

10 metrics for improving the level of management

Pekka Forselius, Senior Advisor, FiSMA ry
Risto Nevalainen, Senior Advisor, FiSMA ry

Contents

- Introduction to selecting measures
- Classification of metrics
- Suggested TOP10 metrics
- Examples of organizational use of TOP10 metrics

The approach for selecting measures

- The aim is to select generic measures, which fit to different types of systems development and for different types of organisations
- As a pointer of generality and significance is a widely approved standard (or family of standards), where a selected measure is included directly or indirectly. Each measure is based on some reference (model).
- Focus is in process improvement, change management and business needs. Every proposed measure is validated in use at least in some FiSMA organisation to satisfy these needs.
- A measure should align with a current state of organisations. They should be beneficial right now (immediately). A level three maturity is assumed for independent use, i.e. quite managed processes and identified measurement needs.

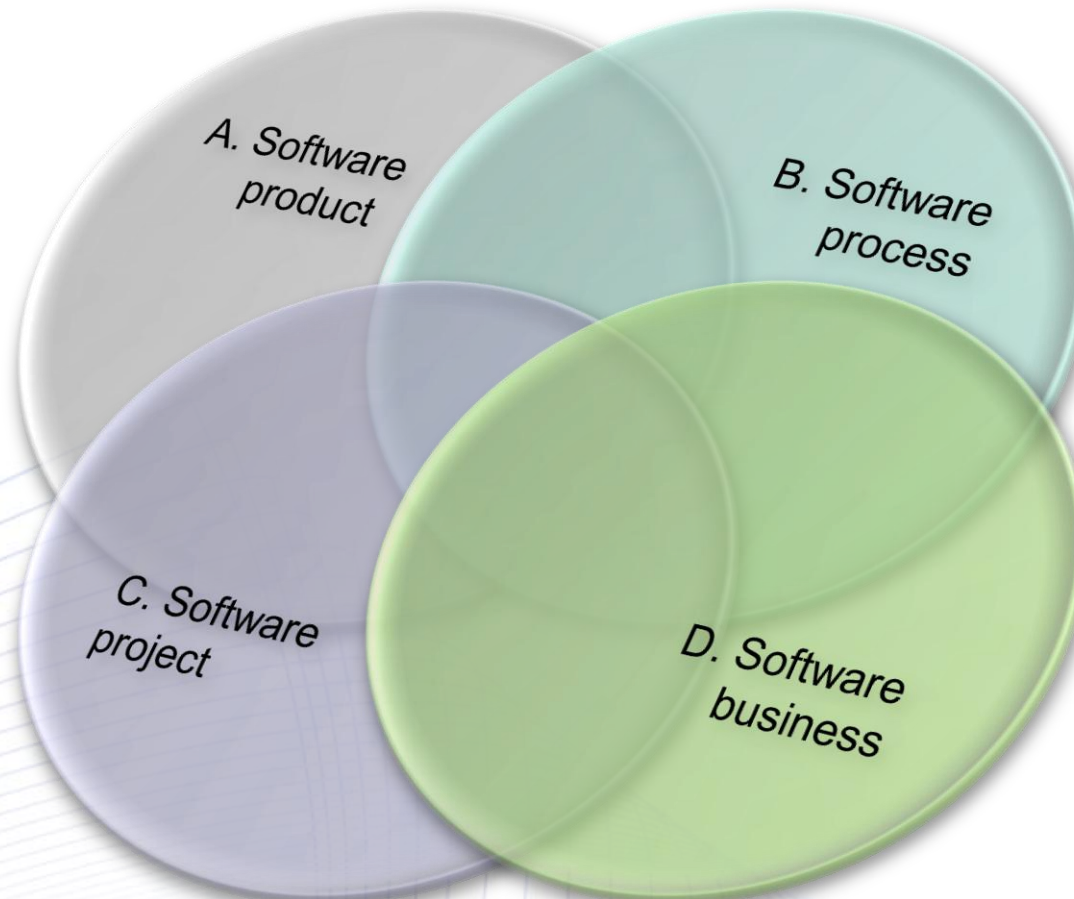
Classification of measures

- Metric = measure and it's value. A value can be a goal or achieved.
 - Example: system's current functional size is 600 function points.
- Metrics are classified in ISO/IEC 15939 as follows:
 - Base measure = measure defined in terms of an attribute and the method for quantifying it
 - Derived measure = measure that is defined as a function of two or more values of base measures
 - Indicator = measure that provides an estimate or evaluation of specified attributes derived from a model with respect to defined information needs

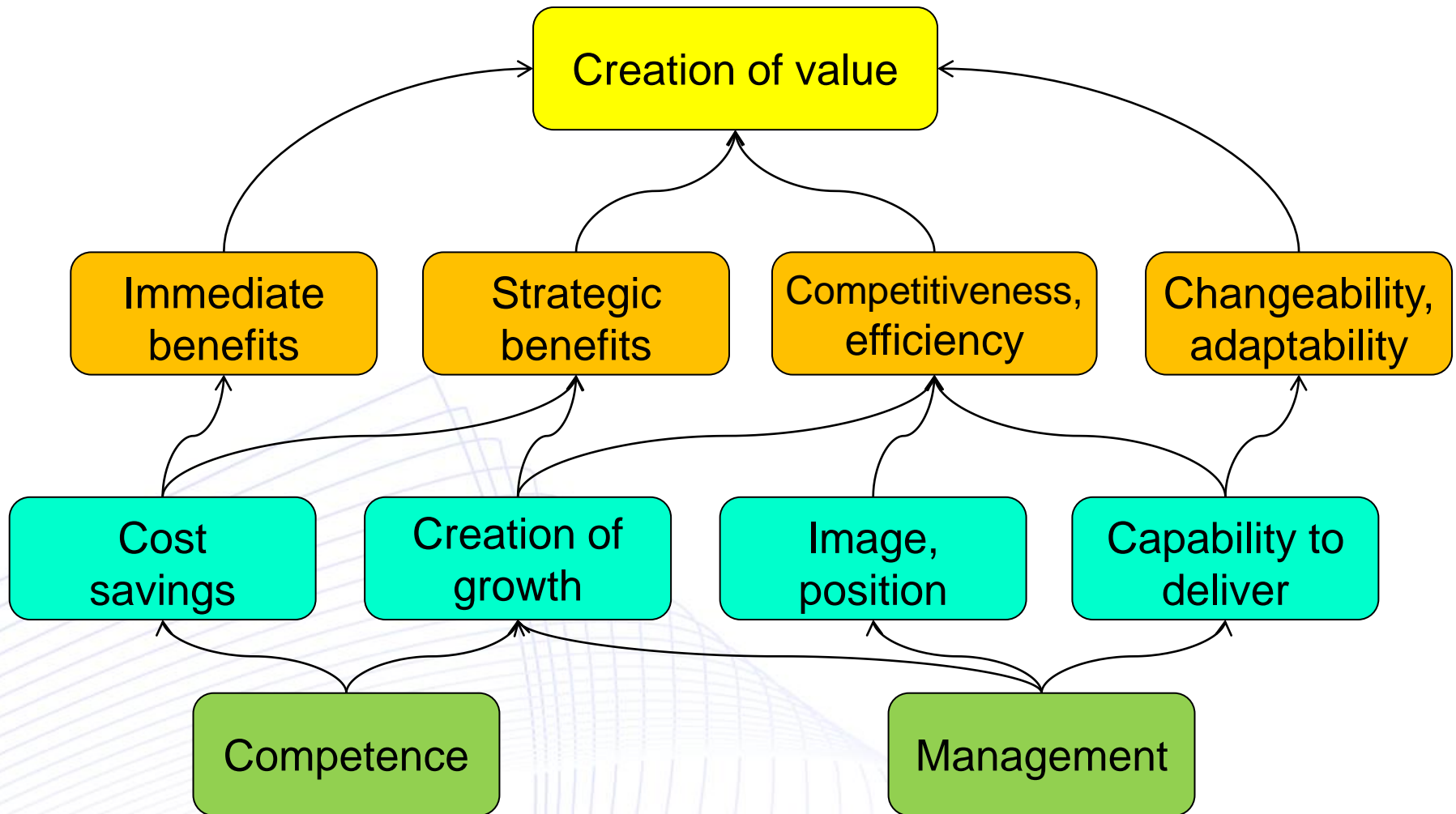
Condition based classification

- Typical areas of software metrics are:
 - Software product (itself)
 - The process used to produce software
 - The management of developing software
 - Leading and managing software business
- Other classifiers:
 - The software product standard: internal, external, in use
 - The software process standard: capability, maturity
 - Software supplier / customer organisation / end-user
 - Project model: cost, time, quality, resources, workload, benefit/profit
 - Critical systems: stability, integrity, method conformance
 - Life cycle model: specification, technical planning, development, verification, validation, production (in all e.g. V&V findings, coverage, traceability)
 - Maturity wise: initial, managed, performed sets

Classifications in this presentation



Measurement needs



Topic A: Quality measures of software product

- **TOP10 A.1 - Improvement of efficiency of end-user's work**
 - Type: Derived measure
 - Main content: A rate of user tasks, which are supported by the software compared to all other user tasks. A recommended method is a case study.
 - What the measure explains: How well and comprehensively the software fulfils user needs.

- **TOP10 A.2 - End-user satisfaction**
 - Type: Base measure
 - Main content : User experience. Could be divided to sub-measures. A recommended method is Net Promoter.
 - What the measure explains : How successfully the software serves the end-user e.g. usability and accessibility.

Topic B: Software process

- **TOP10 B.1 - Maturity of the software process**

- Type: Indicator, indirect measure
- Main content: An operational level derived from a summary of selected processes. Well-known and widely suggested methods are CMMI and SPICE.
- What the measure explains: Process wise capability of the supplier organisation to deliver products or services.

- **TOP10 B.2 - Agility of the software process**

- Type: Indicator, indirect measure
- Main content: A level of agility adaption with the whole software organisation. A recommended method is a survey or an employee inquiry.
- What the measure explains: Ability to react to external changes or requests.

- **TOP10 B.3 - Improvability of the software process**

- Type: Indicator, indirect measure
- Main content: A rate of planned and decided improvement efforts which get completed accordingly. A recommended method is audit.
- What the measure explains: Capability to execute while there is need to change and develop activities.

Topic C: Management of a software project

- **TOP10 C.1 - Functional size of the software**
 - Type: Derived measure
 - Main content: A size of the software to be developed, acquired, maintained or which is the subject to other activity. A recommended method is FiSMA 1.1 or any other ISO/IEC-standard FSM method (e.g. function points, FP).
 - What the measure explains: Functional size enables comparisons of quality, efficiency and price data of systems of different sizes. Also a value of the software's functionality for the end-user.

- **TOP10 C.2 - Workload of the software project**
 - Type: Base measure
 - Main content: The complete workload of a defined development team in assigned activities during the life cycle of the system. A recommended unit of workload is an hour.
 - What the measure explains: Important source data for schedules, pricing and comparison of productivity.

Topic D: Management of software business

- **TOP10 D.1 - Delivery speed of the software**
 - Type: Indicator, indirect measure
 - Main content: Functional size of the software delivered in the project divided by development time (FP/months).
 - What the measure explains: Delivery speed achieved in the project related to comparable ones; indicates competitiveness of both acquiring and supplying organisations.
- **TOP10 D.2 - Cost efficiency of the software purchase**
 - Type: Indicator, indirect measure
 - Main content: Total cost of the acquired software divided by a functional size, €/FP
 - What the measure explains: The cost efficiency of a project compared to similar ones; indicates competitiveness of both acquiring and supplying organisations.
- **TOP10 D.3 - Efficiency of the development portfolio**
 - Type: Derived measure, partly indicator, indirect measure
 - Main content: Revenues of a development portfolio compared to investments. A recommended method RoI or benefit/cost ratio.
 - What the measure explains: A competence to allocate and address IT efforts in accordance with business goals and value creation.

Example 1: A middle size software supplier company

- The organisation is a mid-size software company. It has reached the level of ISO9001 and aims now to develop further. Some measures are in use for management purposes and as a part of a project specific customer reporting. Customer feedbacks are collected from completed projects on a monthly basis.
- The main driver for measurement is improvement of productivity.
- Proposed measures:
 - A.2 End-user satisfaction
 - B.2 Agility of the software process
 - C.1 Functional size of the software
 - D.1 Delivery speed of the software
 - D.3 Efficiency of the development portfolio

Example 2: A large acquiring organisation

- The organisation acquires lots of project type software development services from different organisations. A frame agreement has been made with key suppliers. Software skills of own personnel are on quite solid level, especially concerning the most central systems.
- Proposed measures:
 - A.1 Improvement of efficiency of end-users' work
 - C.1 Functional size of the software
 - D.1 Delivery speed of the software
 - D.2 Cost efficiency of the software purchase
 - D.3 Efficiency of the development portfolio

Example 3: A relatively large software product company

- The company is international. Also R&D takes place in many countries. The company has stable products with wide market share especially among large corporate customers. The company has achieved several certificates according to market demands.
- Proposed measures:
 - A.2 End-user satisfaction
 - B.1 Maturity of the software process
 - B.2 Agility of the software process
 - B.3 Improvability of the software process
 - C.1 Functional size of the software
 - D.1 Delivery speed of the software
 - D.3 Efficiency of the development portfolio

Example 4: An IT project house

- The organisation is small and earns mainly by selling development projects. During offering phase there is often severe competition. Profit margin is low.

- Proposed measures:
 - B.2 Agility/ flexibility of the software process
 - B.3 Improvability of the software process
 - C.1 Functional size of the software
 - C.2 Workload of the software project
 - D.1 Delivery speed of the software
 - D.2 Cost efficiency of the software project