A Comparative Study of Maintainability versus Availability Index of Open Source Software

Varun K. L. Srivastava¹, N. Chandra Sekhar Reddy² and Anubha Shrivastava³

¹Association of Education Training and Research Institute (ASSOED), Noida – 201301, Uttar Pradesh, India; mr.varunsrivastava@gmail.com

²Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal – 500043, Hyderabad, Telangana, India; nreddy28208@gmail.com

> ³G. L. Bajaj Institute of Technology and Management, Noida - 201301, Uttar Pradesh, India; anubha.shrivastava@glbitm.org

Abstract

Background/Objectives: The OSS is generally unique in relation to the proprietary programming. In the OSS improvement situation an absolute designer alternately aggregation for developers composes that source book to the primary rendition of the product and makes it uninhibitedly accessible over those webs. At that point other developers would welcome to help the existing code to its next arrival. Settling on the source book of the programming accessible on the web permits developers around the universe to help code, include new functionality, change of the existing source book and submitting bug fixes of the current arrival. Previously, such a programming improvement situation the upkeep of the open hotspot product is a culprit assignment. Creating an OSS framework infers an arrangement from claiming incessant upkeep deliberations to debugging existing purpose what's more including new purpose of the programming framework. The procedure for making those adjustments to product frameworks then afterward their initially arrival may be known as upkeep procedure. The expression maintainability may be nearly identified with that product support. Portability, adaptability and issue tolerance are the opposite essential part that ensures that seamlessness of the programming item. Those generally utilized programming metric to quantify the maintainability is known as Maintainability list (MI). Methods: In this investigation the MI from claiming four A large portion mainstream OSS to be specific Apache, Mozilla Firefox, MySql and FileZilla for fifty progressive discharges might have been observationally investigated. The MI as far as programming measurements in particular Lines about code (LOC), Cyclomatic unpredictability (CC), and Halstead Volume (V) might have been registered for every last one of fifty progressive variants about four OSS. Findings: Proposed agent based approach are cost effective, efficient and accurate solutions to evaluate the software (web application) RMA metrics including the monitoring in cloud computing services that means "Testing/Debugging as a Service Evaluation". Proposed system has performs faster and produce more accurate results to assure the quality of the software related to the non-functional.

Keywords: Agent, Availability, Cloud Computing, Cyclomatic intricacy, Lines Code, Maintenance, Maintainability, Maintainability Index, Reliability, RMA, Software of Open Source, T/DaaS, Volume of Halstead

1. Introduction

In an evolving environment, product is likewise inclined will change. Programming support will be a standout among the significant forms within that product life cycle. The reason for the programming upkeep will be on keep programming operational, to prevent and right faults in the programming and upgrade the purpose of the product. Upkeep alludes of the adjustments aggravated to programming frameworks following it's initially arrival. Modern computational virtual world is no more isolated from the physical world and it has greater interference of human life. It is now the world which connects people to work together for achieving the target in all aspect of the life. Exponential growth of the this computational world is everywhere allowing the people to search and use seamlessly, on demand with mobility having faster speed and so on. This popularization has gain attention due to surrounding and parallel development and advancement all areas. Computer has size has shrinking every day whereas its power has increases exponentially. The open source technology community has also contributing to enrich this field to be grown on. Advent of the android devices cannot be ignored.

Then again diverse clients have distinctive prerequisites about computational energy and requisition and frameworks programming. Henceforth interest for clients will be heterogeneous done nature thus that varieties of provision (hardware and programming) bring been created to attain the most astounding client fulfillment. Headway for hardware and telecommunication field need finished that employment. Specialization need more guaranteeing over generalization because of finesse for particular job/function be that it additionally need dim sides. Hence the qualitative software is required that can meet the users functional requirements fully without having the bugs on it assuring the availability of the services, flexible enough to modifications, easy to maintain or ease maintenance and reliable also.

Software engineering is an art science of developing the qualitative software products that can fulfill the users/ stakeholders requirement cost effectively.

Software engineering has many phases to developed the product like feasibility analysis and requirements gathering started with the customer communications, analysis (collecting all the required data and meta data, defining new data types and functions and systems), design the product, implementation, testing and maintenance insurability before delivering the product to the costumers. SQA (Software Quality Assurance) is the way to monitor the software from entry point to the exit phase of the software product. Three major metrics are biggest challenges to be persists in the software product i.e., RMA (Reliability, Maintainability and Availability). Also modern product possesses the inherent portability to be fit in any architecture device and platform running on.

Secondly, Different prerequisites oblige various specific gadgets (CPU, capacity and so forth throughout this way, observing and stock arrangement of all instrumentation may be enha.) and product instruments. Buying or permitting for every last bit such required things (devices and applications) may be not practical of the association or people as far as those cosset and establishment. Furthermore the majority of the assets need aid unmoving pulley i.e., habitually not utilized. Consequently the utility sorts for registering standard will assume an import part.

Cloud registering will be another registering standard in light of utility registering model which will satisfy those user's prerequisite rapidly for lease foundation.

1.1 Objective

- To Evaluate RMA of the software SaaS application in context of the PaaS Cloud and their availability of the application and resources (network, CPU, Memory and I/O) in cloud computing paradigm especially in public cloud.
- Deploying a web services under SaaS paradigm and evaluate the reliability, scalability and effectiveness of the application in the cloud environment as Testing/ Debugging as a Service.
- Evaluation of Public Cloud SaaS and PaaS and their interoperability of the agent (java agent) developed on New Relic service integrating.
- Integrating the software agent onto the SaaS to analyze the RMA metrics do the real time monitoring of the same using distributed agent concept.

2. Survey of Literature

As stated the maintainability list may be speaking to concerning illustration An work of normal Halstead Volume (V) for every module, normal cyclomatic intricacy for every module, normal lines about code for every module and normal percent of lines of code for every module¹.

As stated by that mi might a chance to be utilize likewise and pointer of maintainability to article situated frameworks².

In³ examined that to ascertaining the maintainability list different models namely, a three- metric model In light of (Fan-Out, information unpredictability Also McCabe Cyclomatic Complexity), an single-metric model in view of Halstead's exertion and a four-metric model in view of (Halstead's effort, McCabe cyclomatic complexity, lines of code also percent about comments) can be additionally make utilized.

In⁴ suggested a maintainability model that sorted product modules similarly as "easy to maintain" Also "not simple will maintain". Such classification might help on identify the modules, which need aid not simple to keep up.

In⁵ recommended a programming maintainability hierarchy, as far as portion maintainability indicators. As stated by hierarchy, Halstead intricacy and Cyclomatic intricacy are the indicators for maintainability.

As stated by, maintainability will be an inward nature quality which camwood a chance to be communicated likewise a capacity about modularity, portability, readability, testability, reusability, adaptability furthermore congruity.

As stated by Welker the item situated frameworks bring a reasonably secondary mi because of the average more modest module extent. In as much see more modest modules hold numerous lesquerella operators furthermore operands, lesquerella executable paths, also fewer lines for remarks also code in this way the mi has a tendency should make higher⁶. Writer² need surveyed a number scientific categorization of the programming improvement with customary of the present day part based including those the vast majority testing space for example, such that constant programming.

As stated by writer^Z: Non-functional worries for example, occasion when and space predictability, dependability, safety, and All the more as of late security, have an progressively vast frequency with respect to framework advancement clinched alongside high-integrity requisition domains for example, flight Furthermore space, railways, telecom and, prospectively, car.

2.2 Software Development using Coat Hanger Model

According to⁸, In the focal point for author's suggested methodology may be those act for programming development, which may be concerned for making portrayals of the purposes of the software, for its issue domain, about its structure and behavior, of the computations should make performed, of the interfaces between those product and its earth and its users. Product improvement includes various hones including the individuals about analysis, design, execution & caliber management.

2.3 Software Cost Estimation

Writer² need recognized that issue of exact cosset estimation of the programming for up to date advancement nature's domain. As stated by writer: Estimating the expense of a programming one task may be an undertaking from claiming key criticalness previously, venture administration. Both again and underestimations of expense camwood result in genuine issues will an organization. To instance, overestimations might bring about an organization loosing contracts or wasting resources, while underestimations might bring about poor quality, deferred or unfinished product frameworks.

2.3.1 Agile²

The "home ground" for alarmed programming advancement polishes is little groups of exceedingly encountered kin building small, non-critical frameworks for exceedingly unstable prerequisites over an association for acknowledgement from claiming vulnerability. However, alarmed improvement may be progressively received over extensive associations running enormous programming improvement activities utilizing various groups dispersed on a few geological areas. Adopting alarmed routines in this connection introduces new tests identified with scaling, for example, such that inter-team coordination, viable information offering between the teams, outline without An. defined architecture or properly defined requirements, as well as all the challenges of distributed projects.

2.3.2 Maintainability

The main agenda of this research is to test and proposed the software maintainability and availability. In^{10} has explored the concept of the maintainability of the software and proposed a fuzzy based method to evaluate the maintainability of the software.

Maintainability chose by outline will be a noteworthy trademark about result that could make support convenient, quick also prudent. For the conclusion of result design, maintainability gets to be that inalienable item quality. Maintainability assessment will be a paramount manner to assess item maintainability. Previously, late years, a number researcher's need connected those various criteria choice making principle in the field dependent upon mulling over maintainability influencing attributes, and procured useful comes about.

2.4 Methodology

To creating suggested agenize based framework four sorts of general population cloud could a chance to be decided and their benefits need been chosen concerning illustration proving ground to preferred assessment and estimation of the precision of the recommend framework. At long last those code need been assessed and tried utilizing at any point information cloud administration supplier. They need aid taking after with particular purpose in the suggested system-.

- 1. Codenvy- will create and/are requisition i.e., SaaS (Software Likewise a service). To those recommended framework a java web requisition utilizing jsp (Java server pages) provision need been decided to create looking into of the codevny SaaS cloud administration supplier.
- 2. Cloudbees- on send and test our SaaS requisition onto those cloud, recommend framework necessities a stage i.e., Stage Concerning illustration a administration (PaaS). To this cloud bees administration supplier need been coordinated circuit onto the created SaaS provision.
- 3. Everdata-Jelastic, inc., those primary agency that joined together boundless PaaS (Platform-as-a-Service) also container-based IaaS (Infrastructure-as-a-Service) under you quit offering on that one solution, need published their organization for EVERDATA, An heading datacenter also facilitating administration supplier built in Jaipur, india. Gartner as of late accounted that open cloud administrations income over india will scope \$638 million by those end about 2014, a build about very nearly 34 percent, alternately \$161 million again 2013 income. State funded cloud administrations income will be determined from claiming Growth done magic market segments, in particular cloud framework as a administration (IaaS) and programming as a administration (SaaS), two of the biggest segments in the open cloud benefits showcase. Using on SaaS will downright \$249 million done 2014, an expansion from claiming 37 percent in a year ago. Using with respect to IaaS will develop 33 percent on \$77 million throughoutthat same period.
- 4. New relic- on create those center purpose of the recommended framework, i.e., checking also planning utilizing product agenize new relic administration need been subscribed. In this java agenize need been altered to help those observing and planning of the SaaS administrations.

5. Issue space and recommended instrument. Recommend framework need surveyed and distinguished the issue area that must a chance to be tended to done connection of the cloud registering Also Thus available the clue about agenize reconciliation. These would follow. 1. Quickness (speed up). Exact and constant estimation of dependability. 3. Impact for maintainability throughout delivering the benefits ahead of the assets. 4. Assessment from claiming accessibility measurements of the programming item looking into dainty alternately thick gadgets.

Same time creator¹¹ and^{1,12–16} suggested a cloud based answer for unravel the over recorded QoS parameter that extraordinarily influence the execution from claiming cloud administration particularly SaaS.

Be that the fundamental issue same time searching will be the disseminated assessment substance i.e., Selfsufficient code Hosting portability characteristic with assess the measurements as the administration conveyed from those cloud. The fundamental needing purpose in the article may be acceptance of the recommended system.

Point of interest objective of the suggested agenize built T/DaaS administration.

Taking after objectives need been attained alternately tackled with coordinating of the portable agenize will cloud registering administration realization.

To assess and conveyed the cloud registering administrations (SaaS) utilizing agenize (for finer Also quick delivery) utilizing government funded cloud for example, such that "New relic Also at any point data".

Deploying a web administrations under SaaS and empowering T/DaaS on it standard and assess the adequacy of the web provision clinched alongside connection for RMA measurements of the programming in the cloud surroundings with those help about agenize. To SaaS advancement ever data has been subscribed. Done which JSP built provision need been creating also deployed.

Assessment Also general population PaaS (platform as An Service) of the ever data elastic cloud should assess those maintainability of the programming item administration coordinating an SaaS sending on it

Furthermore delivering through agenize. Measuring those execution of the suggested explanatory approach (influenced from Aneka) clinched alongside cloud administrations for example, general population cloud like ever information.

3. Results and Analysis

3.1 Results Obtained using Proposed Agent based Approach

• Apdex Score of the deployed SaaS- Figure 1 shows the apdex score of the SaaS application has been deployed onto the Cloudbees open cloud service provider. Generally Apdex score is an industry standard to measure users satisfaction with the response time of an application or service. It's a simplified Service Level Agreement (SLA) solution that gives application owners better insight into how satisfied users are, in contrast to traditional metrics like average response time, which can be skewed by a few very long responses. Table 1 shows the obApdex score of the deployed SaaS and has been monitor and provisioned by customized agents –

- Average Loading Time- Figure 2 shows the average loading, reposnse CPU usages (during request) of the deployed SaaS application. These statistics has been monitered and collected by various agents mentioned in the chapter 4. Table 1 has shown the average load time of SaaS
- Average Response Time- Figure and Table 2 shows the average response time of SaaS from cloud server using agent. This grapph also tells the time spent of the SaaS in JVM (Java Virtual Memory) of application while requesting and in between response, as shown in small rectangle box.



Figure 1. Average loading of the SaaS monitered using Agent.



Figure 2. Average response time of the SaaS monitered using agent.

Table	1.	Average	response	time.
-------	----	---------	----------	-------

	Average Response Time (in millisecond)
Minimum	6.4 ms
Maximum	98 ms

Table 2. Average load time

	Average Load Time (Maximum)
Desktop (Yellow Line)	1.4 sec
App Server (Blue Line)	4.41 sec

4. Conclusion and Future Work

Cloud registering makes a virtual standard to offering information and computations again a through web alternately organize with those help about hosts or hubs were hosting adaptability. Set for such hubs (scalable) i.e., organize known as cloud. And the deployed provision around such versatile organize is known as cloud requisition. Cloud registering may be up to date to integrations of machine and system innovations for example, such that quick microprocessor, gigantic memory, high-sounding system and dependable framework building design.

In a far-reaching way the administration about cloud need been isolated under three significant classes: SaaS (Software-as-a-Service), PaaS (Platform-as-a-Service), and IaaS (Infrastructure-as-a-Service).

5. References

- Coleman D. Assessing maintainability. Proceedings of the 1992 Software Engineering Productivity Conference; 1992. p. 525-532.
- Coleman D, Ash D, Lowther B, Oman P. Using metrics to evaluate software system maintainability. Computer. 1994; 27(8):44–9. https://doi.org/10.1109/2.303623
- Hayes JH, Zaho L. Maintainability prediction a regression analysis of measures of evolving systems. Proceedings of 21st IEEE International Conference on Software Maintenance; 2005. p. 601–4. https://doi.org/10.1109/ICSM.2005.59
- Hsia P, Gupta A, Kung C, Peng J, Liu S. A study on the effect of architecture on maintainability of object-oriented systems. IEEE Conference on Software Maintenance; 1995. p. 4–11. https://doi.org/10.1109/ICSM.1995.526522

- 5. Glossary of software engineering terminology. https:// www.researchgate.net/publication/224001232_Glossary_ of_Software_Engineering_Terminology_IEEE_ Standard_61012
- 6. Software quality: Concepts and practice. https://www. wiley.com/en-us/Software+Quality%3A+Concepts+and+ Practice-p-9781119134497
- Misra SC. Modeling design/coding factors that drive maintainability of software system. Software Quality Journal. 2005; 13(3):297–20. https://doi.org/10.1007/s11219-005-1754-7
- Oman P, Hagemeister J. Construction and testing of polynomials predicting software maintainability. Journal of System Software. 1994; 24(3):251–266. https://doi. org/10.1016/0164-1212(94)90067-1
- Oman P, Hagemeister J. Metrics for assessing a software system's maintainability. Proceedings of the Conference on Software Maintenance; 1992. p. 337–44. https://doi. org/10.1109/ICSM.1992.242525
- Tahvildari L, Gregory R, Kontogiannis K. An approach for measuring software evolution using source code features. Proceedings of the Sixth Asia Pacific Software Engineering Conference; 1999. p. 10–17. https://doi.org/10.1109/APSEC.1999.809579
- 11. The software maintainability index revisited. https:// www.researchgate.net/publication/255944836_Software_ Maintainability_Index_Revisited
- Panunzio M, Vardanega T. A component-based process with separation of concerns for the development of embedded real-time software systems. Journal of Systems and Software. 2014; 96:105–21. https://doi.org/10.1016/j. jss.2014.05.076
- Päivärinta T, Smolander K. Theorizing about software development practices. Science of Computer Programming. 2015; 101:124–35. https://doi.org/10.1016/j. scico.2014.11.012
- Minku LL, Yao X. Ensembles and locality: Insight on improving software effort estimation. Information and Software Technology. 2013; 55(8):1512–28. https://doi. org/10.1016/j.infsof.2012.09.012
- PaasivaaraM, Lassenius C. Communities of practice in a large distributed agile software development organizationcase Ericsson. Information and Software Technology. 2014; 56(12):1556–77. https://doi.org/10.1016/j.infsof.2014.06.008
- Zhong L, Youchao S. Research on maintainability evaluation model based on fuzzy theory. Chinese Journal of Aeronautics. 2006; 20(5):402–7. https://doi.org/10.1016/ S1000-9361(07)60061-2