Hierarchical Software Size Metrics

Petri Vesterinen
Nokia Research Center
P.O.Box 422, FIN-00045 Nokia Group, Finland
petri.vesterinen@research.nokia.com

Nokia Corporation

- net sales 52.6 billion FIM (nearly 10 billion USD)
- total personnel 36,600
- sales in 130 countries
- main products:
  - mobile phones, communicators
  - digital exchanges and telecommunications networks
  - monitors, satellite and cable receivers, multimedia equipment
Introduction

- Software size metric is the most important software metric used in several planning and tracking activities.
- Used in derived metrics like productivity and quality.
- Software size can be measured in several ways:
  - 'physical' size
    - Lines of Code (LOC, NLOC, ...)
    - Number of modules
    - Amount of memory used, etc.
  - 'functional' size
    - FPA and variants
  - 'logical' size
    - Complexity metrics

Some Problems

- LOC is not suitable for:
  - estimation where LOC is not known accurately enough
  - early estimation
  - new technology, new functionalities
  - measuring productivity or quality
  - different languages & different coding styles
- FP
  - is not suitable for very early estimation where function types can not be identified
  - counting may be too laborious in early phases
  - may not be accurate enough in late project phases
FP & LOC: Primary Functional Areas

- This explains why:
  - FP is often used in estimation (availability)
  - LOC based effort estimates are often more accurate

Available Solutions (1/2)

- Use of only one SW size metric despite of problems, e.g.
  - only LOC because of huge amount of history data
  - automatic code counting tools available for LOCs

- PROBLEMS:
  - selected metric is good only for some purposes
Available Solutions (2/2)

- Different metrics for different purposes, e.g.
  - Object Points, FP and LOC in COCOMO 2.0
  - FP: agree project size and early estimation; LOC: quality measurements and predictor of memory consumption

- PROBLEMS:
  - Several metrics has to be taken -> increased measuring cost
  - Depict different views and not fully comparable
  - Hard to find reliable conversion factors (Jones’s backfiring error may be >30%)

What We Need (1/2)

We need a HIERARCHICAL software size metric that:

- Uses same metric units in all project phases
- Avoids conversion between very different units
- Is cost-effective to measure in all phases
- Is implementation independent in early phases (like FP)
- Is implementation dependent in late phases (like LOC)
- Takes use of the increased understanding of the nature of the work as the specification and design proceeds
- Can be used in derived metrics like productivity and quality
- Is easy to understand
- Is based on solid theory
What We Need (2/2)

Amount of information

Information available

Needed size metric

LOC

FP

estimated

TIME

Nokia Solution (1/3)

NSM: "Nokia Sizing Method"
Every level uses same metric units
(called here as NSP, "Nokia Sizing Point")
Available Solutions for FP

- Early Function Points is hierarchical in the context as it uses the same FP units as IFPUG FPA.
- 3D-FP and FFP are not hierarchical as they use their own units.

Summary

- Current size metrics are suitable only for restricted use.
- It is hard to cope with different size units.
- The metric should 'grow up' or 'mature' along with SW projects.
- There is really a need for a hierarchical software size metric.
- The Nokia solution
  - Preliminary results suggest that the idea really works.
  - Calibration is the critical activity in forming the metric.
- A research interest in the future to form a true hierarchical size measurement framework in the FP context.
- Another interesting context would be OO.