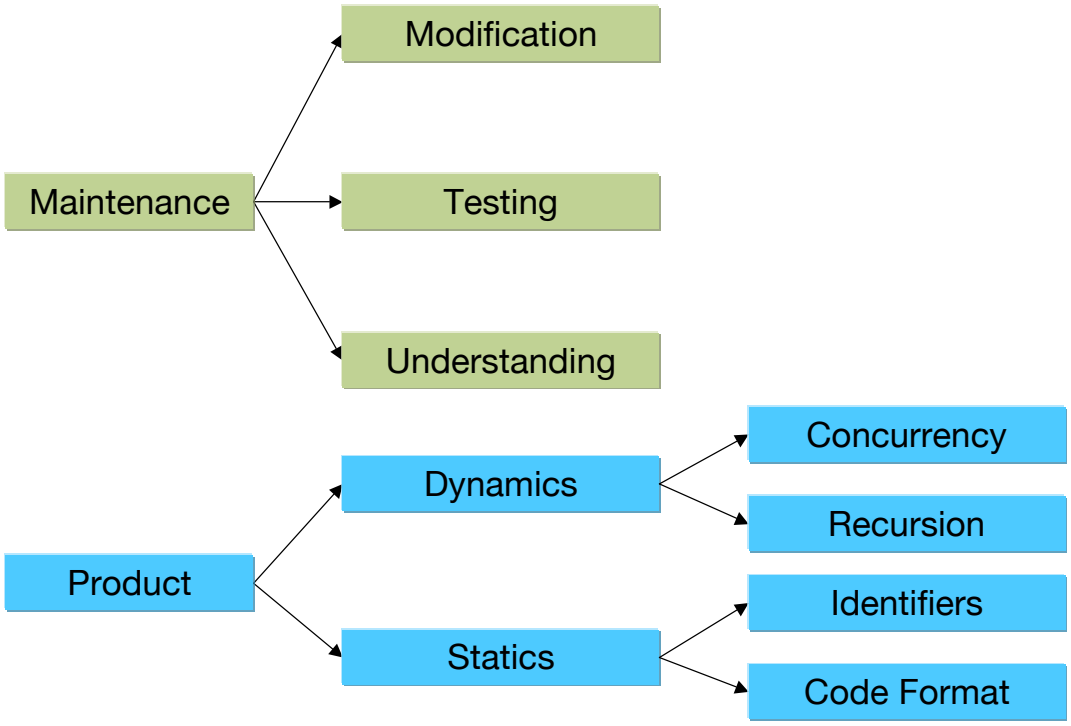
[F. Deissenboeck et al. ICSM. IEEE CS Press, 2007.]

Activity-Based Quality Models



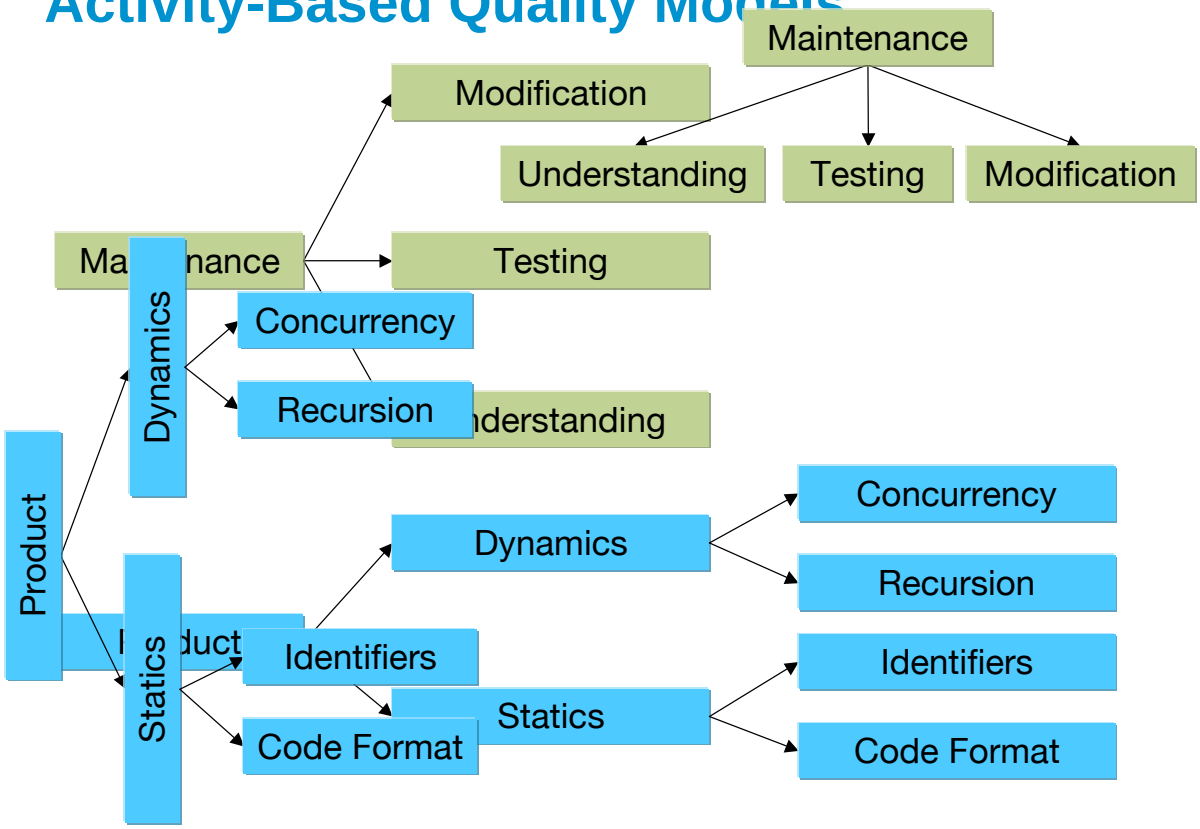
[F. Deissenboeck et al. ICSM. IEEE CS Press, 2007.]

Activity-Based Quality Models



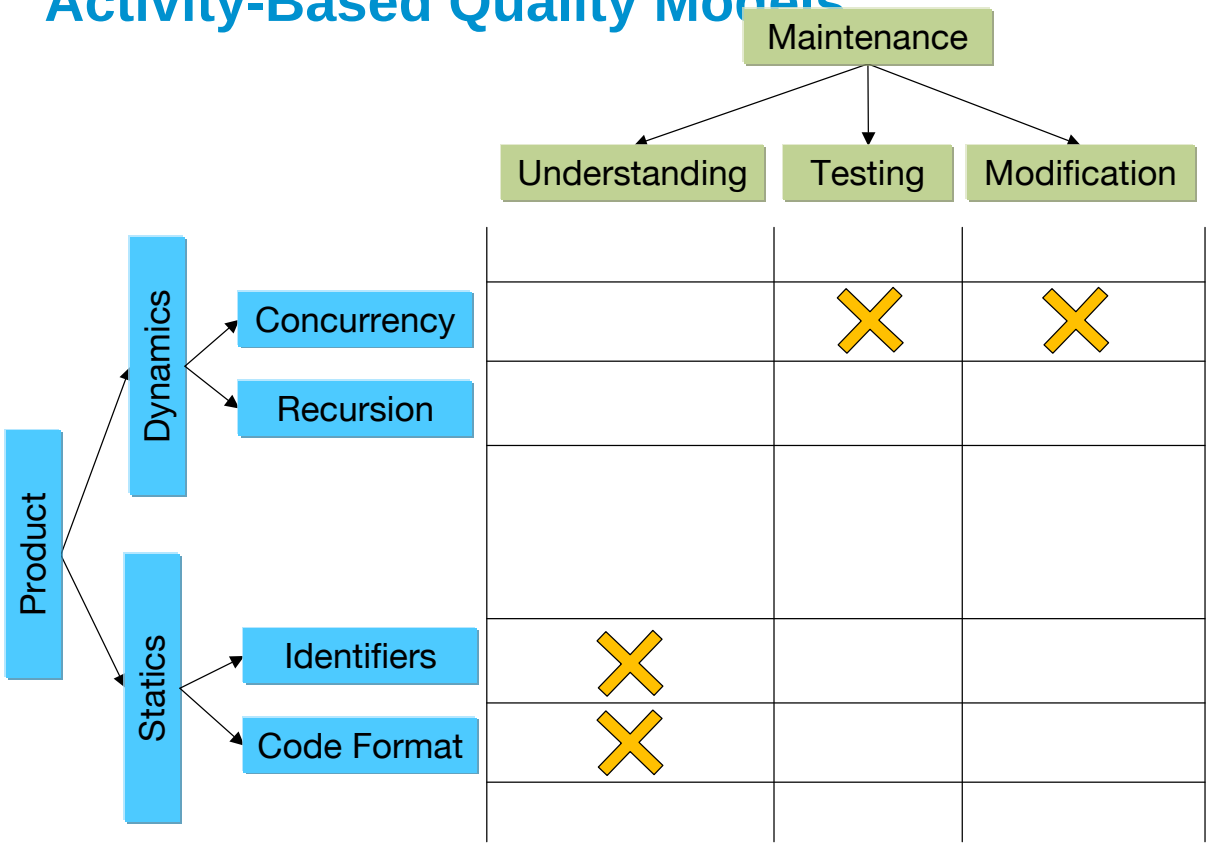
[F. Deissenboeck et al. ICSM. IEEE CS Press, 2007.]

Activity-Based Quality Models

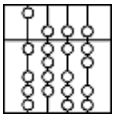


[F. Deissenboeck et al. ICSM. IEEE CS Press, 2007.]

Activity-Based Quality Models



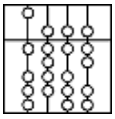
[F. Deissenboeck et al. ICSM. IEEE CS Press, 2007.]

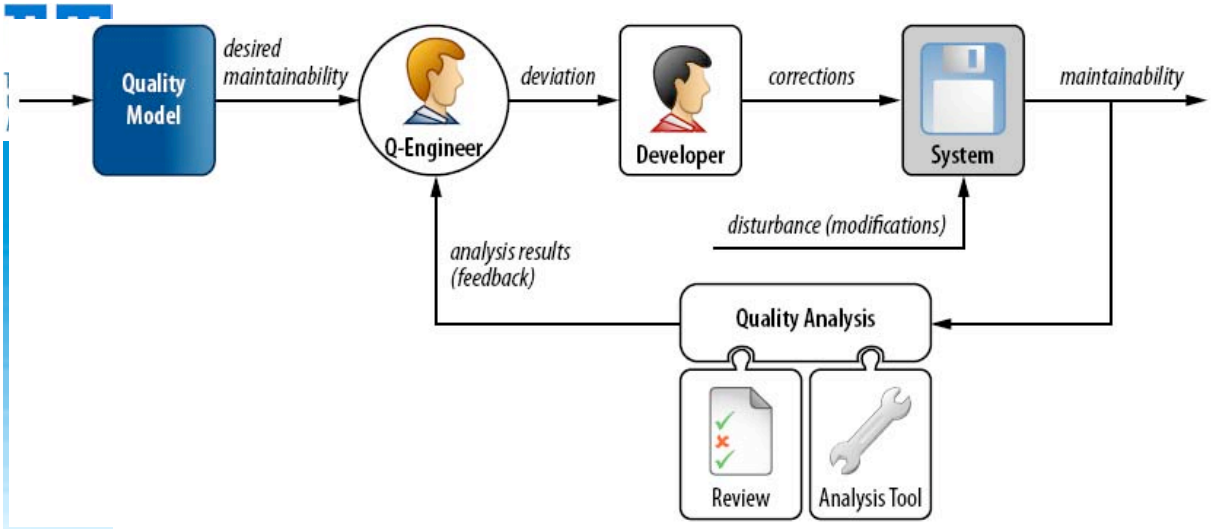


Part II: Quality Control

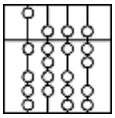
CHAIR IV
Software & Systems Engineering

Quality Control





Continuous Quality Control



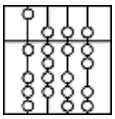
- *»By the time you figure out you have a quality problem it is probably to late to fix it.«* (Reel '92)

- Continuous, real-time quality controlling

- Enhances product quality
- Improves quality skills

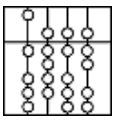
- High skills lead to better product quality

Quality Control Techniques

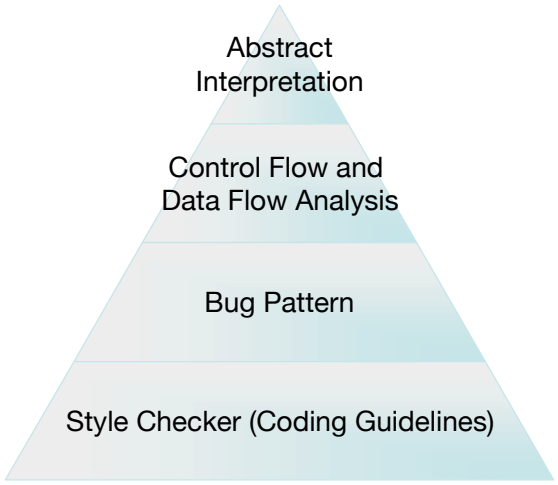


- Inspections
  - Effective and efficient
  - Effectivity
    - Are able to find up to 93% of faults
    - On overage, a third of the faults are found
  - Efficiency
    - Effort to detect a fault 1-2 person-hours / fault
    - Comparable to common testing methods
  - But
    - Also in early phases
    - Also on requirements and design documents
    - Fault removal is the least expensive
- Dynamic Tests
- Bug-finding Tools
  - Very effective in detecting maintenance-related defects
  - Low effectiveness in detecting field defects
  - Efficiency high but can be massively lowered by false positives
- Metrics and Measurement
- Quality Estimation and Predicition
  - Aggregating measurements to general quality estimates
  - Combining sets of factors to an estimation
  - Adding prediction capabilities
  - For example in reliability growth models

Automatic Static Analyses

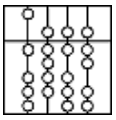


- Analysis of software by software
- But no execution of the analysed software
- Wide spectrum
- Efficient, but many *false positives*
- Examples
  - Checkstyle
  - FindBugs
  - PMD
  - Klockwork
  - Coverity



[S. Wagner et al. An Evaluation of Two Bug Pattern Tools for Java. ICST, IEEE CS Press, 2008.]

Tools and Dashboards

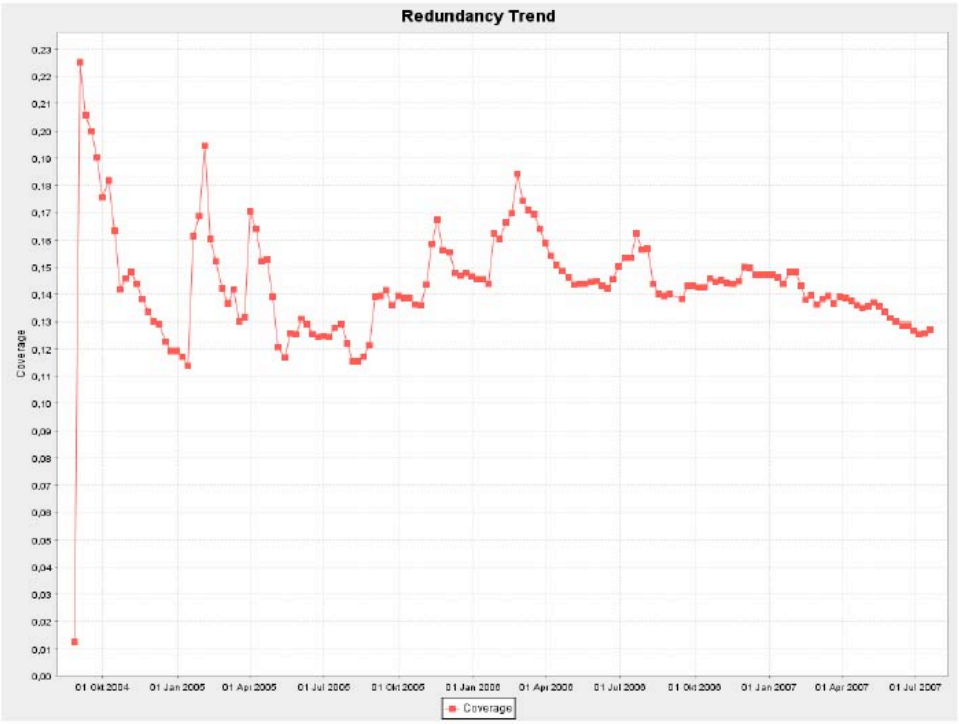


- Quality controlling is expensive
- Automation is crucial
- ConQAT (Continuous Quality Assessment Tool)

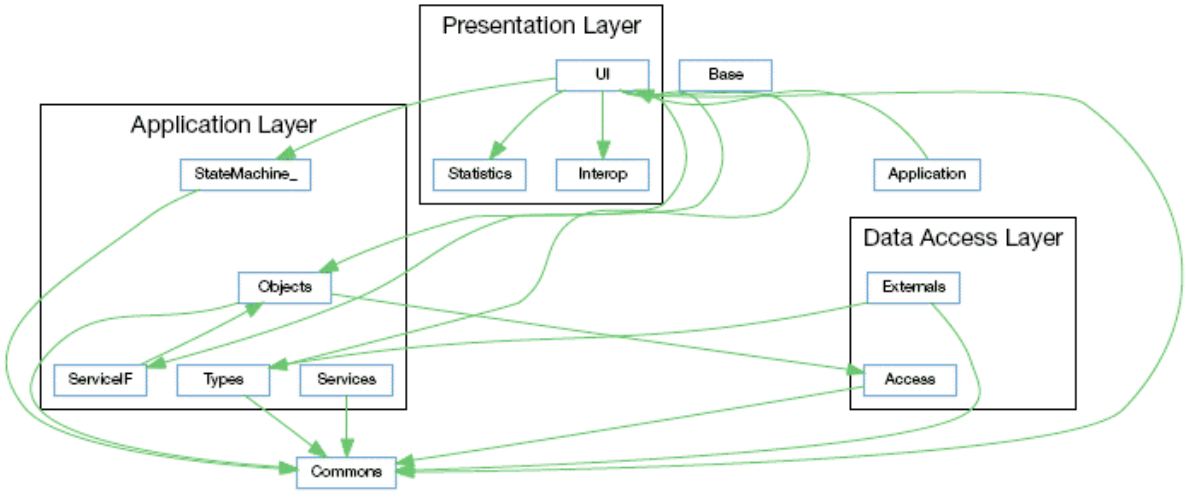
Requirements

- Integrated display of criteria
- Batch processing
- Information aggregation
- Diversity
- Extensibility
- Flexibility
- Scalability

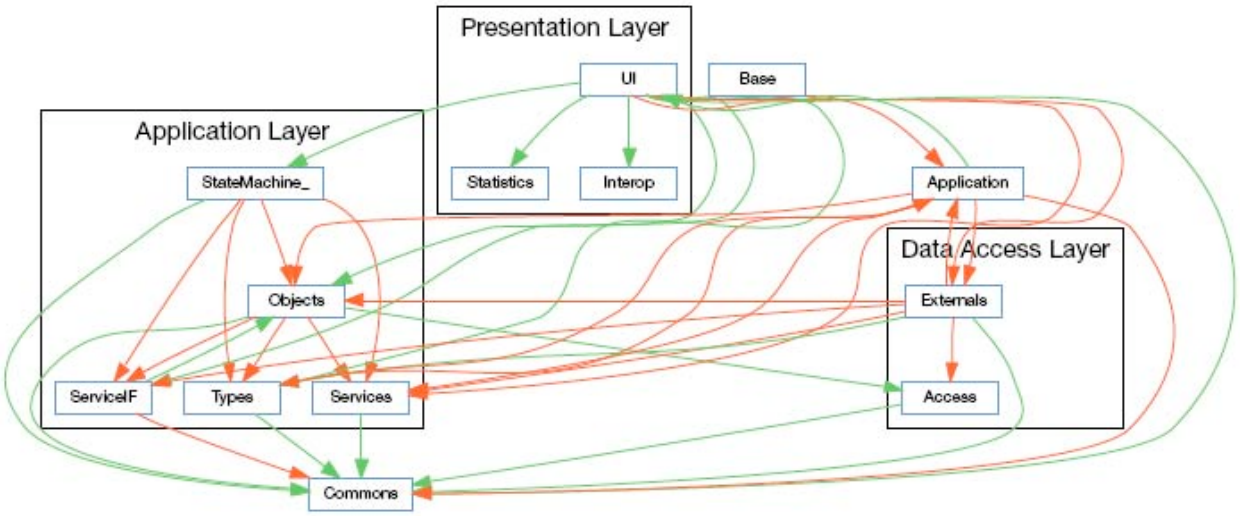
Trend Analysis



Architecture



Architecture

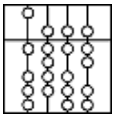


Holistic view: Quality and requirements

Is quality independent of requirements?

- The quality of a software product is closely connected to its valid requirements:
  - Functional requirements: if the software does not address the functions that are needed (or it offers a large number of functions not needed) it is hard to say that it is of high quality!
  - Quality requirements: Tailored quality - no gold plating!
    - There is no absolute quality notion!
    - Different software systems have different requirements needs!
    - Requirements have to lead to a quality profile!
    - Quality attributes that are not required but delivered increase development cost!

Conclusions - holistic SQ



TECHNISCHE UNIVERSITÄT MÜNCHEN

- There is no absolute notion of software quality
- We need a comprehensive structured approach
- A holistic approach to software quality
  - asks for quality profiles for software requirements
  - asks for a systematic scheme for evaluating software system
  - needs a deeper understanding of the relationship between constructive steps to achieve software quality and the reached software quality
- Detailed, descriptive quality models allow a clear decomposition of quality
- Quality model serves as integrated part of quality control process
- Process makes use of various quality assurance methods
- Automation in assessment, collection and aggregation
- Wide industrial experiences