



Statistical Approach for Evaluating Second Brain Applications



Ghada Al-Hudhud*

Department of Information Technology, King Saud University, Saudi Arabia

Submission: December 31, 2017; Published: March 01, 2018

*Corresponding author: Ghada Al-Hudhud, Associate Professor, Department of Information Technology, King Saud University, Saudi Arabia, Tel: 00966552105605; Email: galhudhud@ksu.edu.sa

Abstract

Quality of any software product can be measured in terms of performance, reliability, maintainability... etc. Mind mapping applications can be thought of second brain and represents a mirror of our natural thinking. Free Mind is one of the most popular mind mapping application that is not evaluated by users yet. It is an open source tool which targets a large group of audience (student, teacher, employee, or anyone), and has many functionalities to be evaluated in term of the quality manner. Therefore, in this paper, the application Free Mind is evaluated through a statistical quality model that is based on Six Sigma Approach. The results are compared with ISO standards to determine the overall quality of the Free Mind application.

Keywords: Free mind; Quality evaluation statistical model; Six sigma

Abbreviations: QMP: Quality Management Principles; GUI: Graphical User Interface; DMAIC: Define Measure Analyze Improve Control; DMADV: Define Measure Analyze Design-Verify; DFSS: Define Identify Design Optimize Verify; Cpk: capability Index; LCOM: Lack of cohesion method; WMC: Weighted Method Count; DIT: Depth of Inheritance Tree

Introduction

The software's development is increasing dramatically day by day. Lots of auto-generate and easy-to-use platforms and tools are now available for everyone to implement a software and publish it to the market. Therefore, the market will be overstocked with thousands of products which might satisfy the user needs, but not necessarily meeting the software quality standards. So, for ensuring that the quality standards are met in software products, a term of quality management principles (QMP) has been introduced to be as "a set of fundamental beliefs, norms, rules and values that are accepted as true and can be used as a basis for quality management" [1]. ISO/IEC 9126 established quality software model standardization which defines the quality characteristics and their guidelines to be followed in order to reach the software quality assurance [2]. In this paper, the aim is to evaluate the quality of the "Free Mind" standalone application by following ISO/IEC 9126-1 standard [3]. The paper will be divided into sections; first section will go over the background. Second section reviews the related work of the quality assessment models and tools. While in the third section, the Free Mind description and features will be listed along with the quality management software being used. In the fourth section, the quality management will be discussed in terms of the policies and procedures, quality standards and model. In the fifth section, the quality plan will be showed.

Whereas the quality test result, the calculation of the total quality conformance rate and quality rate scale, are presented in the fourth section. Fifth section will explain the six-sigma process. Finally, the conclusion and recommendation are presented in the sixth section.

The evolution of mind map

A Mind Map is the external mirror of our natural thinking. It is a creative and logical means of note-taking and note-making that literally "maps out" the ideas. This technique used for mapping out thoughts using keywords which trigger associations in the brain to spark further ideas. It originated in the late 1960s by British psychology author Tony Buzan [4]. Ever after, mind maps drawn by hand in unlined paper. Since the evolution of technology and feeling of restricted by paper-based mind mapping, programmers sought a way to combine the creative nature of hand-drawn with the ease and flexibility of technology as in "Free Mind" software. With a shared vision of creating a brain-friendly mind mapping software that would benefit wider users to present ideas in powerful way. Accordingly, editing and arranging information will be easier than before. With all the benefits of computer based mind mapping, there is still a need to check and assess the quality characteristics of the software to ensure if it is satisfying or not.⁷

Background

In this section, an overview of the quality management main concepts and the evolution of quality assessment models are presented. Hence, a review some of the software quality tools, which are useful in quality testing are presented. Quality management is the process of managing the quality of a product. There are main concepts and terms which are used in the quality management field. First, the quality is defined as “the ability of a set of inherent characteristics of a product, product component, or process to fulfil requirements of customers” [5]. While the quality assurance (quality assessment) is “a planned and systematic means for assuring management that the defined standards, practices, procedures, and methods of the process are applied [5]”. In the quality assessment, for each product, a set of policies of interest is identified, procedures and measurements to evaluate the quality of it. So, the policy is “an expression of quality intentions, direction, and aims regarding to quality of products and processes [6]”. While the procedure is “ an approach and operational organization used to achieve the goals set by the quality policy “[7]. Finally, the measurement is an indicator defined to for measuring the performance, collecting data, and used to analyse the information [8].

Quality assessment

Software quality assessment is a critical process that is concentrating on the quality certification of the code in different aspects (functionality, usability, etc.) [9]. Software ranging from operating systems (i.e. Linux), middleware/ database technologies (i.e. MySQL) and end user software (i.e. Firefox web browser), requiring a standardized quality assessment to well-examine their quality of code and perform a white box testing and analysis on them on a standardized way [10,11]. In [5], they illustrate the evolvement of the quality assessment model by first designing McCall’s, Boehm’s until reaching ISO/IEC 9216 quality model which considered as the most widespread quality standard. McCall’s is the first modern quality standard proposed in 1977. It is formed by eleven quality factors as shows in Figure 1, product operation (efficiency, usability, integrity, correctness and reliability), product revision (maintainability, testability and flexibility) and product transition (reusability, portability and interoperability). Then Boehm’s quality model proposed in 1978 uses the same hierarch approach of McCall’s. It is represented by three high level characteristics: maintainability (the effort to understand, test and modify the product), portability (the effort to adapt new environment) and as-is utility (which requires the product to be reliable, efficient and easy to use) as shown in Figure 2. In 1991, the first international standard of software quality proposed (ISO/IEC 9216), it is an evolved model that is based on McCall’s and Boehm’s model approaches. It is considered with three main views of the software quality as shown in Figure 3 external quality (measuring the characteristics while executing the product), internal quality (measuring the characteristics if the product is not executed) and in-use quality (measuring the

characteristics from the end user point of view while executing it in different environments) [12,13].

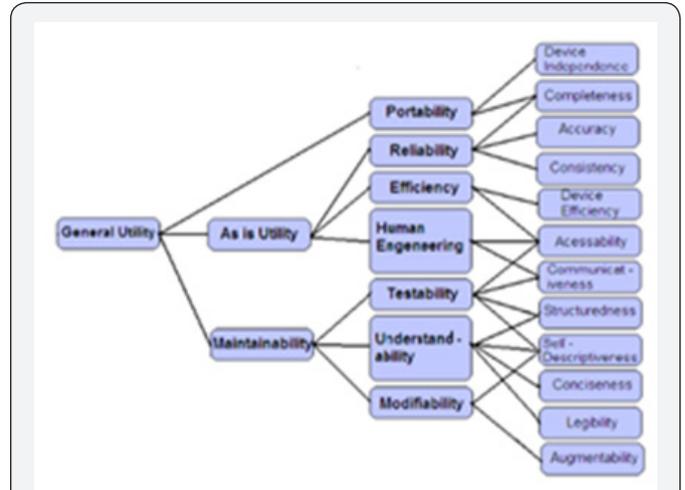


Figure 1: McCall's quality model.

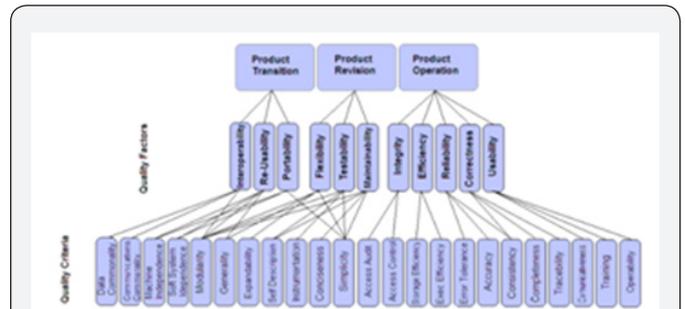


Figure 2: Boehm's quality model.

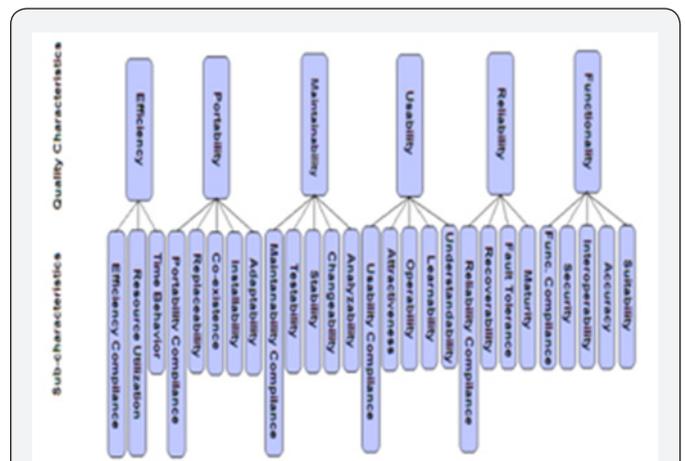


Figure 3: ISO/IEC 9216 quality model.

Quality assessment tools

Quality assessment tools are designed to assist reviewers and developers in focusing on concepts that are core for critical evaluation of the internal quality of applications. The tools were not designed to provide a list of factors comprising a numeric

score. The tools were specific to individual types of included study designs.

In [14] authors discussed the existing software metric tools interpret and implement the definitions of object-oriented software metrics. Eclipse Metrics plug-in is one of the tool that mentioned in the previous study. It is an open source metrics calculation and dependency analyzer plug in for the Eclipse IDE developed by Frank Sauer. It measures various metrics with average and standard deviation, detects the dependencies and graph it. More examples of quality assessment tools are presented in [14]. CodePro AnalytiX is a free Java software testing tool developed by Google and used in Eclipse IDE. It helps developers to improve software quality and working on eliminating costs and fixing errors during the developments process. CodePro tool has multiple software assessment and auditing features that help to keep track of the quality measures. This tool, along with a user guide [15], provides a standardized means to assess study quality and help to develop for study findings.

While, Find Bugs is a free code analyser software to look for bugs in a Java code using a static analysis. It is an open source and platform independent. Also, it can analyses programs compiled for any version of Java. The software is used as a stand-alone GUI application or plug-in available for Eclipse, NetBeans, IntelliJ IDEA ... etc. Additionally, it has two user interfaces: a graphical user interface (GUI) and a command line user interface. The bug patterns that are extracted by Find Bugs are performance bug, security bug, correctness bug ... etc. [16]. Find Bugs can help to improve the quality of your code and eliminate bugs [17].

Proposed quality assessment framework

Application Description: Free Mind is an open source, mind-mapping software programming tool which is written in Java. It has different releases; in this paper, the quality assessment against the latest version which is 0.9.0, 2011 is statistically evaluated. Free Mind is considered as a knowledge organizer tool which is used for organizing, linking, integrating and sharing thoughts by drawing a mind map in a more efficient and user friendly way as shown in Figure 4.

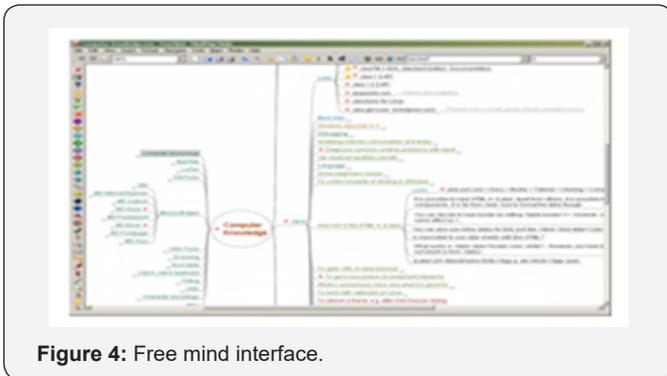


Figure 4: Free mind interface.

Application features: In Free Mind, users can create, edit, delete and format nodes in an easy and attractive way as shown

in Figure 5. The formatting feature can be used by creating trees of plain text notes enriched with colors, icons, cloud-shapes, and other graphics. Whereas the colors will be used to differentiate between the completed and non-completed sub-ideas, icons to prioritize a node and cloud-shapes will help in grouping together the nodes that are interrelated [3]. Also, it has a set of advanced features as shown in Figure 6 [3].

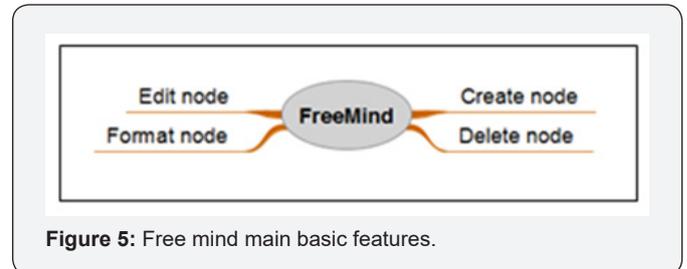


Figure 5: Free mind main basic features.

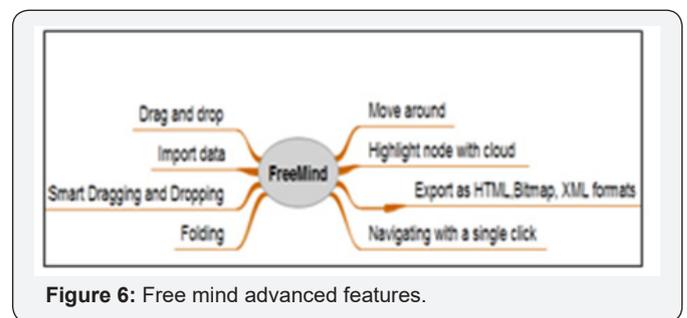


Figure 6: Free mind advanced features.

- I. **Folding**-Free Mind provides the ability to hide or display all the related information (children)of a specific node. This is helpful when the mind map is large and the user wants to fold (hide) the children and leave the main nodes only; for simple representation.
- II. **Navigating with a single click**- This feature allows the user to easily move in the map using the keyboard arrows (up - down - left - right). Also, he can use (Escape) to move directly to the root node or use (page up/ page down) to move to the top or the bottom of the map.
- III. **Smart Dragging and Dropping**-This feature provides the possibility for dragging nodes or styles of nodes and dropping them to be placed directly as a child’s or siblings into another node in the mind map.
- IV. **Smart copying and pasting**- This feature helps in copying the mind map and all its descendants to the clipboard. While the user can also paste a plain text or HTML file from outside to the Free Mind; the Free Mind will convert their structures into nodes and children to be represented as a mind map.
- V. **Exporting map as HTML/XHTML**-This feature converts the mind map to a standard, hierarchical text structure; as an HTML or XHTML, so the user can easily integrate it with any website.
- VI. **Importing map**-Free Mind allows the user to import a folder to be viewed as a mind map; it will convert all the

subfolders and their structures as a tree to be represented as mind map.

VII. **Storing maps in XML format-** Free Mind allows the user to store the mind maps in XML formats; this will let him be free to switch to another mind map application and import his XML mind maps to complete his work in it.

Quality management software

CodePro AnalytiX and Eclipse Metrics Plug-in are used as quality management tools. Foremost, CodePro AnalytiX can be easily integrated into Eclipse environment as plug in; using automated source code analysis to determine the quality issues [18]. It is considered as a useful, comprehensive, and easy to use Eclipse Java code review tool. Secondly, Eclipse Metrics Plug-in along with the CodePro AnalytiX are used. Using the capabilities of both tools, the proposed principles (reliability, maintainability, portability and usability) are tested. This in turn would provide some statistics to help the developers in the future to enhance it, since it is an open source software. CodePro AnalytiX version 7.1.0 and Eclipse Metrics Plug-in version 1.3.6 will be used with

Eclipse version 3.7 (Figure 7) [19].

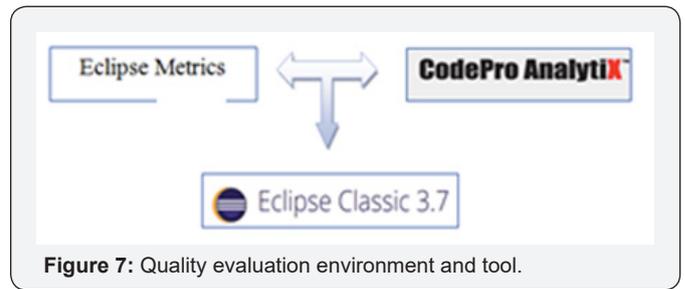


Figure 7: Quality evaluation environment and tool.

Planning quality assessment

ISO/IEC 9126-1 standard introduced a quality model which defines six characteristics (policies) to be used for the software quality assessment (functionality, reliability, usability, efficiency, maintainability and portability). Each of these characteristics has a set of sub-characteristics (procedures) to be measured using specific metrics [20]. In this section, a set of policies and procedures which will be followed in the Free Mind’s quality testing are listed (Table 1).

Table 1: Policies and Procedures.

Policy	Procedure	ISO/IEC 9126-1 Definition
Reliability	Fault Tolerance	Degree to which a system, product or component operates as intended despite the presence of hardware or software faults [20].
	Recoverability	The capability of the software product to re-establish a specified level of performance and recover the data directly affected in the case of a failure. [20].
Maintainability	Changeability	The capability of the software product to enable a specified modification to be implemented. [20]
	Stability	The capability of the software product to avoid unexpected effects from modifications of the software. [20]
Portability	Adaptability	Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments [20].
Usability	Learnability for Reuse	The capability of the software product to enable the user to learn its application. [20].

Policies& procedures

In this quality assessment, Free Mind will be technically evaluated using the following policies and procedures:

a. **Reliability** of Free Mind shall maintain its performance and stability under any sudden condition. So, this policy would be ensured by Fault Tolerance procedure; i.e. if the user performs different events which might lead to an error in the application (i.e. clicking many times on create node and causing a stress by sending lots of requests). Then, Free Mind shall avoid any crash and handle this stress without letting the user feel about anything. And also Recoverability

procedure; i.e. if the user creates a mind map with lots of nodes and texts bind with, then by mistake he deleted the mind map or part of it. In this case, Free Mind shall allow the user to recover his deleted map without causing any frustration to him [21].

b. **Maintainability** of Free Mind shall be modified easily due to any errors or changes in the functions. The policy would be ensured by ensuring the Changeability procedure; i.e. The developer shall be able to modify the code of Free Mind features easily, if there is any change in the requirement or an error. And also the stability procedure; i.e. the Free Mind application shall be stable if any feature has been

modified, and do not let this particular modification effect other classes and features.

c. **Portability** of Free Mind shall work on different operating systems without any need for extra efforts. This policy would be ensured by following the Adaptability procedure; i.e. the developer shall be able to easily adapt the code of Free Mind to meet any future requirements or to make it work in any working environment.

d. **Usability** of Free Mind shall be easy to use, learn and understood while providing a simple and attractive interface

for the user. This policy would be ensured by maintaining Learn ability for Reuse procedure; i.e. Free Mind’s classes shall be learned easily if the developer wants to reuse them in different system [22].

Proposed quality management plan and model

Quality management plan

Figure 8 & 9 Table 2

Table 2: Quality Standards Relation.

Policy	Procedure	Measurement	Relationship	Tool
Reliability	Fault Tolerance	Line of code (LOC)	Positive	Eclipse Metrics
	Recoverability	Efferent coupling between packages (Ce)	Negative	Eclipse Metrics
Maintainability	Changeability	Lack of cohesion method (LCOM)	Negative	Eclipse Metrics
	Stability	Instability between packages (I)	Negative	Eclipse Metrics
Portability	Adaptability	Weighted Method Count (WMC)	Negative	Eclipse Metrics
Usability	Learn ability for Reuse	Depth of Inheritance Tree (DIT)	Negative	CodePro AnalytiX

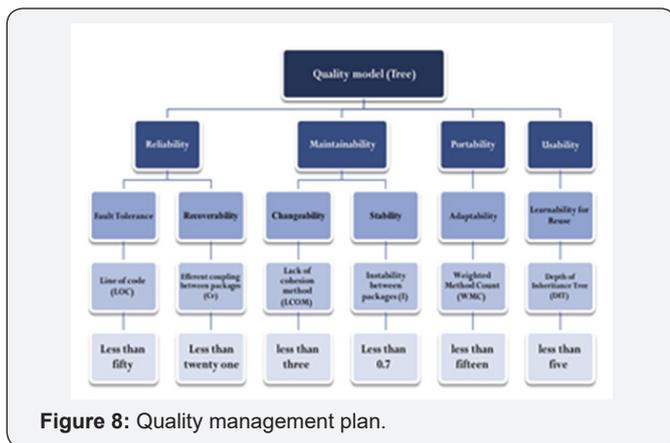


Figure 8: Quality management plan.

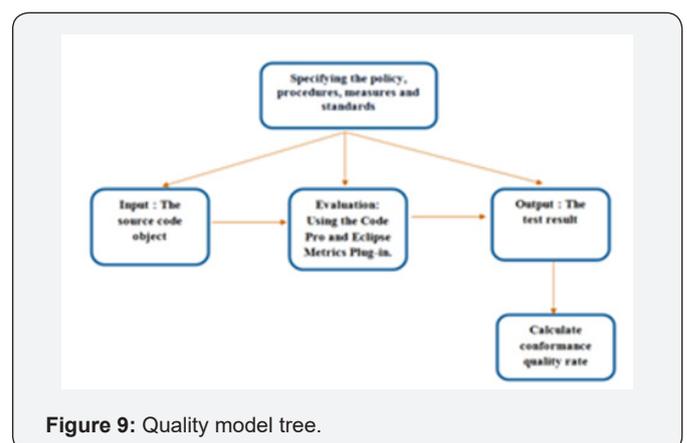


Figure 9: Quality model tree.

Quality standards relationship: Table 2 shows the relationship between the measurements and the procedures which is either positive or negative relationship. Positive relationship is a relationship between two variables such that their values increasing or decreasing together like fault tolerance and line of code. But the negative relationship is a relationship between two variables in which one variable increases as the other decreases, and vice versa like the recoverability with efferent coupling between packages, changeability with efferent coupling between packages, stability within stability between packages, adaptability with weighted method count and usability with depth of inheritance tree[23].

Quality tree model: Quality tree model is a diagram that illustrates the relationship between the policies, procedures, measurements and standards as shown in Figure 9.

Experimental results

Comparing quality results with standards: Table 3 calculating

the quality testing result of each of the metrics and obtaining the procedures’ quality conformance rate by following this formula: ((result/ standard) * weight) or (weight), if it is within the range; Zero, if it is out the range.

The calculation of the total quality conformance rate: As indicated in Table 3, the quality conformance rate for each of the procedures has been calculated individually. Table 4 shows the details of the total quality conformance rate, by representing the policy and its procedures along with the quality rate for both the procedures as an individuals and the overall policy’s quality rate [24].

Quality rate scale

To determine the overall quality of the Free Mind application, the quality rate scale is based on the [25]. The rates are scaling up from poor, weak, moderate, fairly good, good until reaching the outstanding quality rate; based on the quality conformance percentage. In our case, the quality conformance percentage for

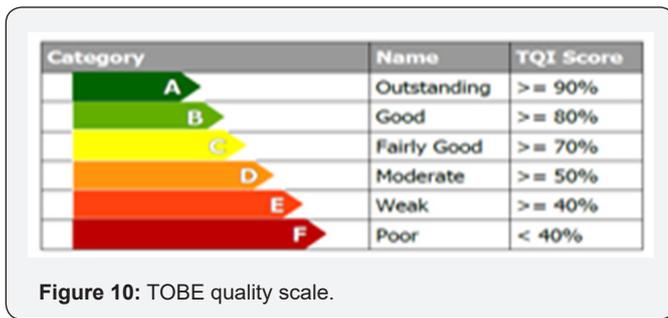
Free Mind (0.5.0) and (0.8.0) are respectively 67.08%, 67.1% considered as a good quality (Figure 10). which is moderate. While Free Mind (1.0.0) is 82.3% which

Table 3: Details calculation of Quality Conformance.

Policy	Procedure	Policy Quality Rate (version 0.5.0)	Policy Quality Rate (version 0.8.0)	Policy Quality Rate (version 1.0.0)
Reliability	Fault Tolerance	17.50%	17.50%	17.70%
	Recoverability			
Maintainability	Changeability	15%	15%	30%
	Stability			
Portability	Adaptability	24.58%	24.60%	24.60%
Usability	Learn ability for Reuse	10%	10%	10%
Total Quality Conformance Rate		67.08%	67.10%	82.30%

Table 4: Calculation measures Capability Index Cpk, k=3.

Policy	Measurement		UTL	LTL	Cpk	Capable or not	Result	
Reliability	Fault Tolerance	161	44.59	=161+ 133.77 =294.77	=161- 133.77 =27.23	$K=3$ $= \min \left\{ \frac{161 - 27.23}{133.77} \mid \frac{294.77 - 161}{133.77} \right\}$ $= \min \{1, 1\} = 1$	Capable	The process is capable at the 3σ level
	Recoverability	264	8.98	=264+ 26.94 = 290.90	=264- 26.94 = 237.06	$K=3$ $= \min \left\{ \frac{264 - 237.06}{26.94} \mid \frac{290.90 - 264}{26.94} \right\}$ $= \min \{1, 0.99\} = 1$	Capable	The process is capable at the 3σ level
Maintainability	Changeability	0.2	0.0074	=0.2+ 0.0222 = 0.222	=0.2- 0.0222 = 0.178	$K=3$ $= \min \left\{ \frac{0.2 - 0.178}{0.0222} \mid \frac{0.222 - 0.2}{0.0222} \right\}$ $= \min \{1, 1\} = 1$	Capable	The process is capable at the 3σ level
	Stability	0.698	0.092	=0.698+ 0.276 = 0.974	=0.698- 0.276 = 0.422	$K=3$ $= \min \left\{ \frac{0.698 - 0.422}{0.276} \mid \frac{0.974 - 0.698}{0.276} \right\}$ $= \min \{1, 1\} = 1$	Capable	The process is capable at the 3σ level
Portability	Adaptability	281.66	52.1	=281.66+ 156.3 = 437.96	=281.66- 156.3 = 125.36	$K=3$ $= \min \left\{ \frac{281.66 - 125.36}{156.3} \mid \frac{437.96 - 281.66}{156.3} \right\}$ $= \min \{1, 1\} = 1$	Capable	The process is capable at the 3σ level
Usability	Learn ability for Reuse	1.6	0.43	=1.6+ 1.29 =2.89	=1.6-1.29 =0.31	$K=3$ $= \min \left\{ \frac{1.6 - 0.31}{1.29} \mid \frac{2.89 - 1.6}{1.29} \right\}$ $= \min \{1, 1\} = 1$	Capable	The process is capable at the 3σ level

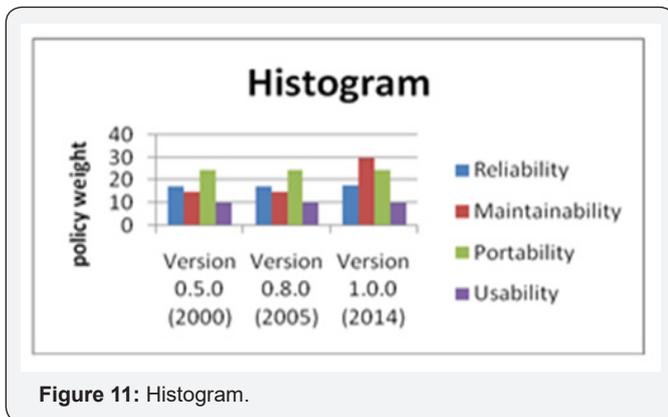


Six-sigma process

Six Sigma is a statistical quality assessment approach that provides a methodology to follow a continuous improvement in customer satisfaction and profit. It has three key methodologies: DMAIC, DMADV and DFSS. DMAIC (Define-Measure-Analyze-Improve-Control) is used to improve an existing business process while DMADV (Define-Measure-Analyze-Design-Verify) is used to create a new product that meets the customer’s needs. And, DFSS (Define-Identify-Design-Optimize -Verify) is used to design or redesign a product or service from the ground up [16]. In our project, we will use DMAIC methodology because we already have an existing system. Six sigma process and charts are designed using QI Macros 2016 tool [17]. The tools that we used are: flow charts, run diagram, control charts, histogram, p are to analysis and capability Index (Cpk).

Histogram analysis

Figure 11 shows the histogram which compare actual weight gained by the versions with the policy weight.

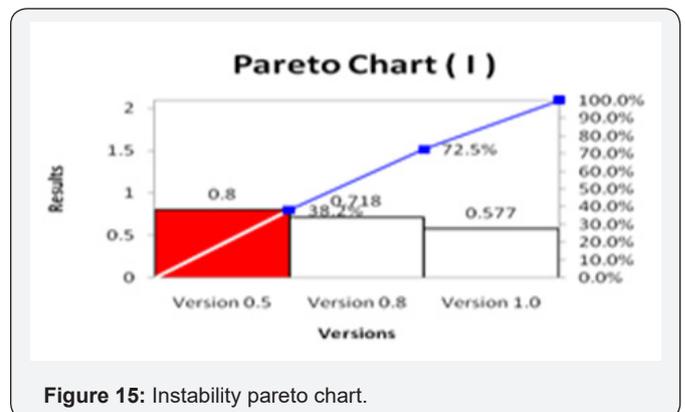
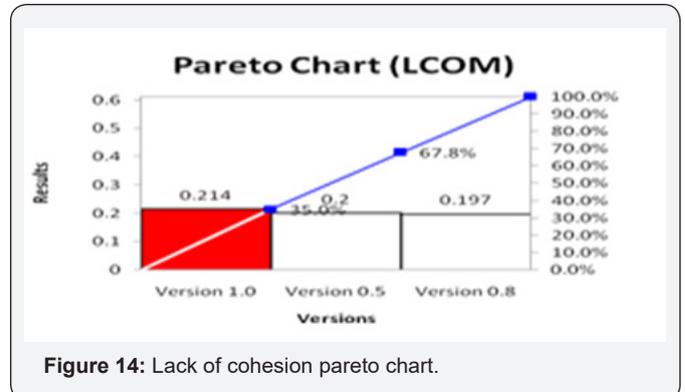
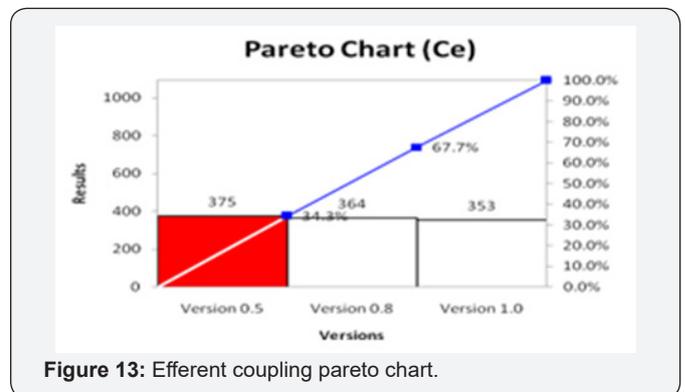
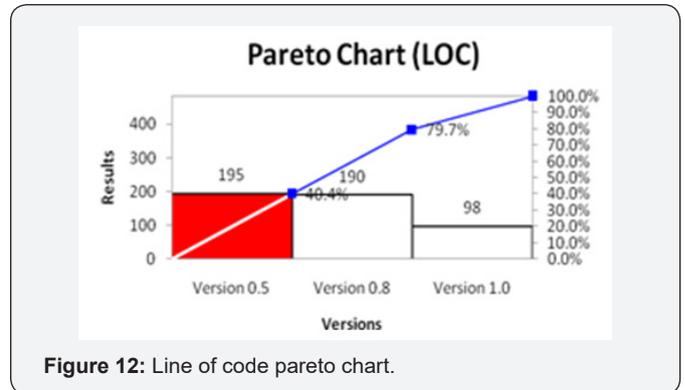


Pareto Analysis

Pareto chart created using QI Macros 2016 [17] to present the improvement analysis of each measure in multiple versions (Figure 12-17).

- Line of code
- Efferent Coupling
- Lack of Cohesion

- Instability
- Weighted Method Count
- Depth of Inheritance



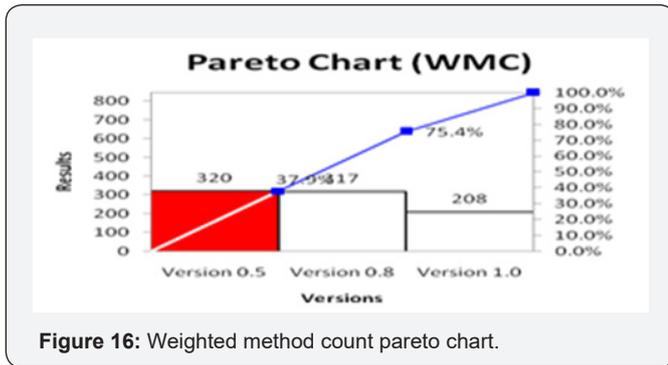


Figure 16: Weighted method count Pareto chart.

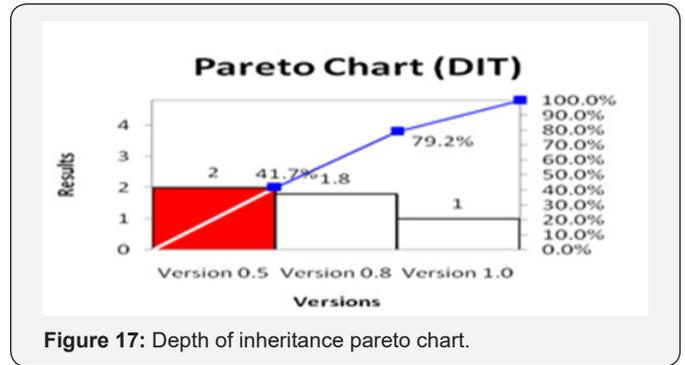


Figure 17: Depth of inheritance Pareto chart.

Average & Variation

Table 5 calculates the average and the standard deviation between the versions in terms of the quality measurements. The average is calculated using the following formula: $(R1+R2+R3)/3$; where R1, R2 and R3 represent the results of

version1 (0.5.0), version 2 (0.8.0) and version 3 (1.0.0). While the standard deviation is calculated using the following formula:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

where the N is the number of versions, xi is the result from the versions, and M is the average (mean).

Table 5: Average & Variation.

Policy	Measurement	Version Result(0.5.0)	Version Result(0.8.0)	Version Result(1.0.0)	Average	Standard Deviation
Reliability	Fault Tolerance	195	190	98	161	0.445
	Recoverability	375	364	353	264	0.898
Maintainability	Changeability	0.2	0.197	0.214	0.2	0.007
	Stability	0.8	0.718	0.577	0.698	0.092
Portability	Adaptability	320	317	208	281.66	0.521
Usability	Learn ability for Reuse	2	1.8	1	1.6	0.43

Capability Index (Cpk)

The capability index shows how well the performance measure fits the specification based on a given tolerance level.

To find Cpk, we need to calculate standard deviation (σ) which was calculated in the previous section. Also, we need to estimate the Upper/Lower Tolerance Limits (UTL, LTL) based on the measure standard. We will start by using $k=3$:

If the $Cpk \geq 1$ which means the process capable at the 3σ level. But, if the result shows that the measure is not capable which means the $Cpk < 1$, Cpk will be recalculated with $K = 6$ using the following formula:

Table 6: Comparison between the three Free Mind versions.

Version 0.5.0 (2000)		Version 0.8.0 (2005)		Version 1.0.0 (2014)	
Test results	Conformance rate	Test results	Conformance rate	Test results	Conformance rate
The total number of methods greater than 50 lines is 195.	$X=(0.18 / 7360)$ $X * 195 = Y$ $0.18 - Y = 0.175$ $=17.5 \%$	The total number of methods greater than 50 lines is 190.	$X=(0.18 / 7357)$ $X * 190 = Y$ $0.18 - Y = 0.175$ $=17.5 \%$	The total number of methods greater than 50 lines is 98.	$X=(0.18 / 6054)$ $X * 98 = Y$ $0.18 - Y = 0.177$ $=17.7 \%$

$$C_{pk} = \min \left\{ \frac{\bar{X} - LTL}{k\sigma}, \frac{UTL - \bar{X}}{k\sigma} \right\}$$

Where,

$$UTL = Mean + k\sigma \text{ and } LTL = Mean - k\sigma$$

The experimentation results indicate that the overall quality percentage for Free Mind (0.5.0) and (0.8.0) are respectively 67.08%, 67.1% which is moderate. While Free Mind (1.0.0) is 82.3% which is considered as a good quality based on rate scale taken from [19]. It has an excellent level of fault tolerance and adaptability. Otherwise, there is few negatively effect from recoverability that is measured by efferent coupling between packages(Ce)(Table 6).

The total number of methods 7360	X= 0.00002447 Y=0.00464863	The total number of methods 7357	X= 0.00002447 Y=0.00464863	The total number of methods 6054.	Where X= 0.00002937 Y=0.177086
Efferent coupling between packages [14] total is 375	0% (out of the range)	Efferent coupling between packages [14] total is 364	0% (out of the range)	Efferent coupling between packages [14] total is 353	0% (out of the range)
Lack of cohesion method is 0.2 which is less than 3	0.15*100= 15% (within the range)	Lack of cohesion method is 0.197 which is less than 3	0.15*100= 15% (within the range)	Lack of cohesion method is 0.214 which is less than 3	0.15*100= 15% (within the range)
Instability between packages is 0.8 which is more than 0.7	0% (out of the range)	Instability between packages is 0.718 which is more than 0.7	0% (out of the range)	Instability between packages is 0.577 which is less than 0.7	0.15*100= 15% (within the range)
The number of methods greater than 15 is 320	X= (0.25/19219) X* 320= Y 0.25- Y= 0.2448 = 24.58 %	The number of methods greater than 15 is 317	X= (0.25/19219) X* 317= Y 0.25- Y= 0.2458 = 24.6 %	The total number of methods greater than 15 is 208	The total number of methods greater than 15 is 208
The total number is 19219	X=0.00001301 Y= 0.004163	The total number is 19219	X=0.00001301 Y= 0.00412352	The total number of methods is 12456	X=0.00002007 Y= 0.00417469
The DIT is 2 which is less than 5	0.1*100= 10% (within the range)	The DIT is 1.8 which is less than 5	0.1*100= 10%	The DIT is 1 which is less than 5	0.1*100= 10%
67.08%		67.10%		82.30%	

Conclusion and Recommendations

Quality Management ensures the increasing on revenues and higher productivity for the organization. Also, it provides a great benefit to both the development team and users. In this paper, the quality of Free Mind is evaluated by doing a quality model that includes suitable policies, procedures, and measures. We use CodePro and Metrics plug-in to produce the result. The conformance rate was calculated for these measures according to the ISO standards. Finally, high quality means success. And, happy customers will be returning and that is what every organization strives for it. Therefore, to improve the application quality, we highly recommend to reduce the (Ce) to decrease the time that is needed by the developer to adapt the code of Free Mind in order to fix it or to meet any future requirements. For Capability Index calculation the result match the quality model results, for all policies at the 3σ level. For future work, we will have a comprehensive comparison analysis between different characteristics of application versions. In addition, we will use another policy and procedure to have more accurate result.

References

- Quality Management Principles (2015), (1st edn), Switzerland.
- Software engineering product quality (2001) ISO/IEC Standard 9126.
- <http://FreeMind.sourceforge.net/wiki/index.php/Documentation>
- <http://www.tonybuzan.com/about/mind-mapping/>
- Team (2006) CMMI Product. "CMMI for Development, version 1.2.
- <http://www.businessdictionary.com/definition/quality-policy.html>
- <http://ccm.net/contents/610-establishing-a-quality-procedure>
- http://www.cpc.unc.edu/measure/prh/rh_indicators/crosscutting/service-delivery-ii.h.2/quality-assurance-approach.html
- Veiga J (2010) Quality Assessment of Java Source Code.
- Gousios G, Karakoidas V, Stroggylos K, Louridas P, Vlachos V, et al. (2007) Software quality assessment of open source software. Proceedings of the 11th Panhellenic Conference on Informatics.
- Adewumi A, Misra S, Omoregbe N (2013) A Review of Models for Evaluating Quality in Open Source Software, IERI Procedia 4: 88-92.
- <http://metrics.sourceforge.net/>
- <https://www.dropbox.com/sh/nxl0ep3vi3x657h/AABTJ1zugfMF27PutpVb2g6Ra?dl=0%22>
- Lincke R, Lundberg J, Löwe W (2008) Comparing software metrics tools. In Proceedings of the 2008 international symposium on Software testing and analysis, pp. 131-142.
- <http://www.barneyassoc.com/documents/WP-CPA-eclipse.pdf>
- <http://www.cs.cmu.edu/~aldrich/courses/654/tools/Sandcastle-FindBugs-2009.pdf>
- <http://findbugs.sourceforge.net/>
- <https://dzone.com/articles/codepro-integration-eclipse>
- <http://www.eclipse.org/downloads/packages/eclipse-classic-37-indigor>
- Information technology - Software product quality, 2000ISO/IEC Standard 9126-1.

21. <http://www.arisa.se/compendium/quality-metrics-compendium.html>
22. <http://staff.unak.is/andy/StaticAnalysis0809/metrics/ce.html>
23. <http://staff.unak.is/andy/StaticAnalysis0809/metrics/i.html>
24. <http://realsearchgroup.org/SEMaterials/tutorials/metrics/>
25. Jansen P (2015) The TIOBE Quality Indicator a pragmatic way of measuring code quality. TIOBE Qual Indic 3(6).



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/BBOAJ.2018.04.555661](https://doi.org/10.19080/BBOAJ.2018.04.555661)

Your next submission with Juniper Publishers

will reach you the below assets

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats (Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission

<https://juniperpublishers.com/online-submission.php>