

Software Metrics and Quality Engineering

CSE 8314 — Fall 2019

Prof. Jeff Tian, tian@lyle.smu.edu
CSE, SMU, Dallas, TX 75275
(214) 768-2861; Fax: (214) 768-3085
www.lyle.smu.edu/~tian/class/8314.19f

Module III: Internal Metrics

- Metrics: Internal vs. External
- Internal Size Metrics
- Internal Complexity Metrics
- Other Internal Metrics

Software Measurement

Basic assumption: The lower the complexity or other metrics values, the more desirable:

- ▷ cheaper to build;
- ▷ easier to maintain;
- ▷ more reliable;
- ▷ ...

Desirable? can it be quantified?

- using (target) external metrics as output/response/dependent variables

Internal (and some external) metrics as input/predictor/stimulus/independent variables

Internal Measures: Why?

Difficulties with external metrics

Need: predictors, leading indicators, controllers, etc.

Internal metrics as answers

- ▷ must be available earlier
- ▷ controllability and observability
- ▷ finer granularity

Granularity: "whole, parts or details"

Internal Measures: What?

Complexity:

general terms, often to indicate all internal metrics

- ▷ complexity dimensions
 - control (algorithm, decisions)
 - data
 - presentation (organization)
- ▷ many metrics, long history

Other metrics

- ▷ size, often as separate metrics
- ▷ information contents
- ▷ volume
- ▷ algorithmic
- ▷ non-code-based metrics

Theory: Complexity Dimensions

Presentation: Physical presentation for readers that has no effect on functionality.

Control: Instructions, control structures, and control dependencies.

Data: Data items, data structures, and data dependencies.

Comments:

- ▷ Control + Data = Abstract;
- ▷ Orthogonal dimensions.

Theory: Measurement Levels

Lexical: Token based measure computation;

Syntactic: Directly syntax based measure computation;

Semantic: Semantic analysis needed for measure computation.

Comments:

- ▷ 27 possible points in a 3-D space;
- ▷ Space proximity \approx Measure similarity;
- ▷ Similarity to ABCS in Usability

Measurement Level

- Measurement level: depending on the computational model for the metrics, lexical, syntactic, semantic

- Lexical
 - ▷ token based metrics
 - ▷ e.g. various counts: LOC, variables, etc.

- Syntactic
 - ▷ language syntax used in computation
 - ▷ e.g., statement count

- Semantic
 - ▷ language semantics used in computation
 - ▷ e.g., dependency (DU pair), naming, etc.

Size Metrics

- Typically lexical or syntactic levels

- Lexical
 - ▷ token based metrics
 - ▷ e.g. various counts: LOC, variables, etc.

- Syntactic
 - ▷ language syntax used in computation
 - ▷ e.g., statement count

- Semantic
 - ▷ language semantics used in computation
 - ▷ e.g., dependency (DU pair), naming, etc.

Size Metrics

- Size and functionality in 5CM
 - also, cone of uncertainty Fig.6-2 (p.85)

- Language factor:
 - ▷ Table 4.1 1991 Gearing Factors in SME (p.37)
 - ▷ Table 4.2 2005 Gearing Factors in SME (p.38)
 - ▷ IBM work: q-code equivalence

- Change, reuse, refactoring, etc.
 - ▷ CSI vs SSI: rational
 - ▷ reuse: verbatim, ($< 25\%$)modified, ($\geq 25\%$)modified, new
 - ▷ use of macros, others

- consistent counting rules needed

Size Metrics: FP

- number of external user inputs, inquiries, outputs, and master files to be delivered by the development project

- Comparison to SLOC variations
 - ▷ availability in time
 - ▷ black-box/functional view/focus

- counting rules and standards
 - ▷ weights and weighted sums are used
 - ▷ example SME Table 4.3 (p.42)

Control Complexity Metrics

- Lexical
 - ▷ control types and related count
 - ▷ e.g. #GOTOs, #decisions, #branches etc.

- Syntactic
 - ▷ concatenation, nesting, GOTO
 - ▷ e.g., nesting level

- Semantic
 - ▷ control dependency
 - ▷ e.g., logical/algebraic sensitization

Control Complexity Metrics

- A few famous metrics
- McCabe's cyclomatic complexity
 - ▷ $v = e - n + 2p$
 - e edges, n nodes, p connected components
 - ▷ $v = c + 1$
 - c predicates for single connected graph
 - ▷ simple rule: loop similar to branching
 - ▷ poor correlation with effort
- Knot count
 - ▷ total # of unavoidable knots when branches drawn on one side of the sequential flow
 - ▷ un-structuredness of a program
 - $kc=0$ for structured programs

Data Complexity Metrics

- Lexical
 - ▷ token based metrics
 - ▷ e.g. variables (unique vs freq.), etc.

- Syntactic
 - ▷ scoping (and other) rules used
 - ▷ e.g., visible variable set metrics

- Semantic
 - ▷ data dependency
 - ▷ e.g., DU pair, etc.

Interface Complexity Metrics

- both control and data aspect
- Mostly lexical: counting by name, type, freq., etc.
 - ▷ function + procedure calls
 - ▷ number of input/output statements
 - ▷ number of input/output parameters
 - ▷ often used in FP and other metrics
- information flow metrics:
 - ▷ $\text{ifm} = (\text{fan-in} \times \text{fan-out})^2$
 - ▷ flow into/out of a procedure
 - ▷ also include global data structure accessed/updated

Interface Metrics vs. Usability

- Interface metrics often related to usability and UI/UX (user interface/user experience)

- Map lexical/syntactic/semantic to ABCS and related metrics
 - ▷ Anthropomorphic: KLM (key-stroke level model)
 - ▷ Behavioral: perception etc.
 - ▷ Cognitive: task-based metrics
 - ▷ Social: communication/coordination models

Volume/Complexity Metrics

- both control and data aspect, often based on information theory
- Halstead Software Science metric: most famous, 1st systematic treatment

m	name & formula (or definition)
η_1	number of distinct operators
η_2	number of distinct operands
N_1	total number of operators
N_2	total number of operands
V	program volume $V = (N_1 + N_2) \log_2(\eta_1 + \eta_2)$
V^*	potential volume or min possible volume
E	programming effort $E = V^2/V^*$ (approximation) $E = \frac{\eta_1 N_2 (N_1 + N_2) \log_2(\eta_1 + \eta_2)}{2\eta_2}$

- HAC metrics by Bail-Zelkowitz

OO Complexity Metrics

- CK metrics by Chidamber and Kemerer
 - ▷ WMC: weighted methods per class
 - ▷ DIT: depth of inheritance tree
 - ▷ NOC: number of children
 - ▷ CBO: coupling between object classes
 - ▷ RFC: response for class
 - ▷ LCOM lack of cohesion on method

- Nominal values/ranges for CK metrics

Presentation Complexity Metrics

- Lexical
 - ▷ presentation token metrics
 - ▷ e.g. comments, blanks, etc.

- Syntactic
 - ▷ rules used in presentation
 - ▷ e.g., indentation rules, etc.

- Semantic
 - ▷ meaning in presentation
 - ▷ e.g., naming, aliasing

- Also link to usability metrics and ABCS

Hybrid Complexity Metrics

- Some example earlier: interface complexity (FP, information flow), and Halstead's software science metrics
- Other ad hoc metrics
 - ▷ based on logical analysis
 - ▷ based on statistical analysis
 - primarily PCA
- other possibilities
 - ▷ multiple metrics as profiler
 - ▷ multiple models
 - ▷ combined models vs metrics vs results

Other Complexity Metrics

- algorithmic
 - ▷ work by Chaitin
 - ▷ minimal algorithmic representation
 - ▷ computational complexity linkage

- non-code-based metrics
 - ▷ measuring specs, design, etc.
 - ▷ FP-like ideas
 - ▷ Card design complexity metrics
 - high/module-level structural/data complexity