UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

UNIVERSITY OF SZEGED

Department of Software Engineering

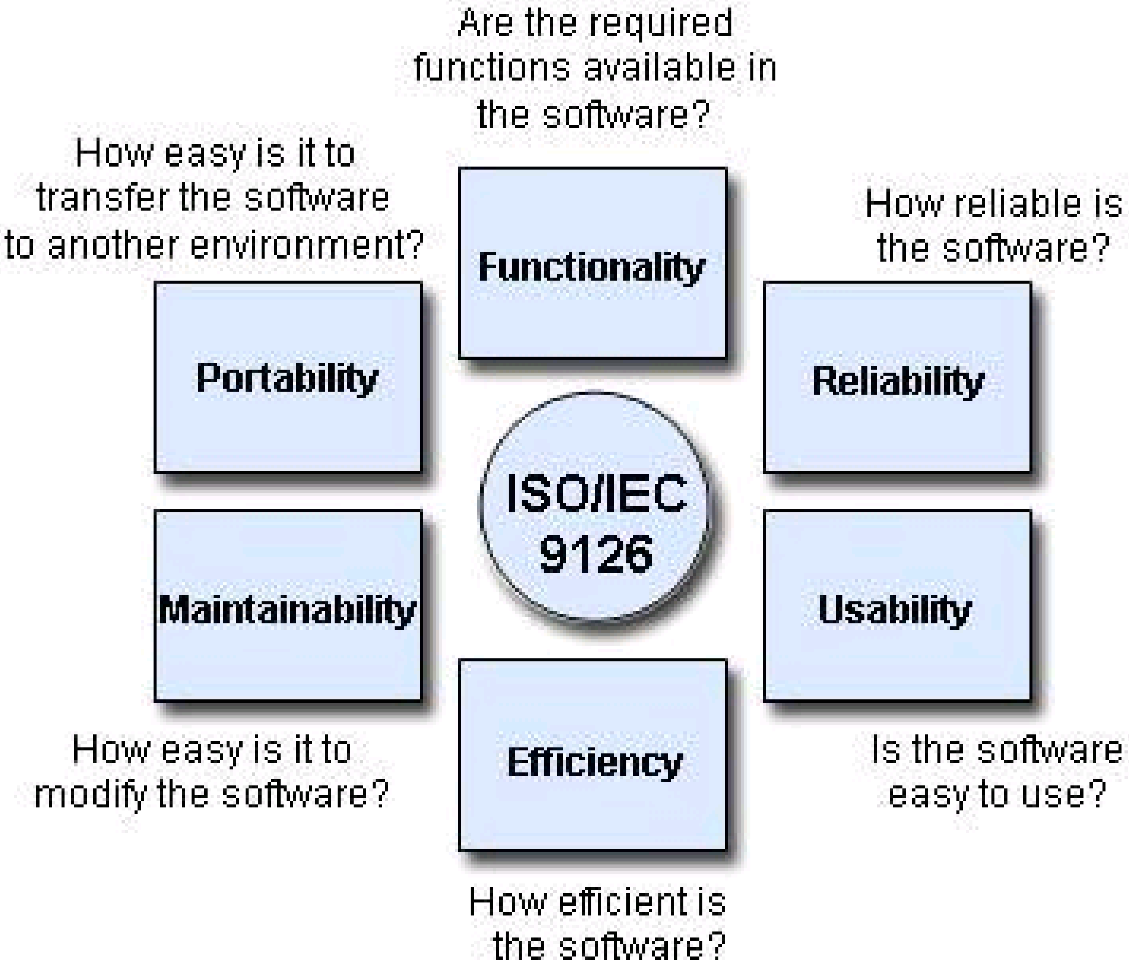
A Probabilistic Quality Model for C#

an Industrial Case Study

Péter Hegedűs

I meant "quality", not "quantity"...

ISO/IEC 9126



ISO/IEC 9126

Functionality: Are the required functions available in the software?

Reliability: How reliable is the software?

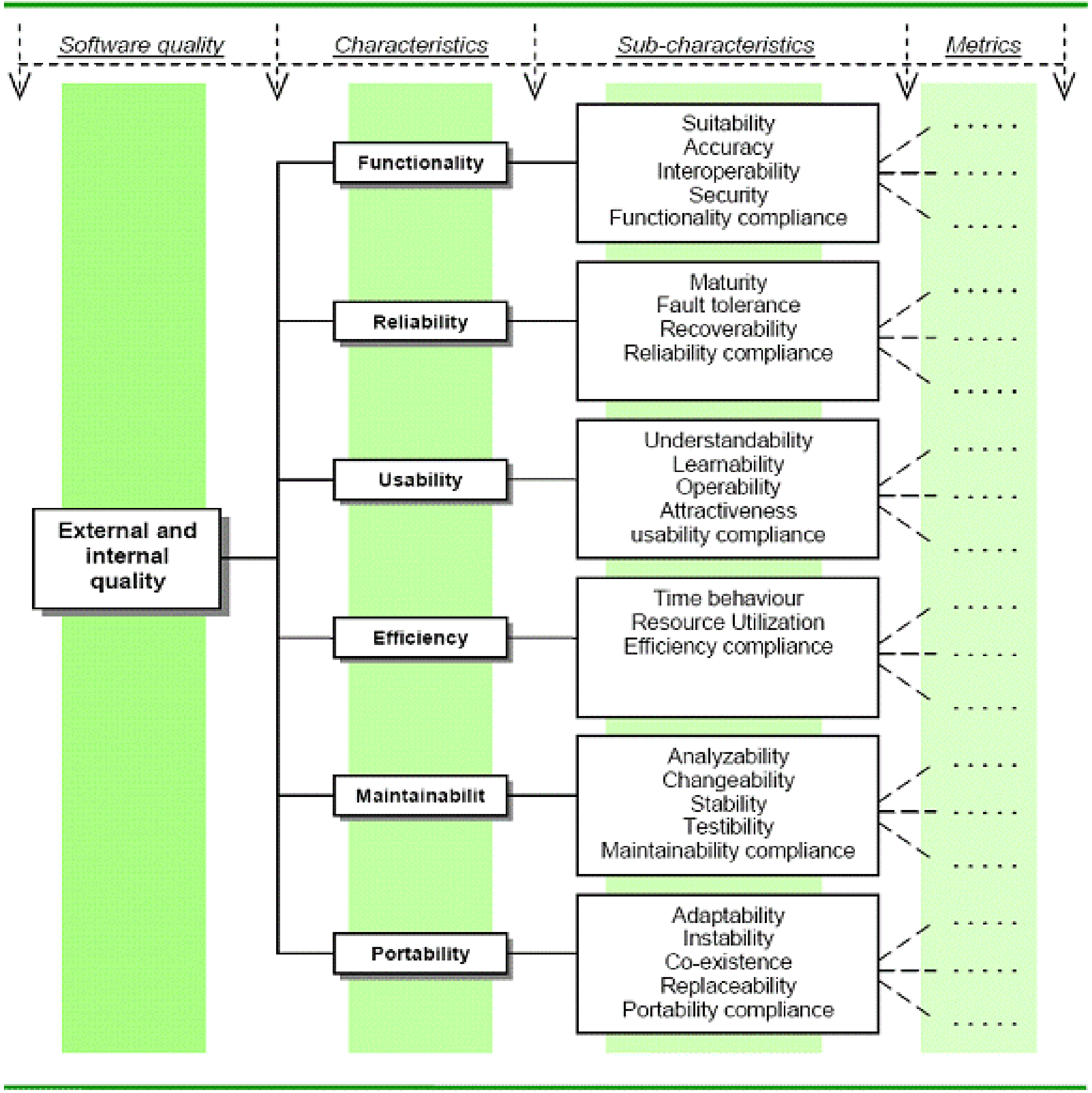
Usability: Is the software easy to use?

Efficiency: How efficient is the software?

Maintainability: How easy is it to modify the software?

Portability: How easy is it to transfer the software to another environment?

ISO/IEC 9126 Sub-characteristics



Software quality | Characteristics | Sub-characteristics | Metrics

External and internal quality

- **Functionality**: Suitability, Accuracy, Interoperability, Security, Functionality compliance
- **Reliability**: Maturity, Fault tolerance, Recoverability, Reliability compliance
- **Usability**: Understandability, Learnability, Operability, Attractiveness, usability compliance
- **Efficiency**: Time behaviour, Resource Utilization, Efficiency compliance
- **Maintainabilit**: Analyzability, Changeability, Stability, Testibility, Maintainability compliance
- **Portability**: Adaptability, Instability, Co-existence, Replaceability, Portability compliance

Using ISO9126 in Practice

- What should be the low level metrics?
- The standard suggests some metrics
  - Complete specification and system plan is required
  - Hard to calculate the values automatically
  - Hard to apply in practice
- Most of the time all we have is the source code
  - Practical, adapted models are needed

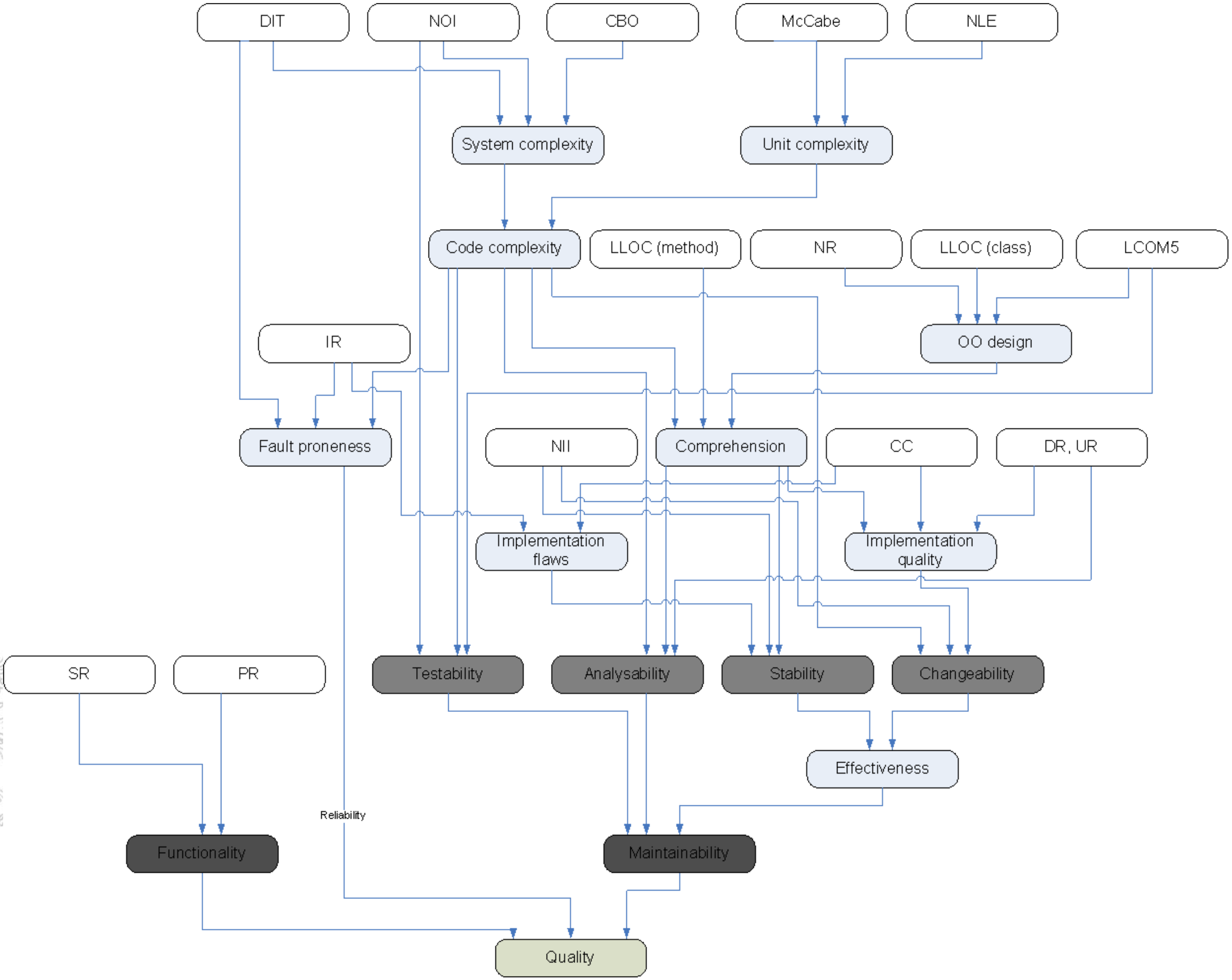
Probabilistic Quality Model

- A new approach is presented in one of our previous works:
  - Uses a benchmark as a base of the qualification
  - Integrates the ambiguity originating from different points of view of the experts
  - The method uses probabilistic distributions instead of average metric values
- A prototype model for Java has been introduced

Aims of the Current Research

- Adapt the approach to C# systems involving industrial experts
  - Creating a new quality model
  - Introducing a new weighting
- Show that our scientific results are applicable in a real industrial environment
  - Involving an industrial partner
- Validate the results of the quality model by comparing to the opinion of experts
  - Manual validation of the qualifications

The C# Quality Model

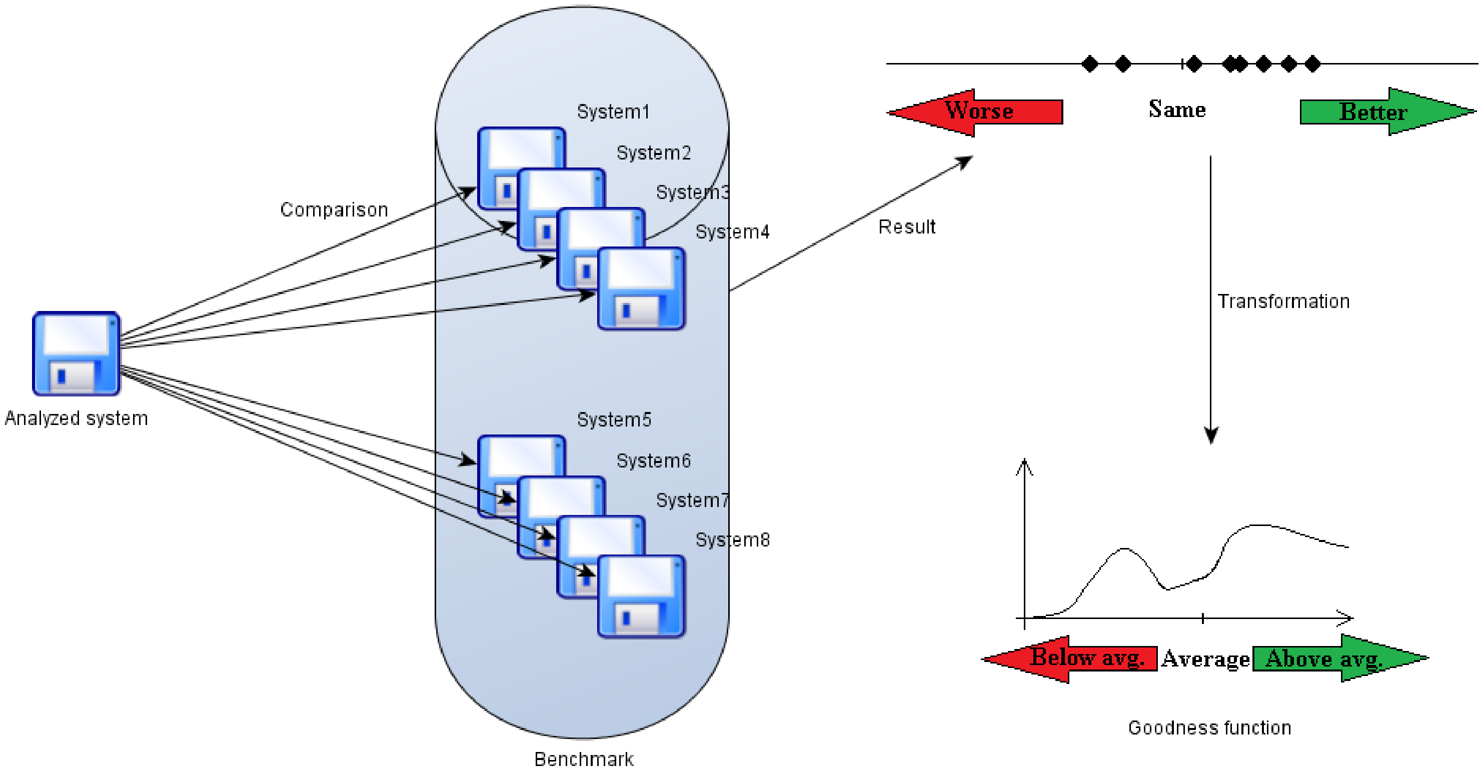


Low-level „Sensor” Metrics

- DIT – Depth of Inheritance Tree
- NOI – Number of Outgoing Invocations
- CBO – Coupling Between Object Classes
- McCabe – McCabe's cyclomatic complexity
- NLE – Nesting level (else if)
- LLOC (method) – Logical Lines of Code in Methods
- LLOC (class) – Logical Lines of Code in Classes
- LCOM5 – Lack of Cohesion On Methods
- NII – Number of Incoming Invocations
- CC – Clone Coverage
- FxCop Rule Violations
  - NR – Naming Rules (FxCop)
  - IR – Interoperability Rules (FxCop)
  - DR, UR – Design Rules, Usage Rules (FxCop)
  - SR – Security Rules (FxCop)
  - PR – Performance Rules (FxCop)

Evaluation of the Sensor Nodes

- Comparing the metric values with the values of the benchmark systems -> „goodness function”

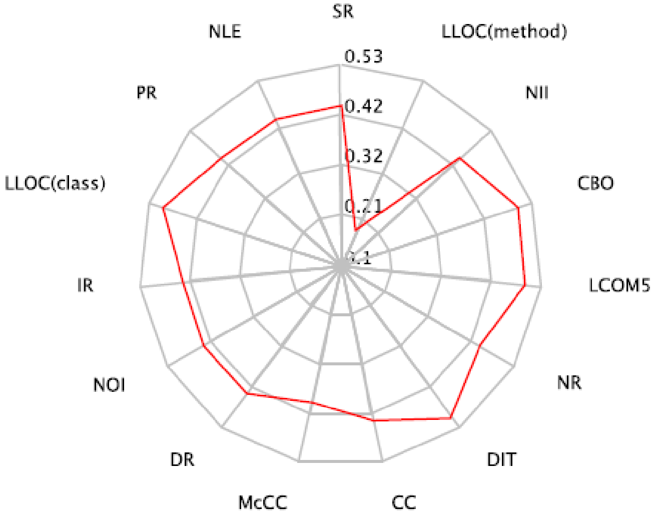


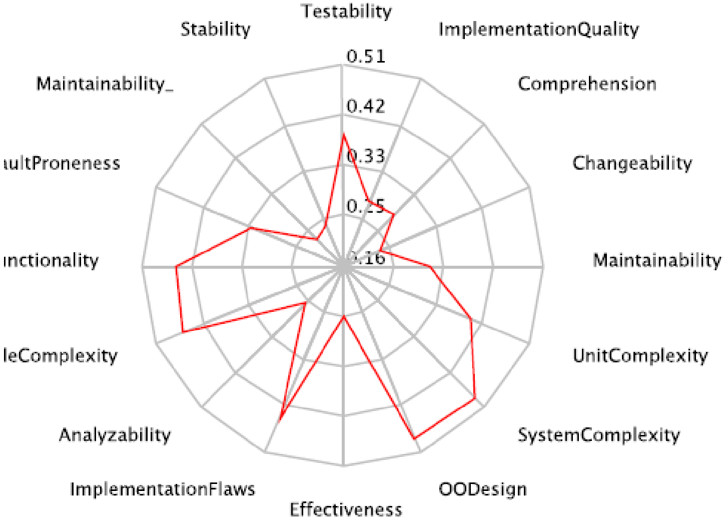
Case Study Setup

- Quality model
  - Presented above for C#
- Benchmark
  - All of the partner’s C# components
    - Over 300 components (dll and exe) analyzed by the Columbus toolset
    - *TLLOC:* 711944; *TNCL:* 4942; *TNM:* 48787
    - An „in house” qualification (components are compared against each other)
- Weights by industrial partners and SED

Case Study Results

- 10 selected components were evaluated
- A sample result
  - Quality: 0.311





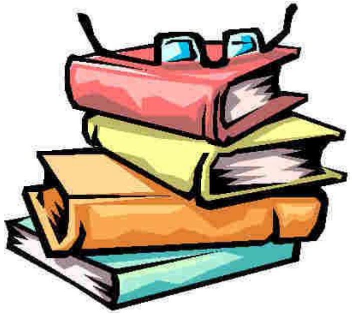
Validation of the Results

- Correlation of the QM results and the expert votes
  - 0,923

QM result	0,311	0,261	0,261	0,261	0,26	0,26	0,221	0,221	0,216	0,178
Avg. human vote	0,56	0,48	0,473	0,53	0,47	0,49	0,4	0,44	0,45	0,3

- The QM and human scale is different
- The correlation is very high
- The industrial partners fully agreed with the results

Related Publications



- Tibor Bakota, Péter Hegedűs, Péter Körtvélyesi, Rudolf Ferenc, and Tibor Gyimóthy. *A Probabilistic Software Quality Model*
  - ICSM 2011 conference
- Tibor Bakota, Péter Hegedűs, Gergely Ladányi, Péter Körtvélyesi, Rudolf Ferenc, and Tibor Gyimóthy. *A Cost Model Based on Software Maintainability*, accepted, to appear
  - ICSM 2012 conference

Acknowledgement

ÚJ SZÉCHENYI TERV

- *The publication/presentation is supported by the European Union and co-funded by the European Social Fund.*
- *Project title: "Broadening the knowledge base and supporting the long term professional sustainability of the Research University Centre of Excellence at the University of Szeged by ensuring the rising generation of excellent scientists."*
- *Project number:*

TÁMOP-4.2.2/B-10/1-2010-0012

National Development Agency
www.ujszechenyiterv.gov.hu
06 40 638 638

HUNGARY'S RENEWAL

The project is supported by the European Union and co-financed by the European Social Fund.

QUESTIONS ???