

Software metrics

Jaak Tepandi

Tallinn University of Technology

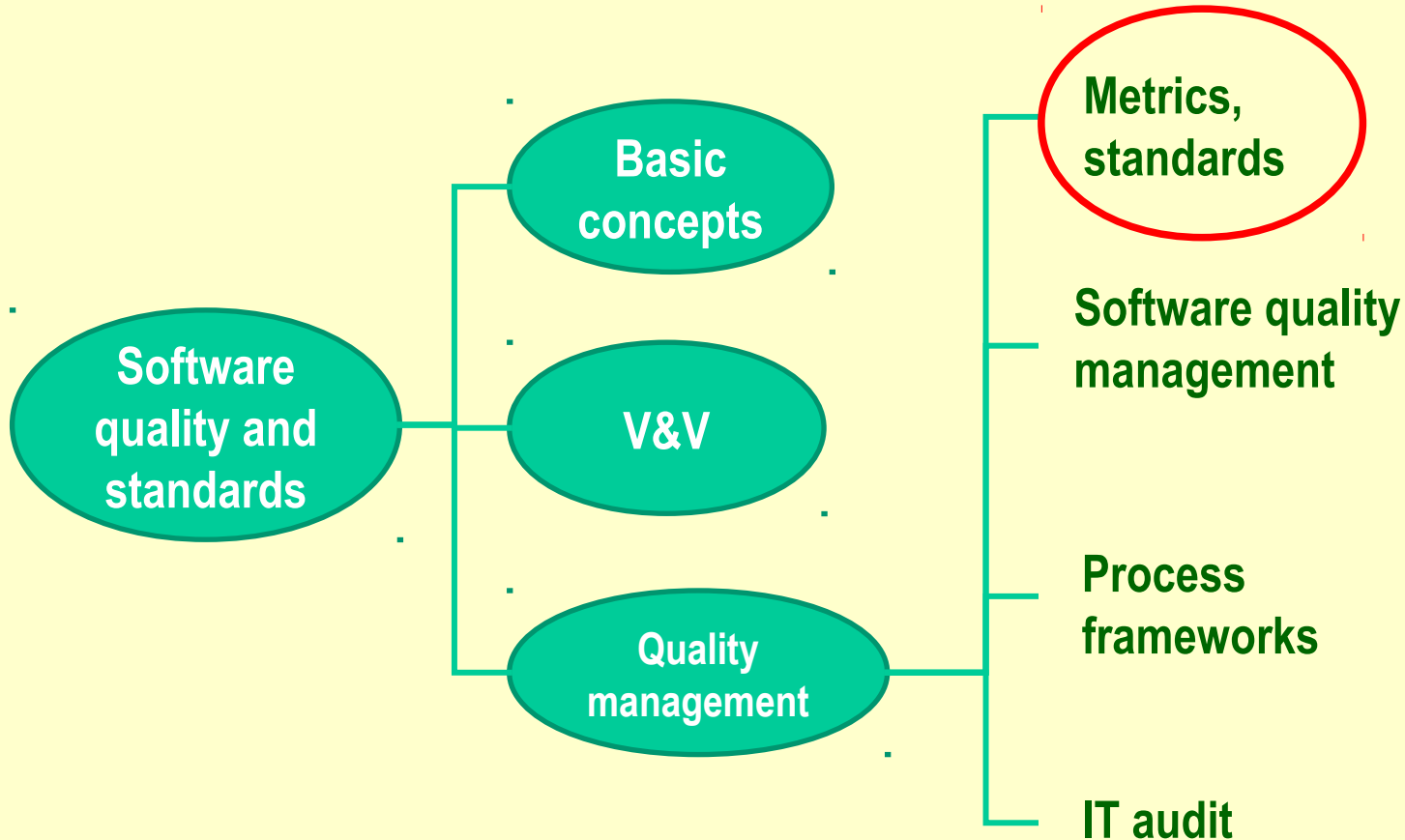
Institute of Informatics

Moodle course: „Software Quality (Tarkvara kvaliteet)”

Alternate download: tepani.ee

Version 16.11.2016

Context and content



Topics

The idea

Examples of metrics

- **SLA related**
- **Program related**
- **Cost prognosis**

Metrics for different quality characteristics and processes

Application and introduction of metrics

- **Requirements for metrics**
- **Organisations and metrics**



Idea

A quantitative measure of the degree to which an item possesses a given quality attribute

Can be measured and/or calculated

- Give examples of metrics in everyday life**
- Give examples of metrics in (agile) software development**

To support management control: “Where are we? Where should we go?”

Product, process, ... metrics

Basis for evaluation of different aspects of quality, eg in development, maintenance etc

Types / examples of metrics (1)

Process/Lifecycle oriented (eg development, maintenance)

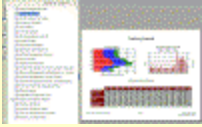
- Code coverage
- Error density
- SLA metrics, eg Mean time between failures (MTBF)

Product/Application oriented

- Function point (FP)
- Source lines of code (SLOC)

Technology oriented

- Floating-point Operations Per Second



Metrics and service level agreements

Some service targets => maintenance and/or operation metrics, eg

- **mean time between failures**
- **percentage of a system uptime**
- **time to fulfil a service request**
- **time to set up a service for a new user**
- **time to reinstate a service after a major failure**

Examples of metrics (2)

Which quality characteristics can be used for software metrics?

Source Lines of Code (SLOC) or productivity (SLOC/man-year)?

- **Quality / functionality is more important**
- **SLOC value may be computed in different ways**
- **SLOC use for evaluation may be counterproductive**
 - **eg in maintenance, negative LOC/man-year may be a good result**
- **Usage**
 - **in case of fixed SLOC counting standards**
 - **in certain situations**
 - **when comparison data is available**
 - **it provides broad characteristics**

Source Lines of Code in COCOMO

- Only Source lines that are **DELIVERED** as part of the product are included -- test drivers and other support software is excluded
- **SOURCE** lines are created by the project staff -- code created by applications generators is excluded
- One **SLOC** is one logical line of code
- **Declarations** are counted as **SLOC**
- **Comments** are not counted as **SLOC**

Examples of metrics (3)

**Modification level in specification $(D+ M+ A)/O$,
where**

- **D – detached functions,**
- **M- modified functions,**
- **A - added functions,**
- **O – original number of functions**

(Value range? The best value?)

Examples of metrics (4)

- **Adequacy between design and specification = (number of functions in design) / (Number of functions in specification).**
- **Mean time between failures = (Functioning time) / (Number of failures)**
- **(Value range? The best value?)**

Software development cost prognosis

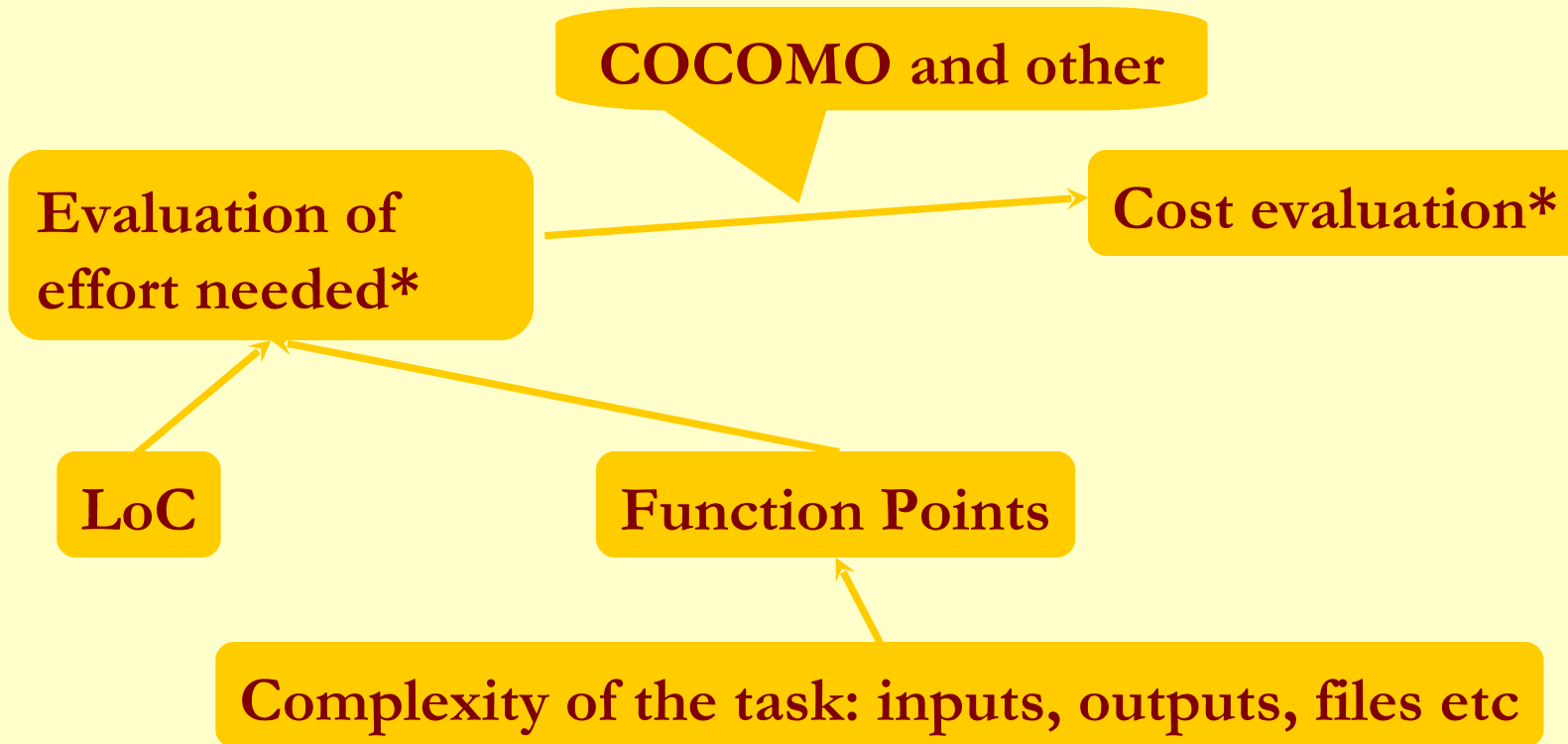
- **needed for example in evaluation of development costs in procurement and bidding**
- **ideally: input = software specification, output = development cost / time**
- **practically: complicated task – what about reliability, security etc? Development environment? Development tools? Level of developers?**
- **experience or prior database for comparison is needed**
- **planning poker, COCOMO model, function points, etc**

Effort prediction 1: Planning Poker

- Each estimator is given one deck of the cards
- A Moderator, who will not evaluate, chairs the meeting.
- The Product Manager provides a short overview + questions + discussion recorded by the Project Manager
- Each individual lays a card face down representing their estimate. Units can be days duration, ideal days, story points, etc. During discussion, numbers must not be mentioned in relation to feature size to avoid anchoring
- Everyone calls their cards simultaneously by turning them over
- People with high estimates and low estimates justify their estimate
- Repeat the estimation process until a consensus is reached. The developer who was likely to own the deliverable has a large portion of the "consensus vote"
- When the timer runs out all discussion must cease and another round of poker is played



Software development effort and cost prediction 2



- *Empirical models on basis of previous experience: project databases
- *Even calibrated models can give results that differ from the actual
- *Literature and software are available

Costar

- **choice of model**
- **creation of the tree of components**
- **evaluation of LoC, FP**
- **cost evaluation, reports, etc**
- **choice of programming languages? Databases?**
- <http://www.softstarsystems.com/>

Which quality characteristics / processes can be supported by software metrics?

Business metrics: measure benefits / risk / resources

Process metrics: measure development and others (see ISO/IEC 12207)

Software product quality metrics: measure the capability of software product to satisfy stated and implied needs when used under specified conditions

Quality in use metrics: measure the extent to which a product used by specific users meets their needs to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use

Data quality metrics: measure the degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions

(SWEBOK, ISO/IEC 25000, ISO/IEC 12207, COBIT)

Requirements for metrics

General

- Relevant
- Valid
- Reliable
- Comprehensive
- Mutually exclusive

Operative

- Easy and simple
- Does not require additional data collection
- Immune to biased interventions

Introducing metrics

Define purpose and type of metric

Define the metric, analyse if useful and satisfies the requirements for metrics

Definition of comparison indicators, target values, etc

Design of processes for metrics data reporting, collection

Different needs and metrics for different organisations

IT company

Customer company

Startup

Large organisation

Small organisation

Software metrics: summary

- **Why? Concept, idea, applications**
- **What? Content**
- **Who? Organisation behind. Who might implement? Compare: IT company, customer company, startup, large, small**
- **When? When to use it, when not? Advantages, disadvantages?**
- **Where? Relationship to other methods**
- **How? How can my organisation benefit? How to implement?**

Concepts

- **Metrics for processes**
- **Metrics for quality char-s**
- **SLOC**
- **Function points**
- **COCOMO**
- **Software cost estimation**
- **Tools examples**

Additional reading (examples)

Daniel Galin, Software Quality assurance from theory to implementation, Pearson - Addison-Wesley. Chapter 21.

Guide to the Software Engineering Body of Knowledge (SWEBOK), IEEE. Chapter 4 Section 4.

Certified Tester Foundation Level Syllabus, ISTQB. Chapter 5.2.5, 5.3.1.

Boehm, B. et al. Software Cost Estimation with COCOMO II. Prentice-Hall, 2000

David Garmus, David Herron. Function Point Analysis: Measurement Practices for Successful Software Projects. Addison-Wesley Information Technology Series, 2000