

Accounting for Inheritance in Object-Oriented Software Quality Measurement

Jehad Al Dallal*

Department of Information Science, Kuwait University, Kuwait

Typically, internal software quality attributes, such as size, cohesion, and coupling, are not of interest for software practitioners unless they are useful in indicating external quality attributes or other related activities of interest. Researchers have identified several external quality attributes, such as adaptability, reusability, understandability, maintainability, testability, and completeness, as somehow related to internal quality attributes. Adaptability refers to the extent to which software or part of it adapts to change in environments other than those for which it was specifically designed. Reusability refers to the ease with which a component can be used in building other components. Understandability, maintainability, and testability refer to the ease with which a given piece of software can be understood, maintained, and tested, respectively. Completeness refers to the extent to which a piece of software is complete, in terms of capabilities. When considering object-oriented software, one of the main research questions is whether a software engineer must consider the inherited methods and attributes in size, cohesion, and coupling measurement to indicate the adaptability, reusability, understandability, maintainability, testability, and completeness of a subclass.

A subclass cannot be moved alone to a new environment. Instead, all of the direct and indirect superclasses must be also moved with the subclass to the new environment. Therefore, when studying the adaptability of a subclass, the code of the direct and indirect superclasses must be considered as well. Similarly, the reusability of the direct and indirect superclasses must be considered when reusing a subclass. A subclass cannot be understood and maintained unless the direct and indirect superclasses are inspected and analyzed. A subclass is not complete by itself without the inherited code. Therefore, studying the completeness of a subclass must include exploring the completeness of the code of the subclass and its superclasses altogether. As a result, none of these applications requires solely analyzing or exploring the code within the subclass; they also require the analysis and exploration of the direct and indirect superclasses. Therefore, it is expected that if the internal quality metrics are applied on the subclasses without considering the inherited attributes and methods, then the obtained quality values will provide incorrect indications for the external quality attributes. However, empirical studies are required to approve or disapprove this expectation.

*Corresponding author: Jehad Al Dallal, Department of Information Science, Kuwait University, Kuwait, E-mail: jaldallal@yahoo.com

Received September 11, 2013; Accepted September 12, 2013; Published September 18, 2013

Citation: Dallal JA (2013) Accounting for Inheritance in Object-Oriented Software Quality Measurement. J Inform Tech Softw Eng 3: e118. doi:10.4172/2165-7866.1000e118

Copyright: © 2013 Dallal JA. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.