A practical model for Measuring Maintainability

Original paper by

Ilja Heitlager
Software Improvement Group
The Netherlands
Email: i.heitlager@sig.nl

Tobias Kuipers
Software Improvement Group
The Netherlands
Email: t.kuipers@sig.nl

Joost Visser
Software Improvement Group
The Netherlands
Email: j.visser@sig.nl

Summary written by:

SOHAN ARGULWAR
Concordia University, Montreal
Student ID: 6757871

For any project, about 20% is effort is on writing the code for requirement and remaining 80% work is on maintainability. So, we can say that huge amount of time is spent on maintaining the product. Even after successful delivery of any project maintainability comes into picture in terms of the technical quality of source code of that system. The ISO 9126 does not provide maintainability related measures from source code of system. The measurement Index (MI) which is quantitative measure helps to achieve this task. This paper talks about a maintainability model and requirements for Maintainability Index (MI). This MI is not dependent on source code and documentation but on technical staff activity. The fitting function is thus dependent on the results gathered from technical activities. But, this MI is not applicable for all the system and it has some drawbacks as it has some requirements. MI is based on several components such as Halstead volume metric (HV), Cyclomatic Complexity metric (CC), average number of Lines of Code (LCOM), and optionally on percentage of comments in module (COM). The software development team uses diverse type of technologies, different thinking and for diverse purpose. So, the MI proposed has some limitations in assessing software quality. The minimum requirements to get MI is like measures should be technology independent, each measure should have straight forward definition and easy to calculate, each measure should be easy to understand, and the measure should enable root cause analysis. The paper describes an example of such maintainability model SIG Maintainability Model based on above requirements. The paper describes several source code properties on maintainability characteristics of a software system. Finally the paper concludes that the maintainability model presented is good reference for software quality but not efficient measure as it has some limitations. MI suffers from root-cause analysis, ease of computation, language independent, understandability, explain ability, and control.