Maintainability: Factors and Criteria

Software Metrics Study: Technical Memorandum 1

Marc Frappier
Stan Matwin
Ali Mili

University of Ottawa

March 3, 1994

0. - Table of Contents

1.- Introduction

The Software Metrics Study was initiated to propose metrics of the maintainability of the software produced by the MDSF project. This memorandum is the first deliverable. Its objectives are to define the concept of maintainability, to describe the factors influencing it and to define criteria by which maintainability can be quantitatively evaluated.

2. - Maintainability

Maintenance is the activity of modifying a software product after initial delivery. Maintainability is the ease with which a software product can be modified. Maintainability is a requirement of the CEI and SWCI specifications. Its importance stems from the fact that MDSF will have to evolve and adapt to a changing environment over the next 30 years. Using sound software engineering principles, the cost of maintenance can be minimized. Following [LS80], we divide maintenance in three categories:

- *corrective maintenance*: the correction of faults when the system does not behave according to its specification;

- *adaptive maintenance*: the adaptation of the system to changes in the operational environment while keeping the same functionality;

- *perfective maintenance*: the extension of a system's functionality and improvement in the services provided.

3. - Factors

Maintainability is a component of a more general concept, software quality, which is described in terms of a hierachy (see figure 1 [EM87]) of factors, criteria and metrics. A factor is a top-level expression of software status for management reporting. Each factor is described by a set of criteria. Each criterion is measured by a set of metrics. A criterion may describe more than one factor and some criteria may be measured by the same metric.

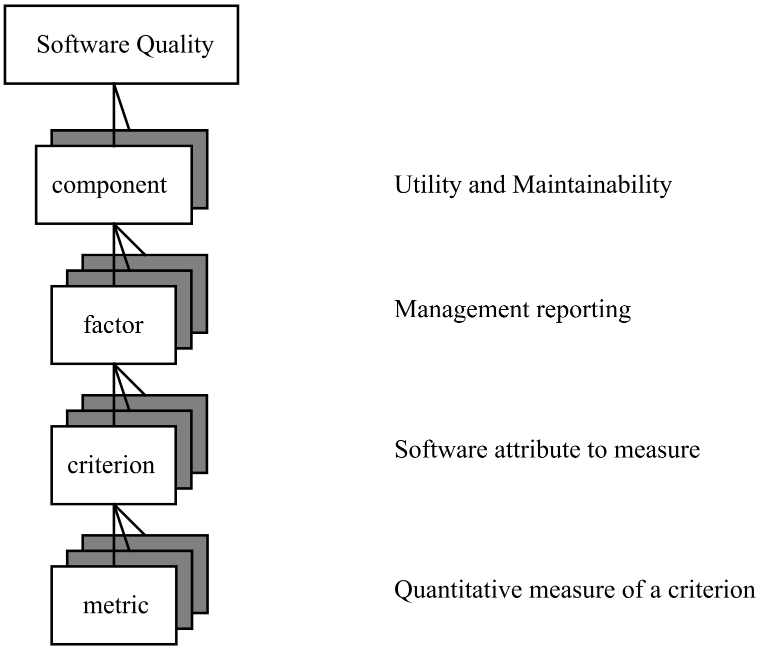
Software Quality
component
factor
criterion
metric
Utility and Maintainability
Management reporting
Software attribute to measure
Quantitative measure of a criterion

Figure 1 - Software quality hierarchy

Several models of software quality have been proposed. We selected the one suggested by Boehm et al. in [BBKL78] for its simplicity, and modified it to be consistent with figure 1. It consists of seven factors (Boehm et al. call them characteristics), and the authors' definition of maintainability is closer to the one just given. The seven factors are:

- reliability: the extent to which a software product will perform without any failure;
- efficiency: the extent to which a software product performs its intended functions without wasting machine resources;
- usability: the extent to which a software product is convenient, practical and easy to use;
- testability: the extent to which a software product facilitates the establishment of acceptance criteria and supports evalutation of its performance;
- understandability: the extent to which the purpose of a software product is clear to a human observer;
- modifiability: the extent to which a software product facilitates the incorporation of changes;
- portability: the extent to which a software product can be easily and effectively operated in a variety of computing environments.

Some factors are potentially conflicting. For example, increased efficiency may be achieved at the cost of a lower modifiability and understandability. According to Boehm et al., the factors influencing maintainability are:

- testability
- understandability
- modifiability

McClure and Martin [MM83] take a broader view and include all the seven factors. The rationale is that a portable, efficient, reliable and usable software should require less maintenance. We agree on the first factor, portability, because the MDSF software may move to different platforms over the next 30 years. Transporting a software to a new platform is just another instance of adaptive maintenance according to our definition. However, we see efficiency, reliability and usability as general software quality factors influencing the quantity of maintenance to perform, not the ability to perform it. Therefore, we retain testability, understandability, modifiability and portability as our high-level factors describing maintainability. Figure 1 illustrates the maintainability factors.

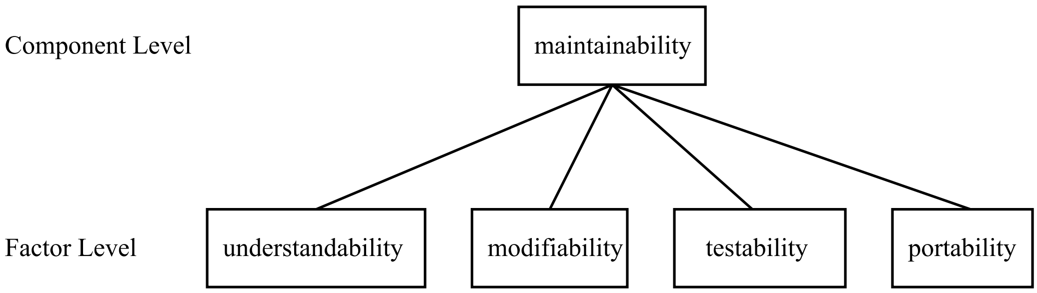

Figure 2 - Maintainability factors

4. - Development Life Cycle

MDSF is developed in three steps. Each step is an iteration over the development life cycle. Our model of the life cycle is derived from the water fall model described in [SPAR-SS-PL-1242A], which was derived from [DOD-STD-2167A]. Its consists of the following phases, milestone reviews and products of interest.

Phase	Milestone Review	Product
• Requirement Definition	System Requirement Review	Contract End Item Specification (CEIS)
• System Design	System Design Review	Requirements Specification (RS)

• Preliminary Design	Preliminary Design Review	Software Preliminary Design Document (SPDD)
• Detailed Design	Critical Design Review	Software Detailed Design Document (SDDD)
• Implementation	Test Documentation Review	Source Code, updated RS & SDD
• Integration & Test	Test Readiness Review	Updated Source Code, RS & SDD, Problem Reports
• Verification & Acceptance	Acceptance Review	Updated Source Code, RS & SDD, Problem Reports
• Sustaining Engineering		Updated Source Code, RS & SDD, Problem Reports

Our goal is to evaluate and monitor maintainability at every phase of the life cycle. It will be achieved by measuring criteria on products of each phase. A criterion should be as objective as possible and easily measurable. We wish to concentrate on criteria measurable before the system has cleared the final acceptance review. This is due to the contractual nature of the development process at CSA. We now provide a list of criteria by product.

5. - Products and Criteria

1 - Contract End Item Specification

1.1 - Readability: the ease of reading the specification.

2 - Requirements Specification

2.1 - Traceability: the ability to relate paragraphs of the requirements specification to paragraphs of the contract end item specification.

2.2 - Readability: the ease of reading the specification.

3 - Software Design Document

3.1 - Traceability: the ability to relate paragraphs of the software design document to paragraphs of the requirements specification.

3.2 - Coupling: the degree of interdependence among modules. It describes the mechanism used to share data between modules and the usage of the shared data. [Myer75]

3.3 - Cohesion: the interrelationship of elements contained in a module. It describes on what basis the elements where aggregated to form a module. This is also called *module strength* [Myer75].

3.4 - Size: the count of some lexical elements of a module. Examples are a line of source (either a comment, a PDL statement or an executable statement), an operator (+, :=, if, array, etc) or an operand (a variable, a constant or a label).

4 - Source Code[1]

4.1 - Traceability: the ability to relate programs to paragraphs of the software design document.

4.2 - Control Structure: determined by the conditional statements (alternation and iteration) of a module. It can be represented by a graph where the nodes are conditional statements and the arcs are sequential statements.

4.3 - Independence: the extent to which the source code can be executed on different computer architecture and on different operating systems.

4.5 - Readability: the existence and quality of comments (correspondance with RSD and SDD) in the source code.

4.6 - Doc. Accuracy: the extent to which the documentation describes the functionality implemented in the code.

4.7 - Consistency: the extent to which the source code contains uniform notation, terminology, and symbology within itself.

Table 0 describes the relationships between criteria and factors.

[1] Coupling, cohesion, hierarchy and size can also be measured on the source code. At the preliminary design and detailed design phase, metrics for these criteria will probably be computed manually, because of the informal syntax of the software design documents. During implementation, these metrics can be computed mechanically and, therefore, economically. There is a trade-off between cost of computation and timeliness of the metrics.

Factor\ Criterion	Understandability	Modifiability	Testability	Portability
Readability	X			
Traceability	X		X	
Coupling	X	X	X	
Cohesion	X	X		
Size	X	X	X	
Control Stucture	X	X	X	
Independence				X
Consistency	X			
Documentation Accuracy	X		X	

Table 1 - Relationships between criteria and factors[2]

[2] This is a preliminary view. Relationships will be revised before the prototype is demonstrated to determine if factors should be orthogonal.

6. - References

[BBKL78] Boehm, B.W., Brown, J.R., Kaspar, H., Lipow, M., MacLeod, G.J., Merrit, M.J. *Characteristics of Software Quality*, TRW series of software technology, North-Holland, 1978.

[DOD-STD-2167A] Department of Defense, Washington. *DOD-STD-2167A: Military Standard, Defense System Software Development*.

[EM87] Evans, M.W., Marcinak, J.J. *Software Quality Assurance & Management*, Wiley-Interscience, 1987.

[LS80] Lientz, B.P., Swanson, E.B. *Software Maintenance Management*, Addison - Wesley, 1980.

[MM83] Martin J. and McClure, C. *Software Maintenance: The Problem and its Solution*, Prentice-Hall, 1983.

[Myer75] Myers, G.J. *Reliable Software Through Composite Design*, New York, Van Nostrand Reinhold, 1975.

[SPAR-SS-PL-1242A] Spar Aerospace Ltd ATSG. *SPAR-SS-PL-1242A: MDSF Phase D1 Development Plan*.