Software (Engineering) Project Management

Metrics-Based Project Management & Project Monitoring

Software Metrics-Based Project Management

What Are Software Metrics?

- Quantitative measures that could be used to measure different characteristics of
  - A software system
    - product metrics
  - A software development process
    - process metrics
  - A software development project
    - project metrics

A Metrics-based Project Management

- Metrics-based project management is another basic approach that modern SE uses!

Measures, Measurement, Metrics, & Indicators

- Measures, metrics, and indicators are distinct (though related) entities.

Measures

- A measure is established when a single data point is collected.
- A measure provides a quantitative indication of the extent, amount, dimension, capacity or size of some attribute or process.
- Collect measures of specific attributes of the process, project and product!

Measurement

- Measurement is the act of determining a measure.
- Measurement: Collect measures of specific attributes of the process, project and product!

Metrics

- A software metric relates individual measures in a meaningful way.
- A metric is a quantitative measure of the degree to which a system, component or process possesses a given attribute.
- Compute metrics from the measures!

Indicators

- An indicator is a metric or combination of metrics that provide insight into the software project, process, or product.
- Obtain indicators by analyzing and evaluating metrics!

Why Measure?

- To improve the software process.
- To assist in the planning, tracking and control of a software project.
- To asses the quality of the product that is produced.

Who Does?

- Software engineers collect software measures.
- Software project managers analyze and assess software metrics using .

Software Metrics

- Use of metrics as a mechanism for improving the software development process and managing software projects.

Process Metrics

- Process metrics are used to make strategic decisions about how to complete the common process framework activities.

Project Metrics

- Project metrics are used to monitor progress during software development and to control product quality.

Product Metrics

- Focus on the quality of deliverables

Software Measures

- Direct measures
  - LOC or defects over time
  - Cost and effort
- Indirect measures
  - Functionality
  - Quality
  - Reliability
  - Maintainability

Normalization of Measures for Metrics

- Measures are normalized to be used for metrics!
  - Size-oriented normalization
    - The LOC (line of code) approach
    - Size-oriented metrics
  - Function-oriented normalization
    - The function point approach
    - Function-oriented metrics

Size-Oriented Metrics

- Size-oriented metrics are derived by normalizing quality or productivity measures over the product size (typically LOC or KLOC).

Typical Size-Oriented Metrics

- Errors per KLOC (thousand lines of code)
- Defects per KLOC
- $ per LOC
- Errors per person-month
- LOC per person-month
- $ per page of documentation

Size-Oriented Metrics

- Some weaknesses of LOC as a measure (like language dependency).
- What to count in LOC (e.g. executable statements) and what not to count (e.g. comments)

Function-Oriented Metrics

- Use a measure of the functionality delivered by the application as a normalization value.

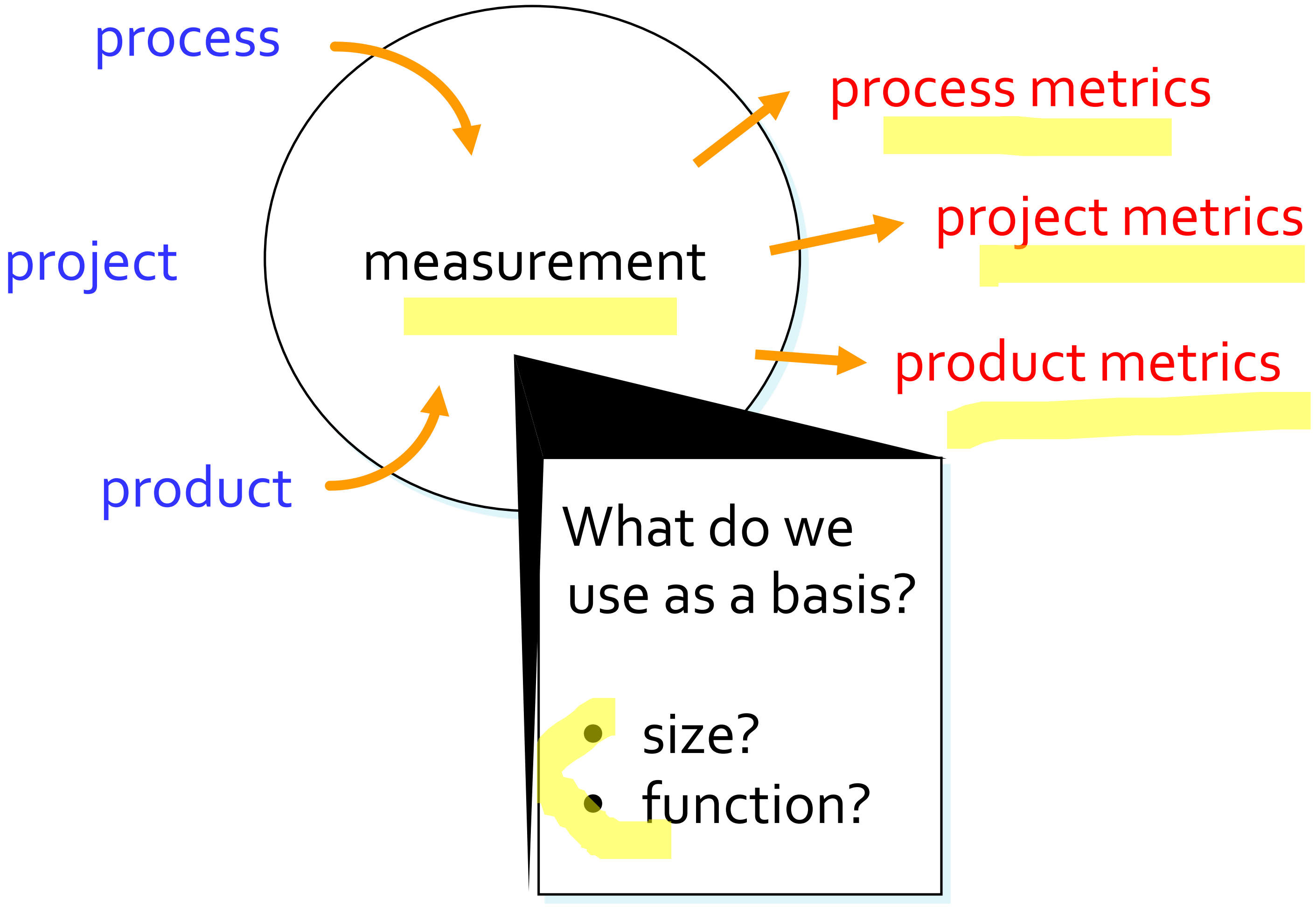
Function Points

- A method of indirectly measuring functionality using other direct measures.
- Function points are derived from measures of the information domain and a subjective assessment of problem complexity.

Typical Function-Oriented Metrics

- Errors per FP (thousand lines of code)
- Defects per FP
- $ per FP
- FP per person-month

Software Metrics



process

project

product

measurement

process metrics

project metrics

product metrics

What do we use as a basis?

- size?
- function?

Metrics in SE

- Measurement is not used in software engineering work as often as it is in other branches of engineering.
- Software engineers have trouble agreeing on what to measure and have trouble evaluating the measures that are collected.
- The only rational way to improve a process is to make strategic decisions based on metrics and indicators developed from measurements of process attributes.

Software Metrics Guidelines

- Use common sense and organizational sensitivity when interpreting metrics data.
- Don’t use metrics to appraise individuals.
- Work with practitioners and teams to set clear goals and metrics that will be used to achieve them.
- Don’t obsess on a single metric to the exclusion of other important metrics.

Project Monitoring/Tracking

Software Engineering Economics

- Software Engineering Economics are about making decisions related to software engineering in a business context.
- Earned Value Management (EVM)
- Earned Value Project/Performance Management (EVPM)

Project Monitoring Technique

- Earned Value Analysis (EVA) is a technique to monitor/track the project status by comparing (at some specific time):
  - How much effort has been expended

    versus
  - How much effort was planned to have been expended

Earned Value Analysis (EVA)

- A quantitative approach to project tracking
- Earned Value
  - A measure of progress
  - Enables us to assess the "percent of completeness" of a project using quantitative analysis!

BCWS

- The **Budgeted Cost of Work Scheduled (BCWS)** is determined for each work task represented in the schedule.
  - $BCWS_i$ is the effort planned for work task *i*.
  - BCWS is the sum of the $BCWS_i$ values for all work tasks that are to be completed by that point in time on the project schedule.
  - **Planned Value (PV)**

BCWP

- **Budgeted Cost of Work Performed (BCWP)**
  - The sum of the BCWS values for all work tasks that have actually been completed by a point in time on the project schedule.
  - **Earned Value (EV)**

BAC

- Budget At Completion, BAC.
  - $BAC = \sum (BCWS_k)$ for all tasks k

Percent Complete

- Percent complete = BCWP/BAC
  - A quantitative indication of the percent of completeness of the project at a given point in time *t.*

SPI & SV

- **Schedule Performance Index** (SPI) = BCWP/BCWS
  - SPI is an indication of the efficiency with which the project is utilizing scheduled resources.
  - SPI=1.0 (Efficient execution of project schedule)
- **Schedule Variance** (SV) = BCWP-BCWS

ACWP, CPI & Cost Variance

- **Actual Cost of Work Performed, ACWP**
  - The sum of the effort actually expended on work tasks that have been completed by a point in time on the project schedule.
- **Cost Performance Index** (CPI) = BCWP/ACWP
  - CPI=1.0 (Within the budget)
- **Cost Variance** (CV) = BCWP - ACWP

Example: Earned Value Analysis

Work Tasks	Estimated Effort in Pers-days	Actual Effort spent so far in Pers-days	Estimated Completion date in mm/dd/yy*	Actual Completion date in mm/dd/yy*
1	10	10	2/5/11	2/5/11
2	15	25	3/15/11	3/25/11
3	30	15	4/25/11	
4	25	20	5/5/11	4/1/11
5	15	5	5/25/11	
6	20	15	6/10/11	

The Status Checking Date : 4/5/2011

BAC?
BCWS?
BCWP?
ACWP?
PercentComplete?
CV?
SV?

Example

- BAC is the sum of the estimated efforts for all the tasks:

  BAC = 10+15+30+25+15+20 = 115 person-days

- BCWS for the date 4/5 is the sum of the estimated effort of all the tasks which were schedule to be completed on or before 4/5:

  BCWS = 10 + 15 = 25 person days

Example

- BCWP for the date 4/5 is the sum of the estimated effort of all the tasks which were actually completed on or before 4/5:

  BCWP = 10 + 15 + 25 = 50 person-days

- ACWP for the date 4/5 is the sum of the actual efforts expended for all the tasks that have been completed on or before 4/5:
-

  ACWP = 10 + 25 + 20 = 55 person-days

Example

- Percent Complete**=**
  BCWP / BAC = 50/115 = .434
  - The project is estimated to be 43% complete as of 4/5

- Cost Variance **=**
  BCWP – ACWP = 50 – 55 = - 5

- Schedule Variance **=**
  BCWP - BCWS = 50 – 25 = 25