Contents

1 Why Measure Software?

- To determine the quality of the current product or process and to make informed comparisons
- To predict qualities of a product/process
- To improve quality of a product/process (often continuously *kaizen*)

2 Metrics

2.1 Example metrics

- Error rates / Defect rates
- A defect is a failure to do something
- An error is a wrong result or illegal action
- Code generation rates. Measured by: per **person**, per **team**, per **project**. Will involve **aggregation** see later
- Errors should be categorized by origin, type etc.

2.2 Main objectives of software quality metrics

- Facilitate management control, planning and managerial intervention. Based on deviations of actual
  - performance from planned.
  - timetable and budget performance from planned.
- 2. Identify situations for development or maintenance process improvement (preventive or corrective actions). Based on:
  - Accumulation of metrics information regarding the performance of teams, units, etc.

2.3 Software quality metrics Requirements

General requirements:

- Relevant
- Valid
- Reliable
- Comprehensive
- Mutually exclusive

Operative requirements:

- Easy and simple
- Does not require independent data collection
- Immune to biased interventions by interested parties

2.4 Metrics evaluations

Metrics can be evaluated on:

- **Products**: Explicit results of software development activities. Deliverables, documentation, other artifacts produced
- **Process**: Activities related to production of software
- **Resource**: Inputs into the software development activities, such as hardware, knowledge, people.

2.5 Process metrics

Rates of production / productivity, quality of estimates, burn down rate (project velocity), refactoring rate. These can lead to **long term process improvement** by driving reflection on the process and suggestions for improvement

2.6 Product metrics

2.7 Types of measures

- Direct: Cost, effort, LOC, speed, memory
- Indirect: functionality, quality, efficiency, reliablity, maintainability

2.8 Size oriented metrics

- Size of software produced
- LOC - Lines of Code or KLOC - 1000 Lines of code
- SLOC statement lines of code (no whitespace)
- Number of function points (NFP)
- Open used as part of density CONT

2.9 Process metrics categories

- Software oricess quality metrics: Error density, severity metrics
- Sof

Reference section

measure
: A quantitative indication of extent, amount, dimension, capacity, or size of some attribute of a product or process. for example: Lines of code (LOC), Number of errors

metric
: quantitative measure of degree to which a system, component or process possesses a given attribute. That is to say a derived measure. May involve guesstimates where human processes are concerned. for example: Error density.

kaizen
: An approach to **continuous** organization improvement with emphasis on continuous measure and understanding of process.