An Object-Oriented Software Metric Tool to Evaluate the Quality of Open Source Software

Abdullah Al Hussein

Assistant Professor
College of Computer and Information Sciences
Majmaah University, Al Majmaah, Kingdom of Saudi Arabia

Abstract
Software efficiency and software superiority control have encouraged the investigation of software metrics technology. Open source software development has increased the importance and directed towards the study on the software quality metrics and the advancement of automated tools to provision the evaluation of software quality metrics. For the successful support of the software assessment, the software quality metrics tool is proposed to provide the quality metrics prototypically. The main aim of this research work is to deliver an automatic software quality metrics tool which supports the managers and users to evaluate the project or foundation code of the open source software package and therefore assess the excellence of the software as per the detailed classified software metrics prototypical. The overall goal of this research work is to the programme quality assessment of basic code for a permitted and open foundation software by study the software metrics and develop a tool. This research work will be built on software quality metrics that are, collective and arranged by professionals, will provide a programmed examination of features such as flexibility, accuracy, modularity, and quality of software maintenance.

Key-words:
Open source software, software metrics, software evaluation, software engineering and quality assurance

1. Introduction

Many open source software's are analyzed using the software quality improvement tools and has remained newly to advance software superiority [1]. The software quality metric tools practice diverse categories of data around the software assembly and past. Mostly, the usage of free and open basis software for the software industry has enhanced confidently in the past era, prompting the worldwide economy implicitly. However, there are many firms and administrations that are still unenthusiastic in adopting open source software's are permitted or juridical subjects, profitable features or uncertainty of the open foundation expansion procedure, products, and sustenance [2]. These research work will be used to evaluate the software application development, to accomplish the brilliance of the foundation code, calculate software quality metrics, examine the software coding rubrics. Overall, this research work will benefit software quality maintainers to recognize huge volumes of information approaching from foundation code sources.

Currently, the software companies face an overabundance of this open and proprietary software substitution intended for respectively precise job or functionality. The software program needs healthy distinct measures to assess the brilliance of such open source software with that the code handlers can choice the substitutions that unsurpassed meet their necessities. Generally, all the estimation is done by the examination, intelligence from previous handlers, and documents analysis. Still, the open source software licenses an additional deep valuation based on data grown from foundation code investigation. For example, the source code metrics give the casual to investigate suppleness, bugs, difficulty, and readability of the source code, which are significant features for the positive development of a software application [3]. Then again, gathering this information and evaluating this data is a difficult job which needs the assistance of automatic tools.

2. Literature Review

Several software evaluation metrics that will benefit in the job of assessing the excellence of a portion of the software application. The source code evaluation metrics can illustration the difficulty and the dimension of software programs. It also shows the inner and outdoor dependences with the effectiveness of encapsulations and the ideas after the examination suites the handling and excellence of software applications. In adding to the foundation code examination, in maximum permitted software developments, an individual can get extra applicable data by retrieving its version controller organization, a list of contributors and creators, and database of bug intelligence and enhancements [4]. Since this information, is conceivable to get information about the team presentation, in what manner quickly the described faults are modified by the software development community, others significant features.

Manuscript received April 5, 2017
Manuscript revised April 20, 2017

Now, some open source software evaluation tools gather such information in an automatic technique. Therefore, it is possible to make an implement to inspect the foundation code of hundreds and thousands of free software developments and stock the consequences in a solitary place, [5] helping as a foundation for investigating software excellence. Evaluation metrics can produce different information tables, diagrams, and locations, allowing practical, unbiased investigation of many choices. Because most valued asset for the open source community is the code of the project. Then, the intention of this study is to develop a new tool intended for automatic excellence assessment of open software application developments, grounded on the metrics nominated and clear in this research work.

According to [6], currently, the practice of software metrics at the backend, in addition, the middleware software groups are extensively available, though the implementation of open source efficiency tools and commercial applications that are in the front end is increasing. Software metrics remain powerfully used to advance the task serious applications, facilities, and products. Numerous engineering areas are accepting the software metrics at diverse duties, through many attentions. The research work [7] review, 93% of the perpetrators supposed that they take encountered their excellence potentials but in nearly times, even beat the quality of results. The fulfillment concerning price remained on a comparable near at 86%. Price is the main acceptance driver, though additional influences are also important as 65% of all innovativeness users name reducing in total software price as the main incentive to use open source, which is distant outside just the value feature. Individuality influence, not to be protected by a solo merchant, is very significant for 53% of all handlers. The software quality metrics works defines a large diversity of metrics, [9] such as dimensions metrics (total lines of code, organization report and occupations facts), Halstead's Produce metrics (program language, measurement, and capacity) Difficulty metrics (cyclometric difficulty, data stream, and bumps), and Excellence metrics (maintainability, fault and dependability metrics).

But, all these software quality metrics are not identical filling [10] as jagged by the SEI statement by means of, in the previous, maximum metrics have been well-defined by a distinct and at that time verified and used individually in a very restricted atmosphere. Currently, valuable metrics and illustrations cannot be dragged off the ledge and used usually, cautious application of accessible metrics and replicas can harvest valuable outcomes if they remain adjusted to a situation. Subcontracting also raised newly in the software business, together with relationships of appointment preservation provision area and in accepting software established by third gatherings. A significant test in this framework is in what way to speedily estimation and distinguish the organizations advanced underneath subcontracting agreements [11].

Metrics for evaluation have been widely used in many disciplines of engineering, up till now in the mainframe software expansion business around still have remained approximately doubts approximately their use of metrics in this software quality ground. Through current ages, the benefits in software metrics practice consume grownup together importantly and progressively in software growth business [4]. Through the development programme and manager concentrating on software effectiveness and software excellence, there exist wants for an improved rehearsal of software development and software metrics throughout the procedure of expansion. Newly object concerned with technology is attractive additional and more widespread in engineering software growth surroundings. This information proposals provision to offer software creation with progressive excellence and smaller care costs. Meanwhile, the old-style software metrics goals at the process-oriented software growth and it cannot attain the condition of the object concerned with software [5], an established of novel software metrics modified for the advents of object knowledge is important in need. Consequently, object-oriented metrics formerly advances an important portion of object known as fit as a respectable software business.

Toward put on the object-oriented metrics in software expansion technology in exercise successfully, an instinctive object-oriented metrics assessment tool is important to care the activities throughout the procedure of open source software assessment. Currently nearby now exist approximately software assessment metrics tools to intention at this investigation unbiassed. However, maximum of them individual produce convinced unstructured autonomous metrics results. To attain some expressive metrics information in user and association level, it is compulsory to direct metrics illustrations to contain the accompanying metrics groups into a united agenda. For this, in this research work, an object-oriented evaluation metrics tool to care such reproductions is significantly wanted and established. This research work to develop a model and designed to a degree the development or foundation code of the object-oriented package mechanically and thus assess the quality of the software rendering to the counted confidential metrics prototypical. Investigation effort delivers an automatic mechanism for software metrics handlers to modify the selected metrics prototypically and cruise the particulars of the software metrics prototypical by using a metrics prototypical database in the tool.

This research work differs from the works discussed in the literature review and in the field mainly in the proposal of a tool that allows the combination of a set of metrics and presents the results of software quality assessment in a language and format that is easily understandable. In addition, the tool lets expert developers configure ranges of values for each metric, associating them with different quality assessments. Thus, the measurements for a specific piece of software will be accepted or not depending on the recommended values for the metrics set up by the user.

This study defines the investigation and connections in emerging software metrics prototypical to assess open foundation software. This study paper systematized as follows in, unit 3 particulars the project of software assessment metrics, which includes three modules: investigating, manipulative and displaying. Unit 4 confers the package examination scheme which is constructed in and the request of software metrics tool component-based software growth situation discourses. Unit 5 debates the process execution and outcomes. Finally, section 6 gives the conclusion.

3. Software Evaluation Metrics Tool

This research model is intended and created to provision object-oriented software excellence assessment for open source software tools. The assessment tool brands it informal for the ordered prototypical to be clear or built and put into exercise. Later software metrics handler adapts the metrics perfect whose minor level metrics can be nominated after a list of performed metrics, the handler can stock its requirement in the prototypical database. Conferring to the full prototypical, the implement computes the connected package data and originates the standards of the prototypical from the data file which comprises complete information nearby the package, which is formerly deposited into metrics outcome database. Figure 1 demonstrations the construction of the proposed research model. The figure 1 shows, the software quality metrics quantity contains 3 major stages

a) Evaluating
b) Scheming
c) Presenting

As per 3 stages, the implement is separated into 3 corresponding components. In the assessing stage, the examination visible end investigates foundation code, abstracts software data and stores it interested in the database through the data storage area. The GUI is established to inspect the foundation code syntactic and semantic which comprises both the interface and execution portion of the tool.

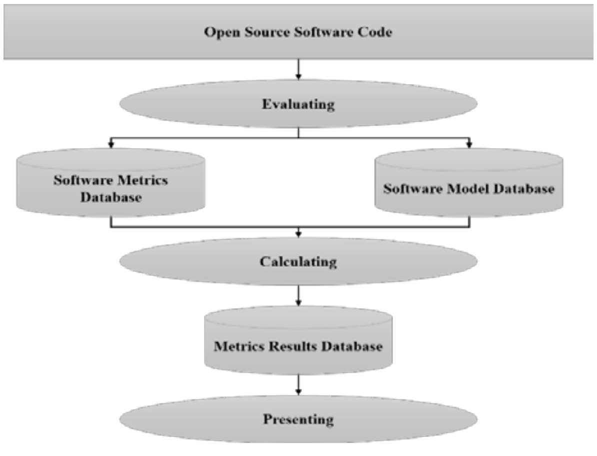

Fig. 1 Architecture model

While the scheming stage, affording to the nominated model into the model database which has previously remained altered by metrics handler, altogether the standards of the model remain considered from the data store and at that time stored into metrics outcome database. The prototypical data store is used to stock the description of approximately graded representations and the metrics outcome data store is to stock the outcome standards of the intended graded prototypical.

The software quality metrics representations are off the record as method, class and assembly level metrics prototypical founded on the choice of the restrained object. As soon as one precise technique of a class is assessed by scheming the metrics information on or after the information inside one technique of the class, the corresponding prototypical is classified as technique metrics prototypical. Uncertainty one class is assessed, the compulsory data for finding the metrics information is frequently imperfect inside the opportunity of one class, at that time the prototypical is classified as class metrics prototypical. Correspondingly, to assess the entire open source software application, the possibility of the info spreads to additional than one class, and this metric prototypical is called organization metrics prototypical. Respectively model is prepared as a graded illustration. The worth of the higher level lump is intended founded on the lower level node.

The other trendy prevalent software metric representations, substances in the prototypical except for the substances of the bottommost metrics equal can usually be measured on or after the standards of inferior level substances affording to the corresponding mass value allocated to them. The standards of the lowermost metrics level can be straight intended on or after the foundation code data. This research work defines a traditional of ten to twenty metrics, classified conferring to their exertion, founded on together non-functional features an investigation of the foundation code. Over measured experimentations and numerical investigation, individually metric will be assessed as to whether it demonstrations important associations with the excellence of free software as professed by professionals, therefore permitting us to examination whether that they can be used as a consistent indication for defining excellence.

Toward outline this established metrics, firstly, the system composed the thoughts of a set of software designer's specialists in software expansion, on or after business, college, and free software communal. The worth of individual piece in the prototypical must be regularized so that it harvests a value among 1 to 5. This value which is near to 1 agrees that the restrained individual might source hitches, though a value near to 5 agrees that the equal distinctive is reserved inside its bounds. Table 1 illustrations the methodical construction of the prototypical data store and software metrics consequence in the data store. The describes the construction data of the graded prototypical system.

Table 1: Representation model

Software Metric	**Key**	**Level No**	**L1**	**L2**	**L3**	**L4**	**L5**
Adaptability	1	1	1	2	1	3	4
Reusability	1	2	2	1	1	2	4
Modularity	1	2	2	1	1	3	4
Understandability	1	1	1	1	2	2	4
Structure complexity	1	2	1	2	3	2	4
Documentation quality	1	2	2	2	2	2	4

The illustration of a five-level prototypical is demonstrated in Table 1. Proceeding the leftward portion of the table, here is a trial metrics prototypical in which the lumps' structure data is deposited in an interactive database through the presentation in the accurate unit of the figure. The classifier construction is definite by essential the standards of the significant, Level No, L1, L2, L3, L4, and L5. Key is the exclusive identifier of the hierarchy and the lumps on or after diverse trees take diverse significant then all lumps of the identical tree take equal key value. The near quantity characterizes the quantity of the near in which the lump is positioned, which is calculated foundation beginning 1 is the origin lump level of the tree.

Here the arrangement quantity of a lump is well-defined as the demand on or after the right to left and beginning highest to lowest displayed in table 1 of this lump in the offspring list of its parental lump. Aimed at the sample, the classification quantity of lump "Understandability" is 2 then "Assembly difficulty" is 1.

When the 5 level is characterized, it likewise mandatory the value of L4 and L5. In detail, the rate of L4 and L5 is established to 5 in the sample displayed in Table 1. The standards of L1 to L5 is kept in the data store of software metrics which resources the software metrics data store sustenance at maximum 5 stages metrics prototypical for in the actual world the metrics representations are usually fewer than six stages.

When the lumps in the prototypical hierarchy can be classified into two categories: inner lump then an outside lump. Outside node consumes not any matter lumps and the inner lump has at minimum one successor's node. The value of individually inner lump can be considered beginning slanted offspring nodes which are related to precise mass standards. By means of aimed at respectively outside node, its value can be attained straight after source code by resources well-defined in metric category table.

Completely metrics must be of an individual of the resulting categories, then separately kind of metrics must be standardizing to harvest standards reaching among one and five conferring to software metrics. The higher boundary metric which is categorized by a discontinuity value and 50% boundary. The data of the accessible metrics is deposited in the metric kind table of the prototypical data store. Organization possibility of metrics mostly contracts through the metrics of the plan possibility such as an entire amount of records or components, roles, instructions, modules, foundation lines of code, and typical resultant classes per class, regular limitations per role etc. The software metrics designated are a group of the maximum general metrics deliberated in the works. The Connection Among Objects "CAO", Deepness of Inheritance Tree "DIT", Deficiency of Consistency in Means "DCM", Quantity of Native Approaches "QNA", Number of Characteristics and Approaches "NCA". Amount of Offspring "AOO", Reply for a Session "RFS", Subjective Technique Amount "STA" using complication as a heaviness for the approaches. Information Concept Connection "ICC", Communication Transitory Connection "CTC", Fitted Session Unity "FSU", Vicinity of Information "VOI", Development of LCOM "DLCOM" and the metrics Outlines of Code "OOC", Deficiency of Certification "DOC", Extent of session terms "ETN". The session possibility of metrics commonly includes the metrics of a session like Weighted Methods per Class "WMC", Complexity of Legacy session etc.

Subsequently, metrics handler chooses the model, in which it is compulsory to compute, the instrument statistically computes the worth on or after the underdone data from data store by means of SQL data store request affording to the requirement of the metrics prototypical tailored by the metrics handler. Since the data comprises comprehensive data of the foundation code, it is acceptable sufficient to originate necessary metrics outcomes after the data. Once the measured information is accessible, the standardized value reaching among 0 to 1 can be considered by the method well-defined in the metrics sort table. In conclusion, the resulting metrics outcome can be kept in the metrics outcome data store for advanced use, whose construction is comparable to the prototypical data.

Current presenting stage, the boundary module loads the metrics information on or after metrics data and offers pictorial demonstration such as a diagram, grid or design to show the metrics outcomes. The standards of a respective lump in the graded prototypical can be shown to metrics handler. At the identical period, the handler can effortlessly modify the edges and defaulting standards of the metrics over the graphics handler interface. The implement grants the outcome from diverse viewpoints to ease package valuation. Figure 2 demonstration the illustration of a metrics prototypical in the implement.

4. Experimental setup and results

Implementation is performed on standard open source software. Altogether procedures remain achieved and investigated on diverse uncluttered foundation tools, on which the transferred source code and the mined information is deposited as well. First, it resolves quantity all recognized software metrics in all present software classifications, or at a minimum, casually choice the established of metrics and examination schemes. Then, the collection is measured by applied special treatment. A comprehensive performance of the assortment of schemes and metrics is providing additional down in this segment. At this time we converse the collection development and the real-world limitations. Intended for this research, 145 open source developments are located nominated randomly separately from the real limitations. In the research, got a huge variation of schemes from different groups in the foundation forge group. Table 2 shows the list of diverse projects.

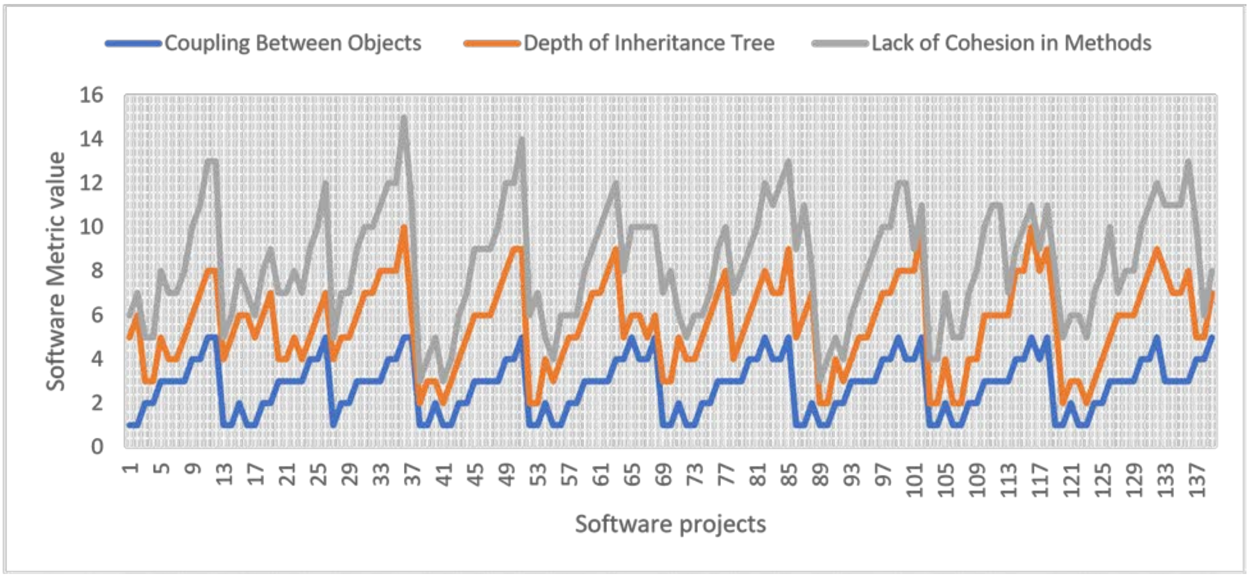

Fig. 2 Metric Evaluation for CBO, DIT, LCOM

This research work performs moderately fine by means of some values clarify mining metadata of twenty diverse assets intended for 546 schemes receipts excerpt in middling 40-100 records. From top to bottom modification appears to originate from changing internet stream of traffic and the foundation server load. Transporting 18 plans with an overall of 253 MB takings SV abstract time of 167 instants, which is nearly the extreme we can grow through our internet connection “1 MB/s”. Estimate and study are achieved afterward the metric intentions are written into the data store designed for arithmetical investigation. Figure 2 to figure 8 demonstrations diverse charts for the metrics involved in the scheme. Respectively figure demonstrations the circulation of classes done the diverse metric standards, i.e., the number of classes through a positive metrics value. The title for respective chart covers the lowest, extreme, typical, middle, and modus standards of a metric and the extreme number of classes for the modus worth over all schemes.

Overall, 24% of the 140 open source software projects included in this research work desired physical solutions in a directive to permit the software metric examination. Covering this study through data about dependences of reference library permitted us to comprehensive the absent dependences in an automatic method. Additional difficulties resolve probably not once be resolvable in an automatic method. All proposals are fundamentally recognized plans connecting shared intelligence, then they are frequently overlooked sense of fittingness or deficiency of time.

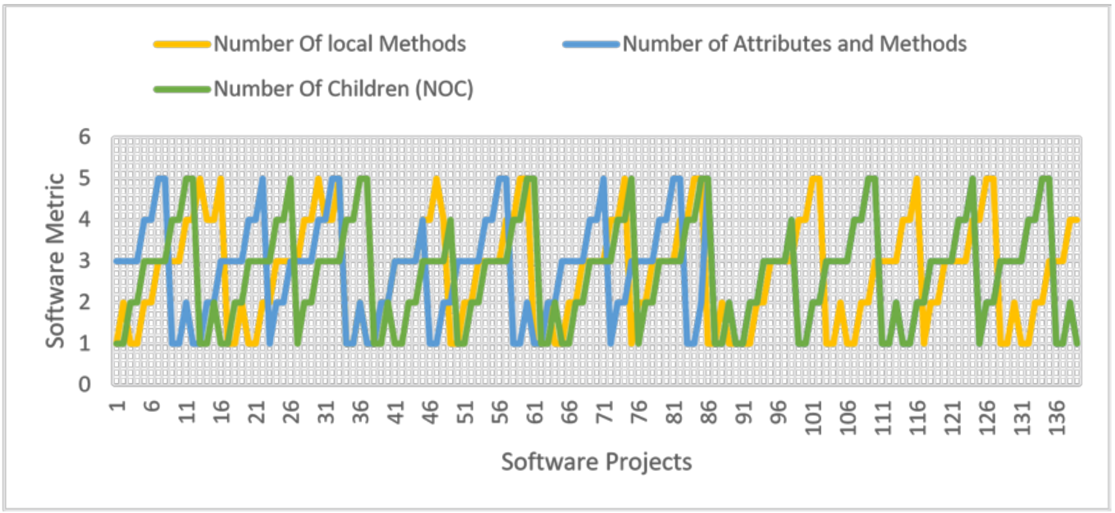

Fig. 3 Metric Evaluation for NLM, NAM, NOC

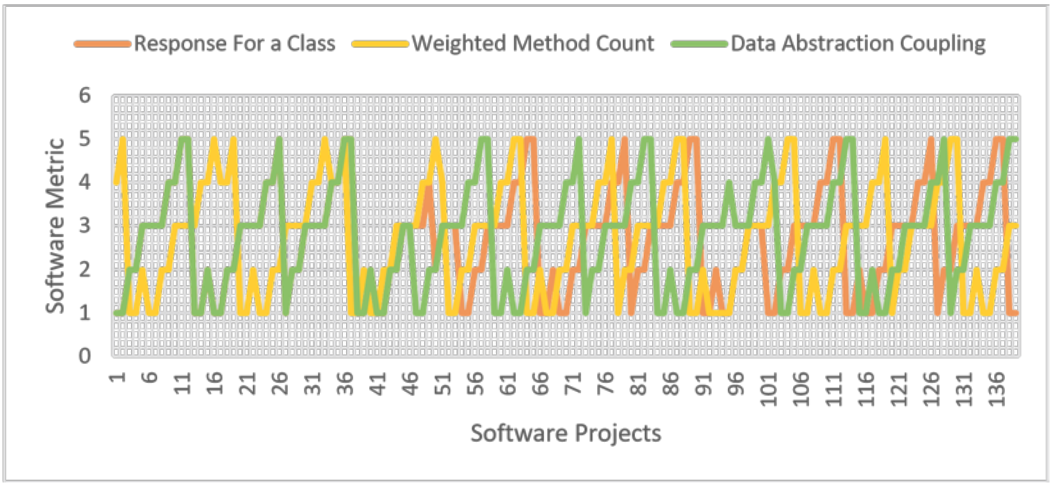

Fig. 4 Metric Evaluation for RFC, WMC, DAC

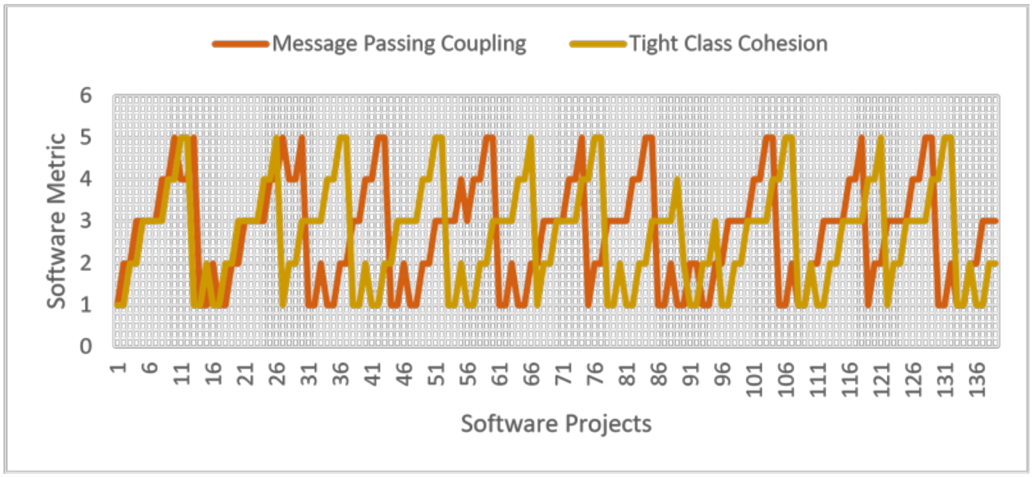

Fig. 5 Metric Evaluation for MPC and TCC

5. Conclusion and future work

Evaluating the open source software quality metrics allows using well-organized metrics grounded software excellence declaration. Vast amounts of computable project information wanted in a directive to attain high numerical implication. As this research work presented in connection with the effort, here it has remained fairly some training focused on the authentication of software excellence metrics. Where all assessable revisions remained partial to insufficient lesser or single large scheme.

In this research work efforts in the direction of refining the effectiveness of authenticating metrics has completed for the open source software tools. The main participants of this investigation effort deliver a tool which provisions for gathering vast amounts of measurable information on open-source software organizations written in diverse software design languages. This study applied the procedure to gather metric information on nearly 140 different schemes with completed 6000 classes and over nine billion lines of program code. This remained thinkable by objective one individual in some limited weeks of part-time effort.

The additional involvement decreases the number of metrics to authenticate for the open source tools. It shows association amongst the distinct metrics, the illustrative that approximately of them look as if to quantity the same belongings. This research training similarly presented the real-world applicability of metric assessment tools for open source software. Lastly, this study work defines metric standards statistically, generous a first impress of the total value series for approximately well-known software metrics ensembles. Investigational assessments of the edges and the handler receipt of this prototypical are upcoming work. An idea to spread the present data store with more schemes. Through this, hope to be intelligent to advance the procedure and tools to permit the completely automatic transfer and examination of projects, requiring no physical communication. To find automatic resolutions for absent dependences and other difficulties recognized. The goal is to experimentally authenticate Software Excellence Representations that are grounded on numerous self-governing, robotically measurable metrics by a presentation their associations to well-accepted excellence qualities, like costs of possession, that remain not automatically expectable.

References

[1] Couto, C., Silva, C., Valente, M. T., Bigonha, R., and Anquetil, N. (2012). Uncovering causal relationships between software metrics and bugs. In European Conference on Software Maintenance and Reengineering (CSMR), pages 223–232.

[2] D'Ambros, M. and Lanza, M. (2010). Distributed and collaborative software evolution analysis with churrasco. Science of Computer Programming, 75(4):276–287.

[3] SourceForge website, Available at http://sourceforge.net. Last accessed December 20, 2010.

[4] Grechanik, M., McMillan, C., DeFerrari, L., Comi, M., Crespi, S., Poshyvanyk, D., Fu, C., Xie, Q., and Ghezzi, C. "An empirical investigation into a large-scale java open source code repository". In ESEM '10: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (New York, NY, USA, 2010), ACM, pp. 1–10

[5] C. Daffara and J. Gonzalez-Barahona. "Flossmetrics Project", 2007. Available at http://www.flossmetrics.org/. Last accessed December 20, 2010.

[6] Haaland K, Groven AK, Regnesentral N, Glott R, Tannenberg A, FreeCode AS. Free/Libre Open Source Quality Models-a comparison between two approaches. 4th FLOS International Workshop on Free/Libre/Open Source Software, pp. 1-17. 2010.

[7] Petrinja E, Nambakam R, Sillitti A. Introducing the OpenSource Maturity Model. In Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, pp. 37-41. IEEE Computer Society, 2009.

[8] V. Yadav, R. Singh, Validating Object Oriented Design Quality using Software Metrics, Proceedings of the International Conference on Advances in Electronics, Electrical and Computer Science Engineering, vol. 2, no. 3, pp. 112-117, 2012

[9] S. George, S. Avinash, J. T. Abraham, Object Oriented Design Metrics, Proceedings of the National Conference on Software Engineering, pp. 283-287, 2014

[10] V. E. Jyothi, S. Kaitepalli, K. N. Rao, Effective Implementation of Agile Practices-Object Oriented Metrics tool to Improve Software Quality, International Journal of Software Engineering and Applications , Vol. 3, no. 4, 2012.

[11] N. S. Bakar, C. V. Boughton, Validation of measurement tools to extract metrics from open source projects, Open Systems (ICOS), IEEE Conference on. IEEE, 2012.

[12] P. Tomas, M. J. Escalona, M. Mejias, Open source tools for measuring the Internal Quality of Java software products. A survey, Computer Standards & Interfaces 36.1 (2013): 244255.

[13] Vukelich, Sdmetrics tool - A tool for measuring objectoriented design metrics from UML models, http://www.sdmetrics.com visited in January 2010.

[14] H. Zhang and S. Kim, "Monitoring software quality evolution for defects," in IEEE Software, vol. 27, no. 4, pp. 58-64, 2010.

M. Tahmassebpour, "Performance Evaluation and Scalability of IP-based and Heuristic-based Job Scheduling Algorithm Backup Systems," Indian Journal of Science and Technology, vol. 9, no. 26, 2016.