

Whitepaper Draft

TODO: Title

Gautam Kumar
Instructor: Prof. Mark Kochanski

February 6, 2016

Contents

1	Software Maintenance and Maintainability	3
1.1	Types of Maintenance	3
2	Quantifying Maintainability	3
2.1	UML based metrics	3
2.2	Related quality attributes	4

1 Software Maintenance and Maintainability

Software maintenance as defined by the International Organization for Standardization is the process of modifying a software product after delivery to correct faults or to improve any quality attributes [1]

Maintainability is a quality attribute that describes the ease of introducing modifications into the software systems. Maintainability has four sub-characteristics namely analysability, changeability, stability and testability[1].

1.1 Types of Maintenance

ISO defines four types of software maintenance, Corrective, Preventive, Adaptive and Perfective.

Corrective maintenance refers to modifications done fix actual errors in the product while preventive maintenance is performed to introduce changes that can potentially prevent problems during the normal operation of the software product.

Adaptive and perfective maintenance are modifications introduced to enhance the core functionality of the software by improving one or more of its quality attributes.

2 Quantifying Maintainability

Understanding maintainability as a quality attribute is useful in learning the benefits of its presence in a product but to derive value from the process of evaluating software maintainability we require a method of measuring maintainability as a metric.

2.1 UML based metrics

One way to measure maintainability is to analyse the class diagram. This is especially useful in projects which use an object oriented programming language and have an existing UML diagram describing the software structure.

Genero et al [2] describe an elegant way of predicting the maintainability of software early in the design phase by measuring the size and structural complexity of the software project.

UML based metrics cannot independently describe the maintainability of a system because they suffer from the same basic flaw that UML adoption faces. Its the fact that UML isn't as widely adopted in an industry setting as one would expect. Though UML

is extremely well defined and documented its complexity and the learning curve needed to effectively adopt UML in an organisation setting serves as a serious impediment to its adoption which in turn reduces the value of using an UML based metric in measuring the maintainability of a software architecture.

2.2 Related quality attributes

Maintainability has a positive relationship with availability, flexibility, reliability, testability and a negative relationship with efficiency [3]. This means that an increase in flexibility, for example, could potentially improve the maintainability of the software system. An increase in efficiency could potentially hamper maintainability.

Understanding these trade off is important because indirect measurements as a factor of one of the aforementioned quality attributes can play a critical role when direct measurements of maintainability aren't available.

Bibliography

- [1] , I. International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering #2013; Software Life Cycle Processes #2013; Maintenance. *ISO/IEC 14764:2006 (E) IEEE Std 14764-2006 Revision of IEEE Std 1219-1998* (2006), 0_1–46.
- [2] GENERO, M., PIATTINI, M., MANSO, E., AND CANTONE, G. Building UML class diagram maintainability prediction models based on early metrics. In *Software Metrics Symposium, 2003. Proceedings. Ninth International* (Sept. 2003), pp. 263–275.
- [3] KARL, W. Software requirements. *Edition-2, Microsoft* (2003).