4.3.3.3 Analysability

Return to Maintainability

About

Analysability is defined by the IEEE glossary of Software Engineering as “the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to changes to the environment”.

One way to understand the Analysability of a system or a program is to understand the size of the system or program. As a rule of thumb, as the size of system or program increases, it becomes increasingly harder to successfully modify the software to correct faults, improve performance, or to adapt to changes in the operating environment. This Analysability can be performed any time while using either the Waterfall Model or the Agile Model during the early stages of analysis and design. As projects mature and use a larger code base, the models can be based directly on either the source code or Unified Modeling Language (UML) models created using reverse engineering. The metrics can also be used during both Greenfield or Brownfield deployments. Another way to examine Analysability could be to collect and use similar metrics on Distributed Computing systems where, instead of using classes (i.e., as in Object-Oriented Programming (OOP)), the number of Nodes, Endpoints and message types could be used. To address **Structural Complexity**, the connections between processes can be used.

Table 1: Metrics for Class complexity[1)]

Type of Metrics	Metric definition
Size Metrics	
Number of Classes (NC)	The total number of classes
Number of Attributes (NA)	The total number of attributes
Number of Methods (NM)	The total number of methods
Structural complexity Metrics	
Number of Associations (NAssoc)	The total number of associations
Number of Aggregations (NAgg)	The total number of aggregation relationships within a class diagram (each whole-part pair in an aggregation relationship)
Number of Dependencies (NDep)	The total number of dependency relationships
Number of Generalisations (NGen)	The total number of generalization relationships within a class diagram (each parent-child pair in a generalization relationship)
Number of Generalization hierarchies (NGenH)	The total number of generalization hierarchies in a class diagram
Maximum DIT (MaxDIT)	It is the maximum DIT value obtained for each class of the class diagram. The DIT value for a class within a generalization hierarchy is the longest path from the class to the root of the hierarchy

Type of Metrics	Metric definition
Size Metrics	
Maximum HAgg (MaxHAgg)	It is the maximum HAgg value obtained for each class of the class diagram. The HAgg value for a class within an aggregation hierarchy is the longest path from the class to the leaves

DIDO Specifics

Return to Top

☐ *TBD - to be added/expanded in future revisions of the DIDO RA*

1)

Marcela Genero, Mario Piattini and Ronda de Calatrava, Empirical validation of measures for class diagram structural complexity through controlled experiments, Accessed 4 August 2020, https://pdfs.semanticscholar.org/dd52/5d80c1f370258e56cd956bcb903706c216dc.pdf

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:2_nonfunc:20_maintainability:analysability&rev=1623264471

Last update: **2021/06/09 14:47**