Software Product Metrics

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Product Metrics

- Product metrics quantify internal software attributes
- Examples of internal attributes
  - Size
  - Coupling
  - Cohesion, etc.

Classification of Metrics

- Dynamic Metrics
  - They are collected by measuring programs during runtime
- Static Metrics
  - They are collected by measuring source code or documents

Dynamic vs. Static Metrics

- Dynamic metrics aim at assessing a system after it has been implemented
  - They quantify attributes like efficiency and reliability
- Static metrics can be collected in earlier phases of development
  - They quantify attributes like complexity and maintenability

Static Product Metrics

Examples of Static Metrics

- Fan-in / Fan-out
- Length of Code
- Cyclomatic Complexity
- Vocabulary Size
- Depth of Conditional Nesting

Fan-in and Fan-out

- Fan-in
  - It counts the number of functions calling a specific function
  - High values may mean high change impact (e.g., ripple effects)
- Fan-out
  - It counts the number of functions called by a function
  - High values may mean high complexity

Length and Complexity

- Length of Code
  - This kind of metrics has shown to be the most useful and reliable in many cases
  - In general, the larger the code, the more complex and error-prone it is likely to be
- Cyclomatic Complexity
  - It measures the control flow of a program (*if*, *while*, *for*, etc.)
  - This metric is related to understandability

Vocabulary and Nesting

- Vocabulary Size
  - It counts the number of identifiers (e.g., class names in OOP)
  - The more identifiers the more meaningful they are likely to be
- Depth of Conditional Nesting
  - It counts the number of nested statements (e.g., *if, while, etc.)*
  - Deeply nested statements are harder to be understood and error-prone

Bibliography

- Ian Sommerville. **Software Engineering**, 9th Edtion. Pearson Education, 2010.
  - Section 24.4 Software Measurement and Metrics