Software Improvement Group

Software Product Quality and its Effects

Joost Visser

November 2011

info@sig.eu
www.sig.eu

Software Improvement Group

Background

- Spin-off from CWI in 2000, self-owned, independent
- Strong academic background, innovative, award-winning, profitable

Activity

- Management advisory, fact-based
- Accredited software analysis lab employs analysis tools and models
- Experienced staff transforms analysis data into advice

Track record

- Finance, government, logistics, telecom, manufacturing, energy, …
- We analyze over 100 systems annually

Who is using our services?

Software Improvement Group

Financials & insurance companies

Public

Retail/Logistics

Technology

Utilities/Telco

Selected services

Software Improvement Group

Software Risk Assessment

- In-depth investigation of software quality and risks
- Answers specific research questions

Software Monitoring

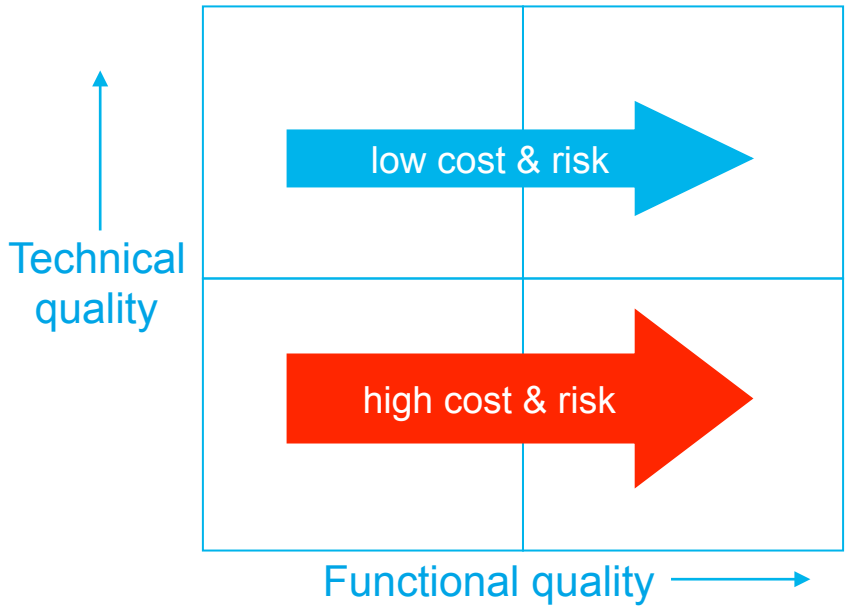
- Continuous measurement, feedback, and decision support
- Guard quality from start to finish

Software Product Certification

- Five levels of technical quality
- Evaluation by SIG, certification by TÜV Informationstechnik

Functional *vs* technical quality

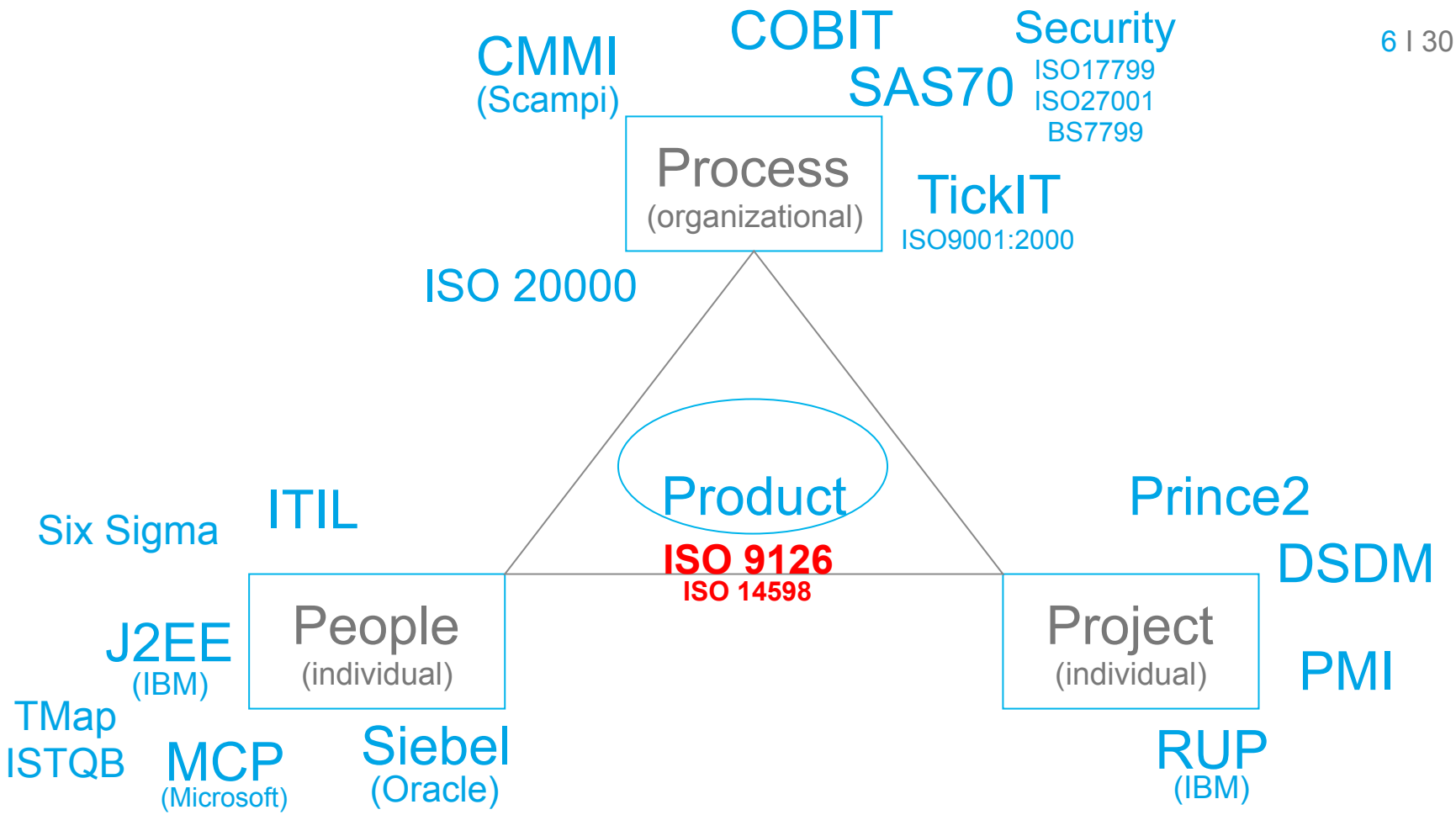
Software Improvement Group



Software with high technical quality can evolve with low cost and risk to keep meeting functional and non-functional requirements.

The Bermuda Triangle of Software Quality

Software Improvement Group



Software product quality standards

ISO/IEC 9126

Software engineering -- Product quality

1. Quality model
2. External metrics
3. Internal metrics
4. Quality in use metrics

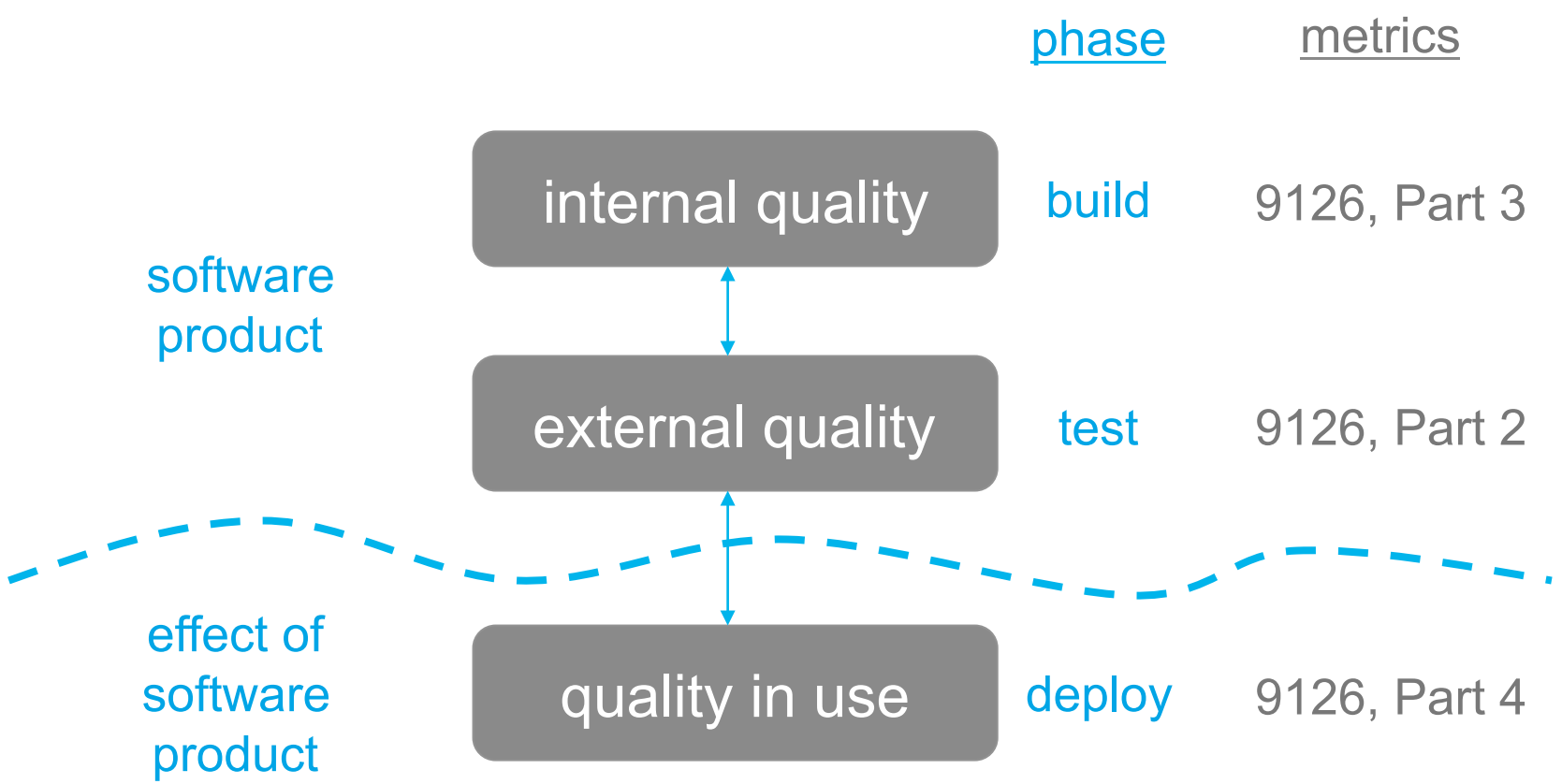
ISO/IEC 14598

Information technology -- Software product evaluation

1. General overview
2. Planning and management
3. Process for developers
4. Process for acquirers
5. Process for evaluators
6. Documentation of evaluation modules

ISO/IEC 9126, Part 1
Quality perspectives

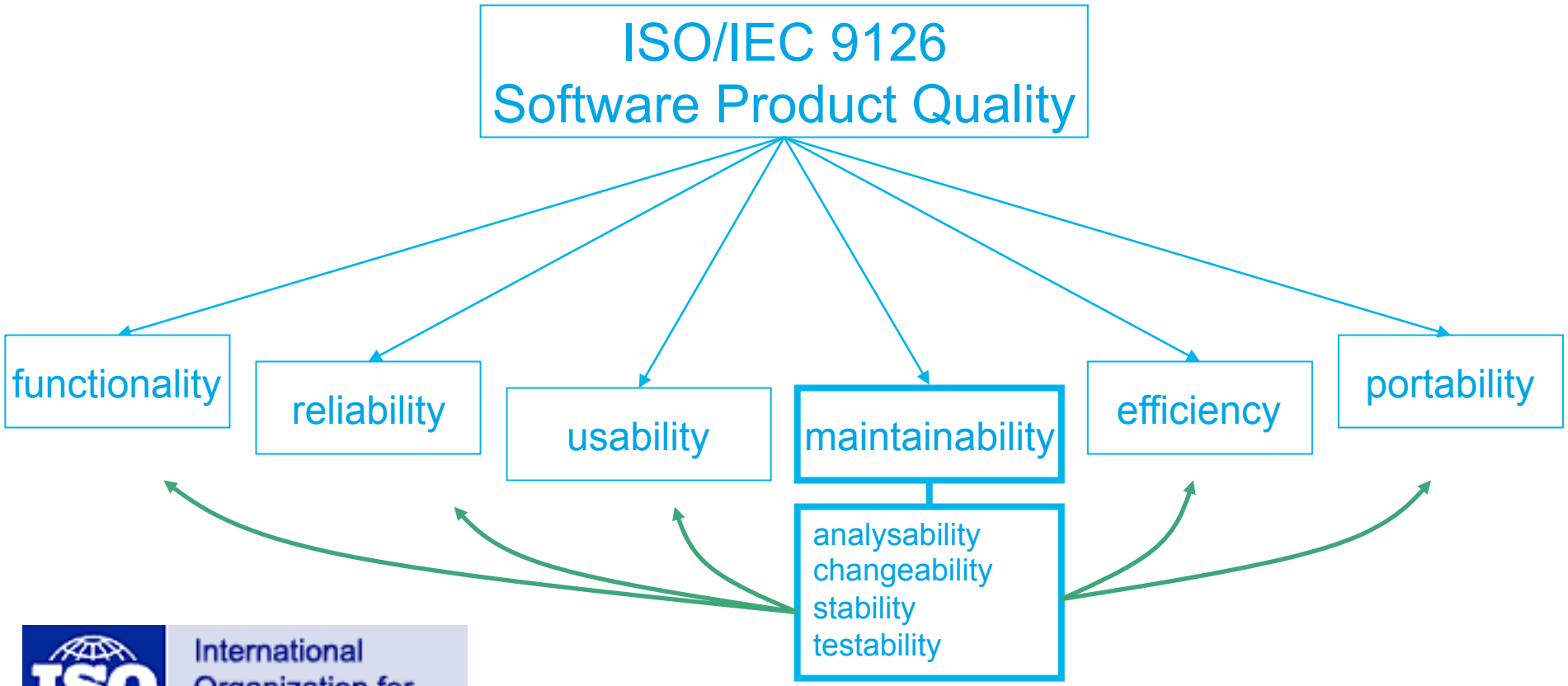
Software Improvement Group



ISO/IEC 9126, Part 1

Software product quality characteristics

Software Improvement Group



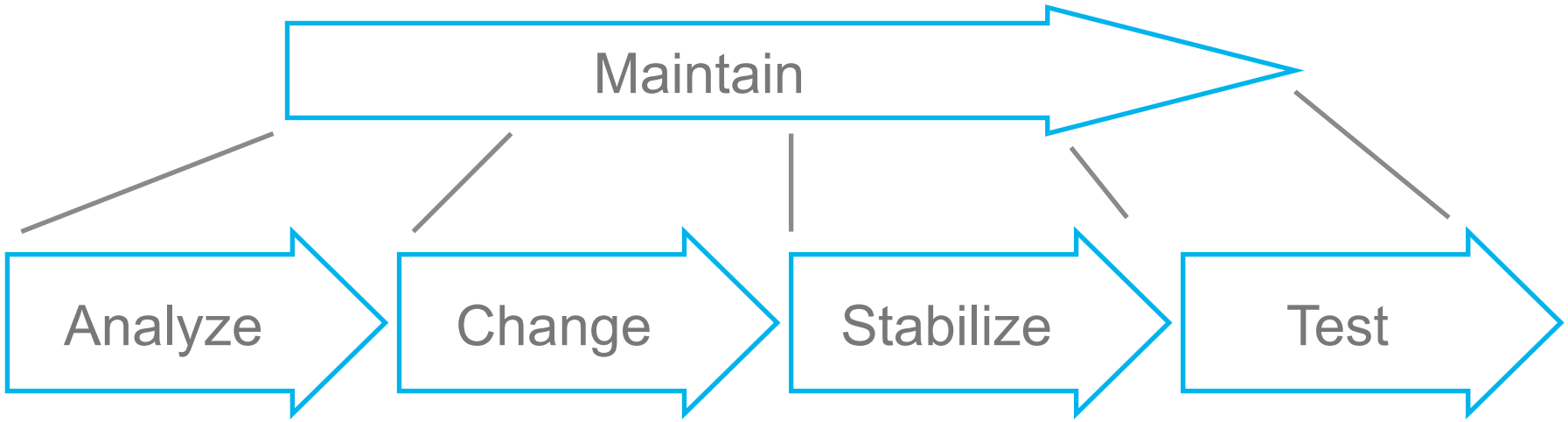
ISO International Organization for Standardization

Software Improvement Group

ISO/IEC 9126: Software Engineering - Product Quality

***Maintainability* =**

- *Analyzability*: easy to understand where and how to modify?
- *Changeability*: easy to perform modification?
- *Stability*: easy to keep coherent when modifying?
- *Testability*: easy to test after modification?



ISO 9126
Part 2,3: metrics

Software Improvement Group

External metrics, e.g.:

- Changeability: “change implementation elapse time”,
  time between diagnosis and correction
- Testability: “re-test efficiency”, time between correction and conclusion of test

Internal metrics, e.g.:

- Analysability: “activity recording”,
  ratio between actual and required number of logged data items
- Changeability: “change impact”,
  number of modifications and problems introduced by them

Critique

- Not pure *product* measures, rather *product in its environment*
- Measure *after* the fact

A Challenge

Software Improvement Group

Use source code metrics to measure technical quality?

Plenty of metrics defined in literature

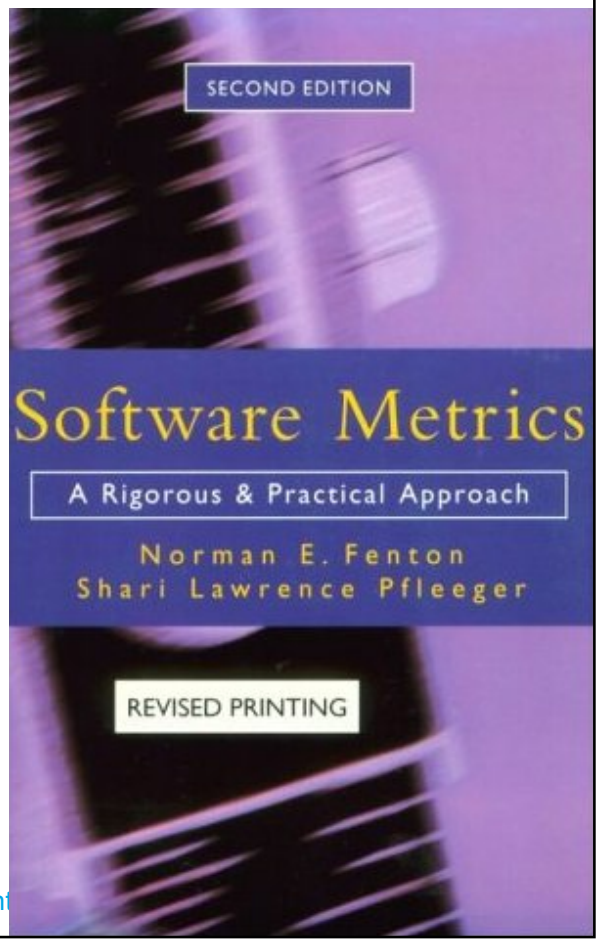
- LOC, cyclomatic complexity, fan in/out, coupling, cohesion, …
- Halstead, Chidamber-Kemener, Shepperd, …

Plenty of tools available

- Variations on Lint, PMD, FindBugs, …
- Coverity, FxCop, Fortify, QA-C, Understand, …
- Integrated into IDEs

But:

- Do they measure technical quality of a system?



Source code metrics
Cyclomatic complexity

Software Improvement Group

- T. McCabe, *IEEE Trans. on Sw Engineering*, 1976
- Accepted in the software community
- Academic: number of independent paths per method
- Intuitive: number of decisions made in a method
- Really, the number of if statements (and while, for, ...)
- Software Engineering Institute:

Table 4: Cyclomatic Complexity

Cyclomatic Complexity	Risk Evaluation
1-10	a simple program, without much risk
11-20	more complex, moderate risk
21-50	complex, high risk program
greater than 50	untestable program (very high risk)

```
IF FHFI-R21015-E60676TLV = 'ARC'
   IF FHFI-R21015-E60677    = 'J'
      IF FHFI-R21015-E606767GV = 'ARC'
         MOVE FHFI-R21015-E60111 TO W-D12-V1    END-IF
      IF FHFI-R21015-E606767GV = 'ASP'
         MOVE FHFI-R21015-E60111 TO W-D12-V2    END-IF
      IF FHFI-R21015-E606767GV = 'ALN'
         MOVE FHFI-R21015-E60111 TO W-D12-V3    END-IF
      IF FHFI-R21015-E606767GV = 'CKS'
         MOVE FHFI-R21015-E60111 TO W-D12-V4    END-IF
      IF FHFI-R21015-E606767GV = 'RHB'
         MOVE FHFI-R21015-E60111 TO W-D12-V5    END-IF
      IF FHFI-R21015-E606767GV = 'CPS'
         MOVE FHFI-R21015-E60111 TO W-D12-V6    END-IF
      IF FHFI-R21015-E606767GV = 'TR ' OR
         FHFI-R21015-E606767GV = ' TR'
         MOVE FHFI-R21015-E60111 TO W-D12-V7    END-IF
      IF FHFI-R21015-E606767GV = 'VR ' OR
         FHFI-R21015-E606767GV = ' VR'
         MOVE FHFI-R21015-E60111 TO W-D12-V8    END-IF
      IF FHFI-R21015-E606767GV = 'FA ' OR
         FHFI-R21015-E606767GV = ' FA'
         MOVE FHFI-R21015-E60111 TO W-D12-V9    END-IF
      IF FHFI-R21015-E606767GV = 'EX ' OR
         FHFI-R21015-E606767GV = ' EX'
         MOVE FHFI-R21015-E60111 TO W-D12-V10   END-IF
   ELSE
      IF FHFI-R21015-E606767GV = 'ARC'
         MOVE FHFI-R21015-E60111 TO W-D11-V1    END-IF
      IF FHFI-R21015-E606767GV = 'ASP'
         MOVE FHFI-R21015-E60111 TO W-D11-V2    END-IF
      IF FHFI-R21015-E606767GV = 'ALN'
         MOVE FHFI-R21015-E60111 TO W-D11-V3    END-IF
      IF FHFI-R21015-E606767GV = 'CKS'
         MOVE FHFI-R21015-E60111 TO W-D11-V4    END-IF
      IF FHFI-R21015-E606767GV = 'RHB'
         MOVE FHFI-R21015-E60111 TO W-D11-V5    END-IF
      IF FHFI-R21015-E606767GV = 'CPS'
         MOVE FHFI-R21015-E60111 TO W-D11-V6    END-IF
      IF FHFI-R21015-E606767GV = 'TR ' OR
         FHFI-R21015-E606767GV = ' TR'
         MOVE FHFI-R21015-E60111 TO W-D11-V7    END-IF
      IF FHFI-R21015-E606767GV = 'VR ' OR
         FHFI-R21015-E606767GV = ' VR'
         MOVE FHFI-R21015-E60111 TO W-D11-V8    END-IF
      IF FHFI-R21015-E606767GV = 'FA ' OR
         FHFI-R21015-E606767GV = ' FA'
         MOVE FHFI-R21015-E60111 TO W-D11-V9    END-IF
      IF FHFI-R21015-E606767GV = 'EX ' OR
         FHFI-R21015-E606767GV = ' EX'
         MOVE FHFI-R21015-E60111 TO W-D11-V10   END-IF
   END-IF
END-IF
```

Source code metrics
Coupling

Software Improvement Group

- Efferent Coupling (Ce)
  - How many classes do I depend on?
- Afferent Coupling (Ca)
  - How many classes depend on me?
- Instability = Ce/(Ca+Ce) ∈ [0,1]
  - Ratio of efferent *versus* total coupling
  - 0 = very stable = hard to change
  - 1 = very instable = easy to change

Figure 1. Coupling graph

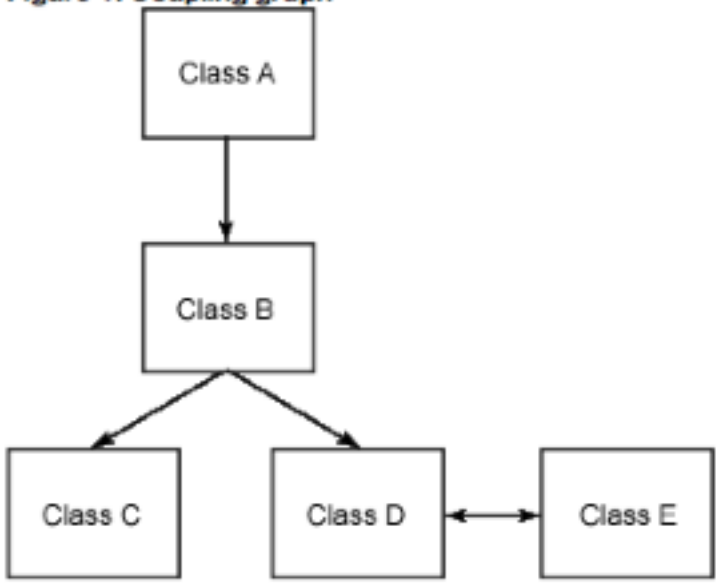
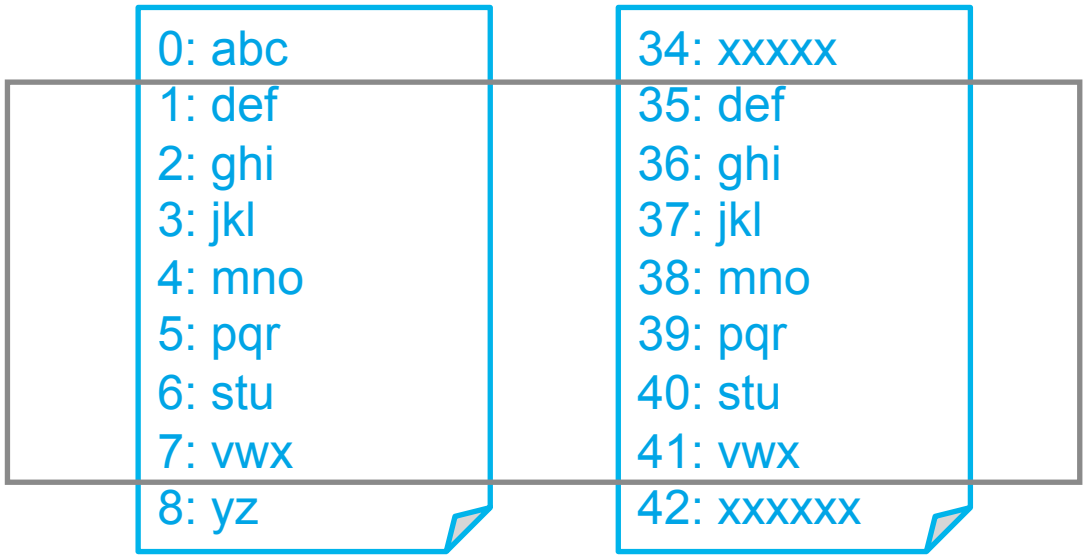


Table 1. Results of compiling a single class

Class to Compile	Other Classes Compiled	Afferent Couplings	Efferent Couplings	Instability Factor
A	B,C,D,E	0	4	1
B	C,D,E	1	3	0.75
C	-	2	0	0
D	E	3	1	0.25
E	D	3	1	0.25

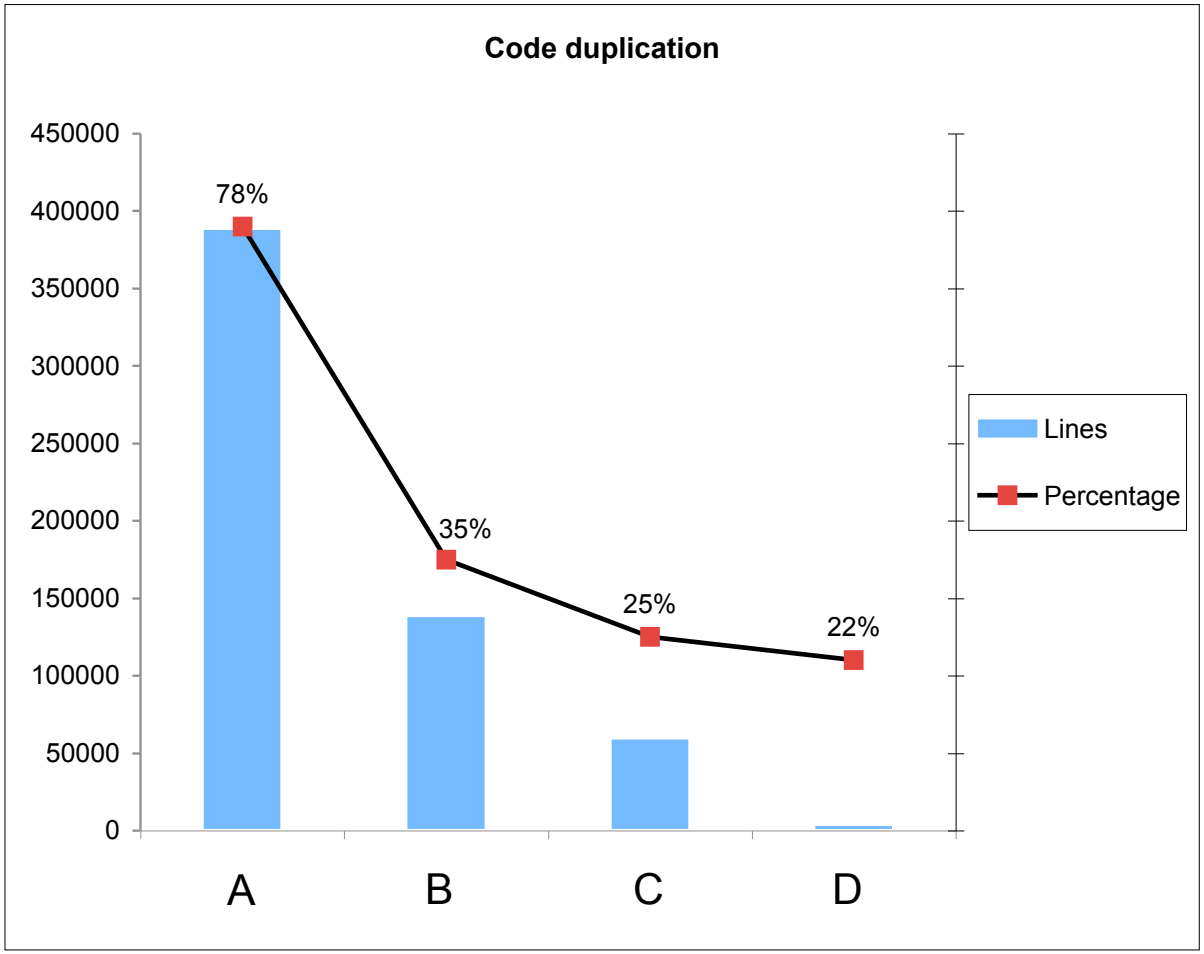
Software Improvement Group

Code duplication measurement



Number of duplicated lines: 14

Code duplication

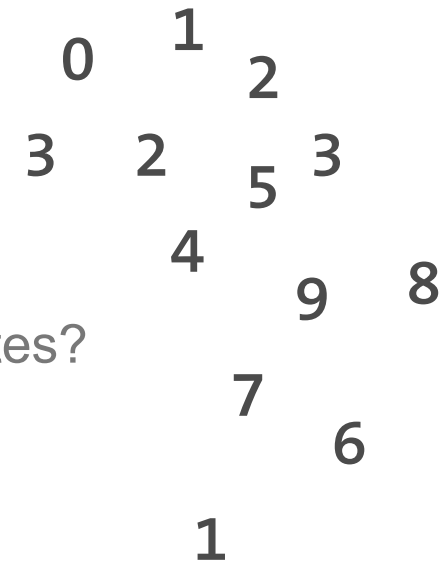


How to measure?
Software metrics crisis

Software Improvement Group

Plethora of software metrics

- Ample definitions in literature
- Ample tools that calculate



Measurement yields data, not information

- How to aggregate individual measurement values?
- How to map aggregated values onto quality attributes?
- How to set thresholds?
- How to act on results?

SIG quality model handles these issues in a pragmatic way

Measurable software attributes

Volume

- How big? How much invested effort?

Duplication

- How lean or bloated? How repetitive?

Modularity

- How well organized / subdivided into parts?

Complexity

- How much logic / knowledge / decision points?

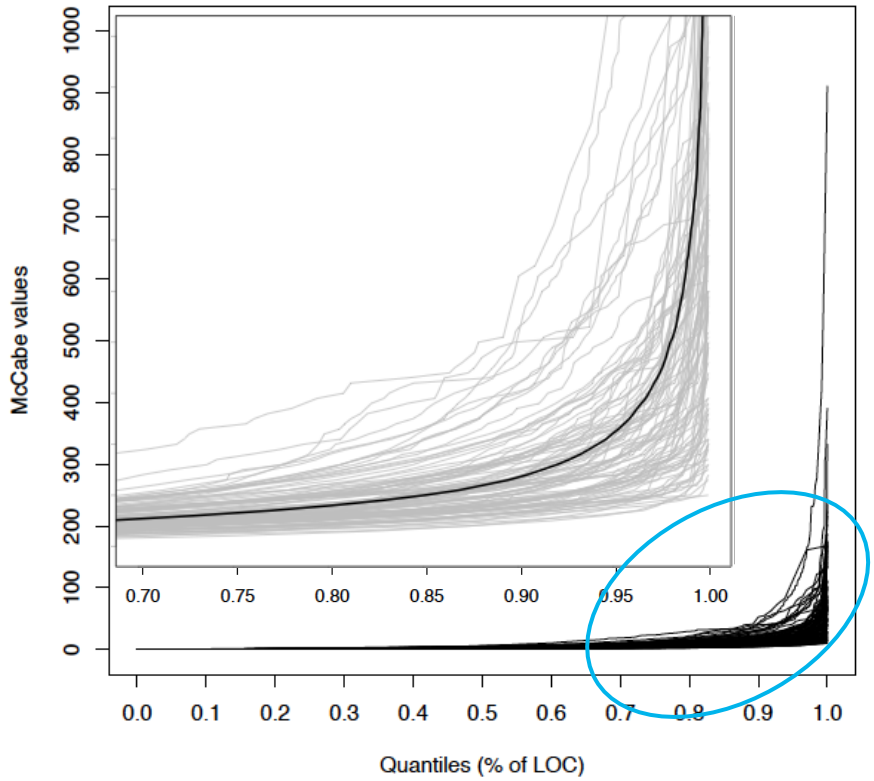
Coupling

- How many interconnections? How intricately weaved together?

The statistical nature of software metrics

Go where the variation is

Software Improvement Group



Deriving Metric Thresholds from Benchmark Data
by T. Alves, C. Ypma, J.Visser in *ICSM 2010*

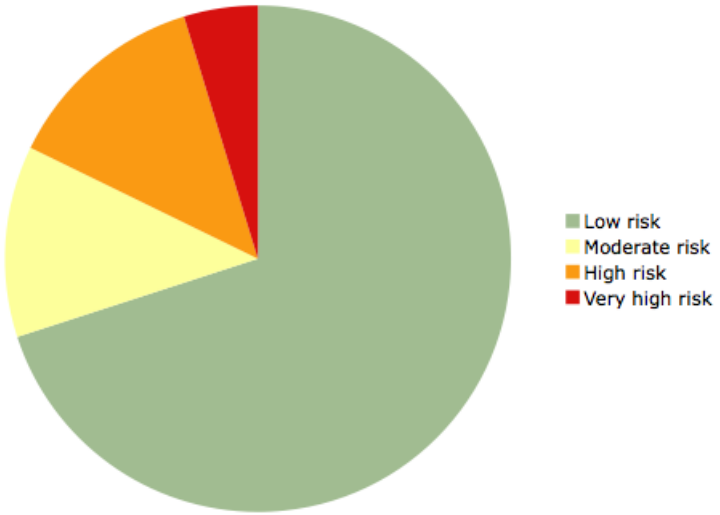
Observe for all:

- Systems are similar in low percentiles. Systems differ in higher percentiles.
- Interesting differences occur mostly above the 70% percentile

SIG Quality Model
Quality profiles

Software Improvement Group

1. Measure source code metrics per method / file / module
2. Summarize distribution of measurement values in “Quality Profiles”



Cyclomatic complexity	Risk category
1 - 10	Low
11 - 20	Moderate
21 - 50	High
> 50	Very high

Sum lines of code per category

Lines of code per risk category			
Low	Moderate	High	Very high
70 %	12 %	13 %	5 %

SIG Quality Model

How do measurements lead to ratings?

Software Improvement Group

A practical model for measuring maintainability

Heitlager, Kuipers, Visser in QUATIC 2007, IEEE Press

a. Aggregate measurements into "Quality Profiles"
b. Map measurements and quality profiles to ratings for system properties
c. Map ratings for system properties to ratings for ISO/IEC 9126 quality characteristics
d. Map to overall rating of technical quality

Maintainability Model
Standard two-phase calibration process

Software Improvement Group

1. Determine metric thresholds

- At level of metric (copybook fan-in)
- Based on metric values per file

==> four risk categories

"Deriving Metric Thresholds from Benchmark Data" by T. Alves, C. Ypma, J. Visser (SIG, U. Minho, U. Utrecht) in 26th IEEE International Conference on Software Maintenance (ICSM 2010).

2. Determine mapping to ratings

- At level of property rating (module coupling)
- Based on risk profiles for each system

==> mapping of risk profiles to property ratings

"Benchmark-based Aggregation of Metrics to Ratings" by T. Alves, J.P. Correia, J. Visser (SIG, U. Minho) in 21st International Workshop on Software Measurement and 6th International Conference on Software Process and Product Measurement (IWSM-Mensura 2011)

best paper

Data used

- Selection of “modern systems” from curated warehouse of software analysis results.

“Benchmarking Technical Quality of Software Products” by J.P. Correia, J. Visser (SIG) in 15th IEEE Working Conference on Reverse Engineering (WCRE 2008)

SIG Quality Model
Empirical validation

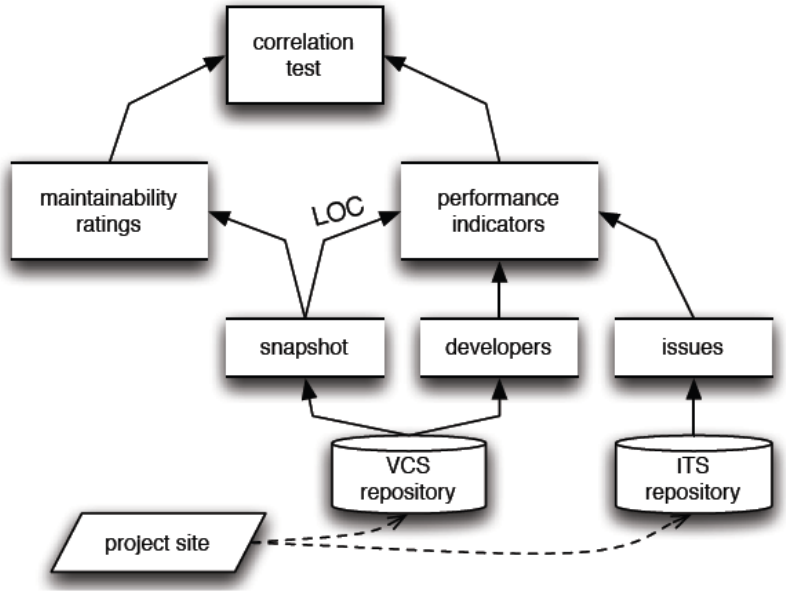
Software Improvement Group

Research

- Data: 16 open source systems (2.5 MLOC)
- Mining issues from issue trackers (50K issues)
- Analyzing source code (150 versions)

- Internal quality: maintainability of source code
- External quality: issue handling

1. Correlation analysis
2. Quantification of impact

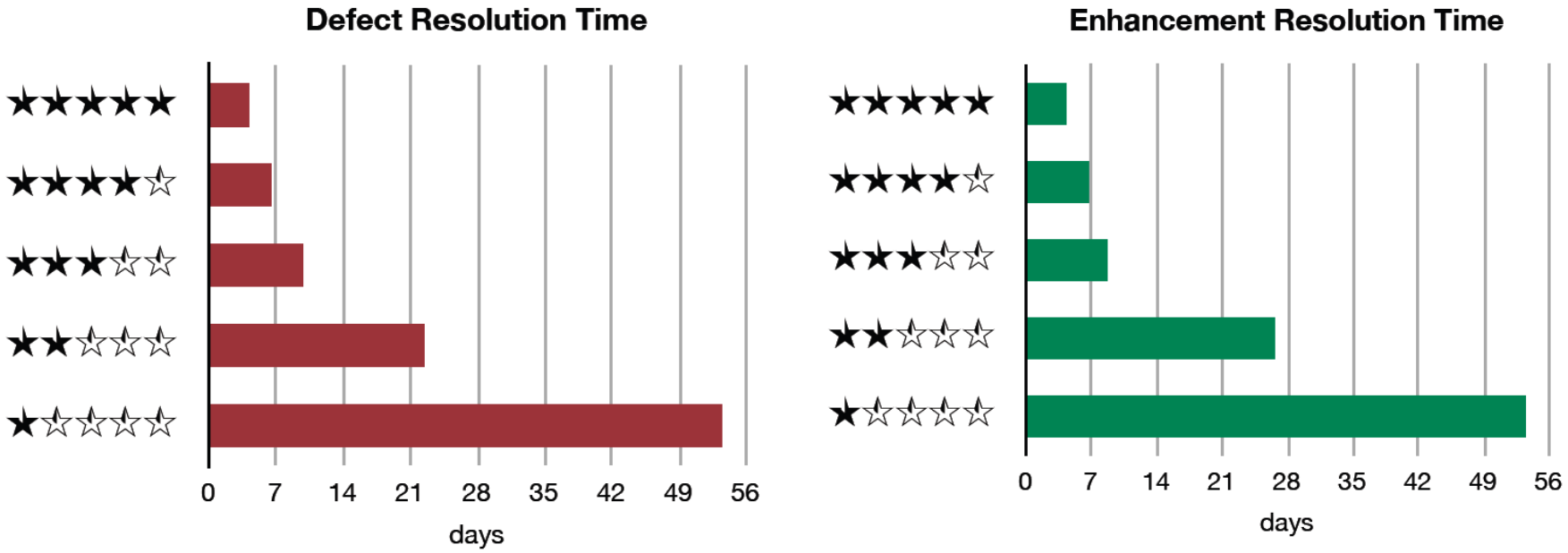


- *The Influence of Software Maintainability on Issue Handling*
  MSc thesis, Technical University Delft
- *Indicators of Issue Handling Efficiency and their Relation to Software Maintainability,*
  MSc thesis, University of Amsterdam
- *Faster Defect Resolution with Higher Technical Quality of Software, SQM 2010*

SIG Quality Model
Quantification

Software Improvement Group

Resolution time for defects and enhancements

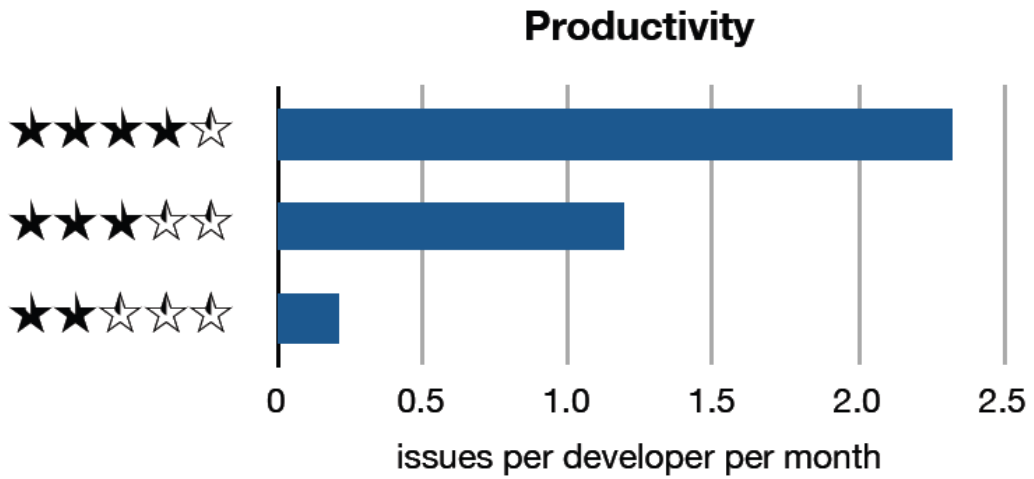


- Faster issue resolution with higher quality
- Between 2 stars and 4 stars, resolution speed increases by factors 3.5 and 4.0

SIG Quality Model
Quantification

Software Improvement Group

Productivity (resolved issues per developer per month)



- Higher productivity with higher quality
- Between 2 stars and 4 stars, productivity increases by factor 10

Software Product Certification

Certification

- Third party gives written assurance that a product conforms to specific requirements

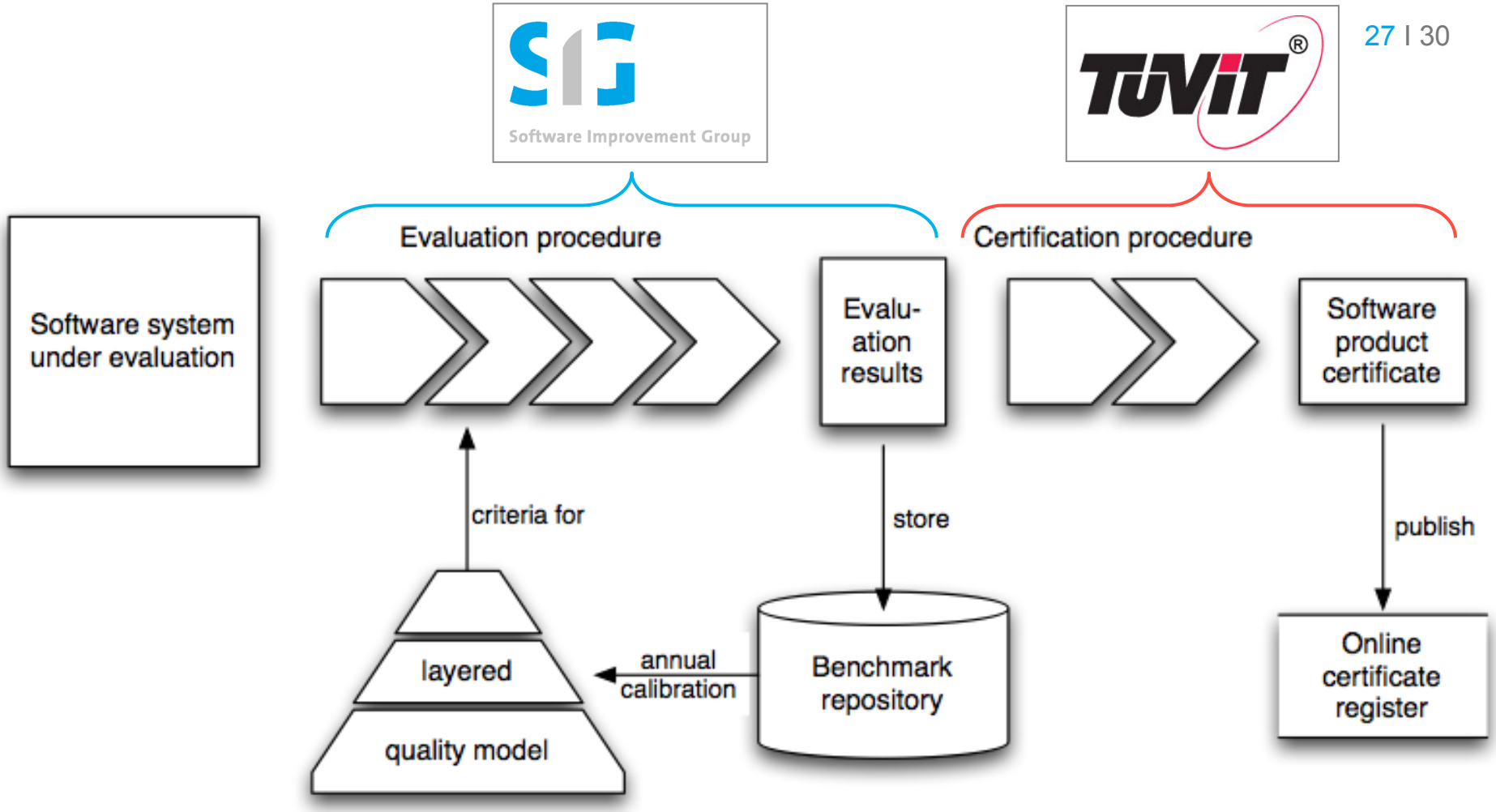
Essential elements

- Criteria: based on ISO/IEC 9126
- Evaluation body: institute that examines the product
- Certification body: institute that confirms evaluation process and result

Results

- Evaluation report, including measurements
- Certificate and quality mark:
  “TÜViT Trusted Product *Maintainability*”

Overview


Software Improvement Group
TÜVIT
Evaluation procedure
Certification procedure
Software system under evaluation
Evalu-ation results
Software product certificate
criteria for
store
publish
layered
quality model
annual calibration
Benchmark repository
Online certificate register

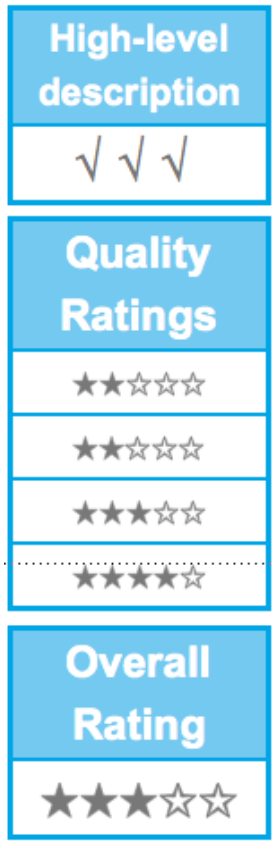
Evaluation Criteria

Calibrated against benchmark repository

Software Improvement Group

Eligibility for quality mark

- High-level description: fulfill minimal requirements
- Quality ratings: 2 stars or more
- Overall rating: 3 stars or more



Calibration w.r.t. SIG Benchmark Repository

- At level of property ratings
- Against large set of systems
- Multiple technologies, multiple domains
- E.g. about 5% of all systems reach 5 stars for the *complexity* property

★☆☆☆☆	5%
★★☆☆☆	30%
★★★☆☆	30%
★★★★☆	30%
★★★★★	5%

Software Product Certification
Who uses and how?

Software Improvement Group

Kas BANK	Tri-party collateral management	internal development
Rabobank	Bank-lobby console CRM	Ordina / Cognizant India
ProRail	On-board track visualization	Sogeti
KLM	Transfer kiosk	Accenture
SIDN	Domain registration	Profict
Agentschap BPR	Exchange of citizen information	internal development
GlobalCollect	Online payment	QuadroVision
Ordina	Insurance	internal development
MetaPress (USA)	Document management	SpringerLink
IT Mobile	Vehicle tracking, fleet management	internal development
RIPE NCC	Internet resource certification	internal development
Havenbedrijf Rotterdam	Harbour management	internal development
Rijkswaterstaat DICT	Infrastructure management	Logica

Current applications of SIG/TÜViT evaluation criteria

- Meet criteria before **acceptance** or **deployment**
- Define improvement **roadmaps** towards certifiability
- Include criteria in **RFPs**, **contracts**, and **SLAs**

What should you remember from this lecture?

Technical quality of software can be defined and measured

- ISO/IEC 9126 provides definitions
- SIG quality model performs quantification and rating

Measurement used to …

- Set technical requirements
- Monitor quality and progress
- Certify products

Thank you!

To help achieve …

- Project success
- Reduction of test effort and rework
- Fast resolution of defects and other changes
- Adaptability under changing requirements

Software Improvement Group

Dr. ir. Joost Visser
j.visser@sig.eu
http://twitter.com/jstvssr
www.sig.eu
+31 20 314 0950