© 2019 IJSRST | Volume 6 | Issue 3 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Manage And Inhance Software Quality to Improve Performance Through User Development

Nirvikar Sharan Katiyar[1], Dr. Pradeep Mittal[2]
[1]Ph.D Scholar, Kurukshetra University Kurukshetra, Kurukshetra, Haryana, India
[2]Kurukshetra University Kurukshetra, Kurukshetra, Haryana, India

ABSTRACT

This essay presents a tutorial that discusses software quality in the context of total quality management (TQM). Beginning with a historical perspective of software engineering, the tutorial examines the definition of software quality and discusses TQM as a management philosophy along with its key elements: customer focus, process improvement, the human side of quality, and data, measurement, and analysis. It then focuses on the software-development specifics and the advancements made on many fronts that are related to each of the TQM elements. In conclusion, key directions for software quality improvements are summarized. Software metrics have a direct link with measurement in software engineering. Correct measurement is the prior condition in any engineering fields, and software engineering may be not an exemption, as those size and complicated nature of software increases, manual examination of software becomes a harder assignment. Most Software Engineers worry about the quality of software, how to measure and enhance its quality. The overall objective of this study was to asses and analysis software metrics used to measure the software product and process. In this Study, the researcher used a collection of literatures from various electronic databases, available since 2008 to understand and know the software metrics. Finally, in this study, the researcher has been identified software quality will be a method for measuring how software is designed and how well the software conforms to that configuration. A percentage of the variables that we would be searching for software superiority and Correctness, item quality, Scalability, completeness and absence of bugs of those quality standard that might have been utilized from you quit offering on that one association will be unique in relation to others for this reason it may be better to apply the software measurements to measure the quality of software and the current is most common software metrics tools to decrease the partiality of faults during the valuation of software quality. The central influence of this study is an indication around software metrics to illustrate for development in this field by critical investigation about key metrics initiated on both developer and user interaction a unified definition of software quality management on User and Developer (SQMUD) is proposed.

Keywords : Quality of software, Literature review, Existing work, Proposed work, software testing and faults, Entropy, Total quality management.

I. INTRODUCTION

The total quality management (TQM) concept represents a fundamental change in the definition and treatment of quality in product development. Traditionally since the beginning of the industrial revolution, US industries had a product-focused mentality. Correct measurement is the prior condition in several engineering fields, and software engineering is not an exemption. Software metrics

require a direct link with measurement in software engineering. Software metrics will reduce the subjectivity of faults during the assessment of software quality and it provides a measurable foundation for creating choices around the software quality. Metrics are the numerical value of software and it is used to predict the fault [3]. Software metrics occur file level, class-level, component-level, method-level, process-level and quantitative values-level metrics [4], this helps the project manager and software engineers to find defects and making the prevention method for the defect. Software metrics can be applied to each software development phase. During requirement analysis software metrics can be developed, for instance, in order to determine cost estimation and resource needed. At the time of system designing we also develop metrics in order to count function point. Metrics applied at implementation phase are also used to measure software size [5]. According to Vikas Verma , having software metrics have a number of benefits such as provide a foundation for approximation and simplifies preparation by means of controlling status reporting, identifying risk areas and effectiveness and efficiency of testing[23]. Measuring the software project has a number of benefits for company it saves development effort, time and money. In addition to this for complex projects using metrics have easy to understand, identify common problems early, and manage resources [7].As mentioned above even if it has the benefit there is also drawbacks that are better understanding (knowledge) and need a lot of effort and time. Software metrics empower software developers to investigate their code and make upgrades assuming that required. Metrics could be developed for software size, cost estimation, software quality, maintainability, deformity analysis and software testing [5].

II. LITERATURE REVIEW

Many software development enterprises are using total quality management (TQM) methodology to improve developmental efficiencies and enhance the quality of software. How effective are these TQM programs in improving software quality? To answer this question, this paper examines the relationships between TQM implementation factors and several measures of software quality, including user satisfaction, the Capability Maturity Model level, and the extent of compliance with ISO 9000-3 guidelines. Data gathered from 247 software development sites show that, in general, there is a positive relationship between the TQM implementation factors and the indicators of software quality. However, the strengths of these relationships vary according to the factors considered.The first survey on software metrics was done by Kafura in 1985 and he suggests existing code metrics, complexity metrics and validation metrics. Generally, in this survey work presented the major relations exist among the software metrics and quality aspects like comprehensibility of code, error features, length of coding time, and structural soundness [28]. According to Ming Chang et al [29] discussed the role of software metrics and software measurement for software quality. Authors also classified the software metrics according to various manners which are commercial, important, observation, measurement and software development. In addition to this, the author also discussed various methodologies which are around 15 measurement methodologies and 24 types of testing with their definitions, formula and effects. Poornima Gupta et al [30] presented the software fault prediction using artificial intelligence methods and this research work focused on related work on software metrics particularly on AI approaches and software metrics. Kunal Chopra et al [31] discussed about software metrics complexity using N depend to measure

software product like size metrics, control flow metrics and data flow metrics. The final contribution by researcher was introducing the most commonly known and utilized software metrics projected and evaluate the use of software metrics in creating simulations of software expansion procedure.

III. EXISTING WORK

We identify top management leadership, a sophisticated management infrastructure, process management efficacy, and stakeholder participation as important elements of a quality-oriented organizational system for software development. A model interrelating these constructs and quality performance is proposed. Data collected through a national survey of IS executives in Fortune 1000 companies and government agencies was used to test the model using a Partial Least Squares analysis methodology. Our results suggest that software quality goals are best attained when top management creates a management infrastructure that promotes improvements in process design and encourages stakeholders to evolve the design of the development processes. Our results also suggest that all elements of the organizational system need to be developed in order to attain quality goals and that piecemeal adoption of select quality management practices are unlikely to be effective. Implications of this research for IS theory and practice are discussed.

IV. PROPOSED WORK

The proposed study involves an improved version of AZ-Model after obtaining the opinion of experts and obtaining expertise after proper implementation for various sizes of projects in organizations with different sizes. ・ Furthermore, statistical analyses are performed to examine the significance of AZ-Model.

V. IMPLEMENTATION

1) Productivity and quality are conflicting goals. Improving quality consumes additional corporate resources that are needed to maintain productivity. Therefore, quality can be improved only at the expense of productivity.

2) Quality is defined as conformance to specifications or standards. Such conformance pays no attention to incorrect specifications or obsolete standards that are prevailed in most companies.

3) Quality is measured by degree of nonconformance. It is usually measured by the defect count in "parts per million"— the famous six-sigma measurement. Such measurement focuses on the degree of non-conformance in stead of customer satisfaction.

4) Quality is achieved through intense product inspection. Such inspection consumes much of the corporate resources. If a product fails the inspection, it needs to be reworked or scraped.

5) Some defects are allowed if a product meets minimum quality standards. This implies that customers are willing to pay for a “buggy” yet working product.

6) Quality is a separate function and focused on evaluating production. It is assumed that the production group will welcome such independent quality function.

7) Workers are blamed for poor quality. However, replacing a worker does not mean improving quality. Furthermore, poor quality may come from the supplier side.

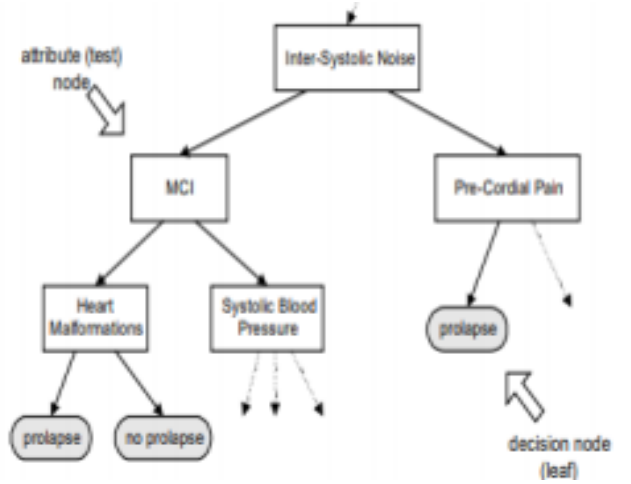
8) Supplier relationships are short-term and cost-oriented. There is no way to control the quality of raw materials or parts delivered by the suppliers.

The SQMUD May be a screening procedure which may be used to guarantee the quality on whole software development lifecycle methodology. It will be a nonstop evaluation system which facilitates specific methods for task development with

particular guidelines alongside documentation. The methods if be used to guarantee personal satisfaction Conclusion (zero defects) Also venture victory. Toward a secondary level, the capacity from claiming SQMUD may be with performs those following:

• Software undertaking planning: nature polishes ought to a chance to be arranged ahead of time which can wood make actualized further.

- Client requirement: necessities ought to further bolstering a chance to be checked done whole task development transform to fulfill the client necessities.
- Plan procedure & Coding: sure methodologies need aid emulated for configuration transform. Coding standard and rules must a chance to be made and actualized.
- Software coordination and Testing: product joining and trying if be arranged Also aggregated Likewise for every prerequisite.
- Direct irregular and planned audits: perform SQMUD audits to guarantee those fundamental controls are set up. The SQMUD procedure comprises of a mixed bag for stages for particular exercises. These exercises ought further bolstering make performed Toward a SQMUD less group which is answerable for product quality certification planning, analysis, And reporting weight. SQMUD is more successful the point when it reports dependent upon through An differentiate administration less group thereabouts they can stay dedicated of the procedure And remain objective of the deliverable. Those responsibilities of the SQMUD less group incorporate survey from claiming documentation to culmination and adherence to standards, support Previously, inspections, Audit for test results, Also occasional audits for controls. Past exploration fill in the suggested examine includes a moved forward rendition about. AZ-Model then afterward acquiring that assessment for masters and acquiring smoothness after best possible usage for Different sizes for undertakings done associations with diverse sizes. And, measurable analyses were performed should. Analyze the noteworthiness of the recommended AZ-Model. Recommended research worth of effort. Those suggested study includes a moved forward versify from claiming. AZ-Model after getting that assessment for masters and acquiring dexterity then afterward fitting execution for Different sizes about undertakings clinched alongside associations with distinctive sizes. And, decision tree analyses were performed on. Inspect the noteworthiness of the suggested AZ-Model. Product measurements and dependability product development will be an intricate Also confounded methodology for which product faults would embedded under those code toward mistakes throughout the advancement methodology or upkeep. It need been indicated that those example of the fault's insertion phenomena is identified with measurable qualities of the software, particularly for the product measurements. For example, an expansive software framework comprises for Different modules and every about these modules might make described as far as quality measures – it might remain calm helpful should have the ability with build ―dangerous module‖ prediction models in view of these measurable qualities.



VI. ENTROPY

The intuitive and web-based Entropy® Software implements rapidly so you make performance improvements faster. No more paper forms, juggling spreadsheets or inconsistent data from your various sites. Instant and consistent data for corporate, regional, and individual sites is readily and easily available for authorized managers anywhere in your organization. With its incredible flexibility you get the score carding and reports you want to manage risks, take action, and improve your bottom line.

VII. TOTAL QUALITY MANAGEMENT

The Definition OF TQM According to the Webster's Dictionary, "quality" is "a degree of excellence; a distinguishing attribute." That is, quality is the degree to which a product lives up to its performance, endurance, maintainability, and other attributes that a customer expects to receive from purchasing this product. In order to produce quality product, one must instill TQM concept into one's product development process. The word "total" means the total of everything in an organization. That is, it covers every process, every job, every resource, every output, every person, every time and every place. According to the American Society for Quality Control (ASQC), total quality management (TQM) "is a management approach to long-term success through customer satisfaction. TQM is based on the participation of all members of an organization to improving processes, products, services, and the culture they work in. TQM benefits all organization members and society. The methods for implementing this approach are found in the teachings of such quality leaders as Philip B. Crosby, W. Edwards Deming, Armand V. Feigenbaum, Kaoru Ishikawa, and J.M. Juran." [Bemowski, 1992] The Department of Defense also provides a concise and accurate definition as below: "TQM is both a philosophy and a set of guiding principles that represent the foundation for a continuously improving organization. TQM is the application of quantitative methods and human resources to improve the material and services supplied to an organization, and the degree to which the needs of the customers are met, now and in the future. TQM integrates fundamental management techniques, existing improvement efforts, and technical tools under a disciplined approach focused on continuous improvement." [Department of Defense, 1991] TQM View of Quality under the TQM concept, quality is defined and judged by the customer. Therefore, it acknowledges a customer-driven economy. It focuses on continuous process improvement to achieve high quality of product (or service). Its strategy tries to achieve "total quality" throughout the entire business, not just in the product. It suggests that any improvement that is made in the business, be it a better design of a component or a better process of a system, will help to improve the "total quality" of the organization and the quality of the final product. Under this philosophy, the view of quality is very different from the traditional one: 1) Productivity and quality are not conflicting goals. Productivity gains can be achieved through quality improvements. Better quality of product and process will reduce

rework, errors, and waste. This, in turn, improves the productivity. 2) Quality is correctly defined requirements that satisfy users needs. The ultimate quality of a product is its ability to satisfy user's needs. One should take one step further to get the consumer involved in defining the product requirements. It is plausible to say that quality is defined and judged by the customer. 3) Quality is measured by user satisfaction as well as by continuous process and product improvement. Just as one would expect, customers prefer to purchase software that fits their needs and performs beyond the quality standards. The TQM practice shuns the old adage of "don't fix it if it ain't broke." 4) Quality is achieved by effective product design and process controls. Relying on product inspection implies that errors will definitely be made. Quality cannot be achieved by inspection. It should be built in, not added on. To build in quality, one must perform effective product design and process controls. 5) Defects are prevented through process control techniques. Zero defect and perfection of processes should be the goals if a company wishes to keep improving quality. The Journal of Quality Assurance Institute, Vol. 14, No. 1, January 2000, pp. 4-6 & 35-41. 3 6) Quality is a part of every function in all phases of the product life cycle. It simply does not make sense to go about production haphazardly or without a quality-laden plan and expect a quality good or service and a happy customer for that matter. 7) Management is responsible for quality. Only management has the authority to change the working conditions and processes, and only management has the knowledge to coordinate quality function in all phases of the product life cycle. Therefore, management should be responsible for quality, not the workers. 8) Supplier relationships are long-term and quality-oriented. Suppliers are just as an important part of the team as any other members. Since management is responsible for quality, it must also take charge of building long-term and quality-oriented relationships with suppliers. 3. Principles of TQM Implementation Based on the lessons learned by various companies implementing TQM as reported in the literature, there are some principles of practices that are instrumental to the success of TQM implementation [Department of Defense, 1990; Moore, 1990]: 1) Quality is everyone's business. Quality is the concern of not only the management but also the workers. By empowerment, that is empowering employees with the ability to stop production if quality is sacrificed, quality can be dramatically improved. Workers feel a sense of belonging to the process and a pride in the quality of their work. Quality is perceived as a team effort. Auto factories are now empowering their employees with the ability to stop the whole production line if someone discovers a quality defect. 2) Customer Emphasis. One must focus on satisfying internal and external needs, requirements, and expectations, not just on meeting specifications. This is essentially creating a customer focus. Since customers are the ones that drive production, their needs and expectations should be the focus of all improvement efforts. Customers are not only those who buy finished products. There are also workers within the company who use the components produced by other workers. These workers are internal customers. A production line can be conceived as having a string of customers, starting with the one who is making the contract. Each person is responsible for improving the quality of the product that they pass on to the next customer. Under the TQM culture, everyone has a customer. 3) Quality must be built into the product. Quality cannot be an afterthought. It must be constantly measured and quantified. The question: "Is this good quality?" must be a centerpiece in any development project. It must be a concern from beginning to end. Quality is not determined by picking the best of the bunch after production and recycling the bad ones.

"Bad ones" should never exist in a TQM environment. Defects should be discovered before any production occurs. This is accomplished by building quality into the product. It is easiest to understand this concept by thinking about quality as a part of a product. The product cannot work without the quality component installed. Therefore, during every stage of the development process, the developer must ask himself: "Have I installed the quality component in this product?" 4) TQM requires management commitment and involvement at all levels. TQM must be implemented from the top down in every organization. If management does not have a commitment to a TQM culture, it will fail. The management must provide leadership in implementing the change; the workers do not have the power to do so. Do not blame the workers for poor quality; the management and the systems are responsible for quality. 5) TQM accomplishment involves continual training. Continuous improvement includes the improvement of one's ability in performing one's job. An employee must be trained in TQM principles and in the tools and techniques for implementing TQM. Such training credential should be treated as an accomplishment for performance evaluation. 6) Leadership is substituted for slogans and exhortations. We have all heard the slogan: "Quality is job one." This example is from Ford Motor Company. However, slogan means nothing when we say it but we do not do it. Ford still produces buggy cars despite its great success in Team Taurus project [Walton, 1986]. What Ford needs is better leadership in quality improvement. 7) Long-term emphasis on measurable processes and productivity improvement. TQM cannot be implemented overnight. It is a long-term process that takes years to implement. It is a complete cultural change in the organization to focus on continuous improvement. The problem with achieving continuous improvement is that it requires measures to be compared against. Therefore, a The Journal of Quality Assurance Institute, Vol. 14, No. 1, January 2000, pp. 4-6 & 35-41. 4 key element of the TQM culture is qualified metrics–measurements taken continuously in order to chart progress. 8) Understand the current process before improvement begins. We must understand how things work in the organization to be able to improve it. Understanding how it works involves being able to measure the process in order to compare "improvements" against it. 9) Cross-functional orientation and teamwork. The essence of cross-functional teams is to integrate many different parts of the organization into the development process. For instance, programmers must involve users from finance, accounting, marketing, and other departments in the development of a software product. This philosophy is developed with the thought that developing a product is not just the designer's concern. Everyone who is involved with the development, distribution, and maintenance of a product should have a say in the development of the product. 10) Effective use of statistical methods and quality control tools. Statistical quality control and process control techniques should be used to identify special causes of variation that are Table 1: Quality Control Tools Quality Seven (Q7) Tools Management Seven (M7) Tools 1. Cause-and-effect diagram. This diagram is also called "fishbone diagram" or "Ishikawa diagram." It identifies, explores, and displays all possible causes or contributing factors of a problem or event. 2. Checksheet. This document is designed to tabulate the results through routine checking of the situations. It is passed between major checkpoints during the production process and acts as a safeguard from defects. 3. Control chart. This chart serves to detect special causes of variation. The chart has control limit lines at the center, top, and bottom levels. Sample data are plotted in dots on the chart to evaluate process situations and trends. 4. Histogram. This

diagram graphically displays a set of frequency data in bar graphs and enables evaluators to determine problems by checking the dispersion shapes, center values, and the nature of dispersion. 5. Graphs. There are many types of graphs that are useful to evaluators. Line graphs, also called run charts, are used to illustrate variations over a period of time. Bar graphs compare categorical values via parallel bars. Circle graphs, or pie charts, indicate the categorical breakdown of values relative to the total value. Radar charts assist in analyzing previously evaluated items each may have its own axis of measurement. 6. Pareto chart. This chart classifies problems according to cause and phenomenon. It makes use of bar graphs sorted in a prescribed order to display the relative importance of problems by selected categories. 7. Scatter diagram. This diagram is also known as X-Y chart. It displays what happens to one variable when another variable changes in order to test a theory or make forecasts. 1. Affinity diagram. This is essentially a brain storming output. It is based on group work in which every participant writes down his ideas and the ideas are then grouped and realigned by subject matter. 2. Arrow diagram. This diagram is also called "flow chart." It is often used in PERT (Program Evaluation and Review Techniques) and CPM (Critical Path Method). It uses a network representation to show the steps necessary to implement a plan or to complete a process. 3. Matrix diagram. This diagram is used to clarify the relations between two different factors. It is often used in deploying quality requirements into counterpart (engineering) characteristics and then into production requirements. 4. Matrix data analysis diagram. This diagram is used when the matrix diagram does not provide sufficiently detailed information. It is the only M7 tool that is based on data analysis and gives numerical results. 5. Process Decision Program Chart (PDPC). This diagram is commonly used in operations research community. It is a hierarchical chart that displays how an optimal alternative is arrived. It is similar to a decision tree diagram. 6. Relations diagram. This diagram is also known as causal model diagram. It clarifies the interrelations in a complex situation involving many interrelated factors and serves to clarify the cause-and-effect relationships among factors. 7. Tree diagram. This diagram is similar to a functional decomposition chart. It is applied to show the interrelations among goals and measures and among processes and activities. Sources: Adapted from Ishikawa [1976], Imai [1986], Brassard [1989], and Walton [1986]. The Journal of Quality Assurance Institute, Vol. 14, No. 1, January 2000, pp. 4-6 & 35-41. 5 points outside the control limits. Actions should be taken to remove these special causes. Moreover, any abrupt shifts or distinct trends within limits are also signals for investigation. Quality control tools such as the Quality Seven (Q7) tools and the Management Seven (M7) tools [Brassard, 1989; Imai, 1986; Ishikawa, 1976] may be used to plan for actions, collect valuable data, and chart for progress. Table 1 lists the names of these tools and their descriptions while Figures 1 and 2 display their formats. The Q7 tools are used to analyze historical data for solving a particular problem. Most problems occurring in production-related areas fall into this category. One the other hand, not all data needed for decision making are readily available and many problems call for collaborative decision among different functional areas. Under these situations, the M7 tools (also called the New Seven tools [Imai, 1986] are useful in areas such as product quality improvement, cost reduction, new-product development, and policy deployment, etc. 11) Constant process, product, and service improvement. A culture of constant improvement must be developed for TQM to succeed. All employees should be empowered with the ability to influence an organizational process that helps to improve quality. Once given this authority,

employees must show their desire and commitment to constantly improve the company. They must be always looking for ways to improve not only their part of the organization, but also the organization as a whole. Management must foster this culture through proper reward and recognition. 12) Incentivize TQM involvement. Incentive is a form of position reinforcement that is the fuel of the TQM torch. Most TQM implementers use a suggestion program to solicit cost reduction ideas from employees. The ideas are evaluated by a cross-functional suggestion evaluation team and the ones with significant contributions are implemented, and the suggesters are recognized and rewarded with money and fame. Texas Instruments has a recognition function called the Quality Hall of Fame ceremonies. 13) Information sharing. Teamwork is the key to the success of TQM, yet it relies on sharing the necessary information and know-how among the team members and across functional areas. It has been proven that sharing such information as profit, budget, schedule, progress, errors, etc. can provide the employees a sense of ownership and importance. It encourages the employees to push themselves to work harder in order to achieve the company goals as well as their personal goals. Nonetheless, any unnecessary or problematic information such as pay scale or bonus level should not be shared because it is dysfunctional and counter-productive. 14) Eliminate communication barriers. Under TQM culture, there should be no communication barriers between workers and management, and between functional areas. The management must make themselves available to and easily accessible by the workers. Employee suggestion program could be implemented in order to eliminate communication barriers. 15) Suppliers must have a TQM philosophy. A company cannot produce a quality product if the components of which it is made are faulty. Therefore, the supplier of a company must be trained and certified as a TQM supplier. Without such a certification, any components that are purchased from the supplier cannot be guaranteed to have the quality necessary for a company to establish a TQM culture. Similar to the JIT philosophy, the TQM philosophy advocates a strong relationship with its suppliers. One should cut down the number of suppliers and provide only a few TQM certified suppliers with long-term business commitments. This motivates the suppliers to make changes for continual quality improvement and ensures that the quality of the company's products will not be sacrificed. 4. The Deming Management Method Although Walter Shewhart is considered as the founding father of statistical quality control system, W. Edwards Deming is the first one who introduced the TQM concept. Deming offered the management his fourteen points of management obligations and identified seven deadly diseases and some obstacles of TQM implementation. The fourteen points as listed below are also known as the Deming management method [Walton, 1986]. 1) Create constancy of purpose for improvement of product and service. 2) Adopt the new philosophy of total quality. 3) Cease dependence on mass inspection to achieve quality. 4) End the practice of awarding business based on price tag alone. 5) Improve constantly and forever the system of production and service. 6) Institute training on the job. 7) Institute leadership.

VIII. CONCLUSION

In software development, software testing is highly desirable to assure the quality of the software product. Software testing performed via manual and software metrics, the former one (manual) is costly and it required high time interval to perform it because of it now a day software engineer moves to systematic measurement method which is software metrics. This study conducted to reveal to asses and analysis's

software metrics used to measure software quality particularly software product and process. Software metrics utilized to extent the software product and process. The researcher used a collection of literatures from various electronic databases which available since 2008 to understand and know the software metrics; the researcher has been identified Product personal satisfaction may be a method for measuring how product is intended what's more entryway great the software conforms to that configuration. Exactly of the variables that we need aid searching for software quality would Correctness, result quality, Scalability, culmination And nonattendance of bugs, In any case the quality standard that might have been utilized from one association will be unique in relation to others for this reason it will be better will apply those product measurements will measure the nature of product and the current most common software metrics tools. In the future the researcher recommends the specific application area of each software metrics and how can perform by the researcher to enhance the quality of software applications.

IX. REFERENCES

[1]. Ahad, A., Yalavarthi, S. B., & Ali Hussain, M. (2016). A new approach for integrating social data into groups of interest doi:10.1007/978-81-322-2755-7_24 Retrieved from www.scopus.com 2. Chandra Prakash, V., Sastry, J. K. R., Kantharao, V., Sriharshini, V., Sriram, G., & Ganesh, C. H. V. S. (2017). An expert system to assess memory power of a student for selection of a suitable career. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue 6), 309-321. Retrieved from www.scopus.com.

[2]. Chaitanya Krishna, B., Sai Mounish, P., Vinay Kumar, U., &Divya Mallika, A. (2018). Online business site quality assessment. International Journal of Engineering and Technology(UAE), 7, 838-840. Retrieved from www.scopus.com.

[3]. Sharma, N., &Yalla, P. (2016). Software engineering and natural language processing-how can they be together? International Journal of Software Engineering and its Applications, 10(12), 389-396. doi:10.14257/ijseia.2016.10.12.32.

[4]. Preetham, C. S., Siva Ganga Prasad, M., Pratap Reddy, N., &SashankaTeja, L. (2016). Outage probability analysis of amplifyandforward and decodeandforward dual hop relaying with hardware defects. Journal of Theoretical and Applied Information Technology, 86(2), 290-298. Retrieved from www.scopus.com.

[5]. Naga Malleswari, D., Suresh Babu, S., Moparthi, N. R., Mandhala, V. N., & Bhattacharyya, D. (2017). Hash based indexing in running high dimensional software systems. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue), 34-43. Retrieved from www.scopus.com.

[6]. Chandraprakash, V., Sastry, J. K. R., Sravani, G., Manasa, U. S. L., Khyathi, A., & Harini, A. (2017). Testing software through genetic algorithms – a survey. Journal of Advanced Research in Dynamical and Control Systems, 9(Special Issue 2), 1607-1633. Retrieved from www.scopus.com.

[7]. Shahabuddin, S. M., &Yalla, P. (2017). Impact of lean software development into agile process model with integration testing prior to unit testing. Journal of Theoretical and Applied Information Technology, 95(22), 6163-6175. Retrieved from www.scopus.com.

[8]. Naga Malleswari, D., Kumar, M. P., sathvika, D., & Kumar, B. A. (2018). A study on SDLC for water fall and agile. International Journal of

Engineering and Technology(UAE), 7(2), 10-13. doi:10.14419/ijet.v7i2.32.13516.

[9]. NagaMalleswari, D., & Subrahmanyam, K. (2018). SIS framework for risk assessment through quantitative analysis. International Journal of Engineering and Technology(UAE), 7(2.32 Special Issue 32), 367-370. Retrieved from www.scopus.com.

[10]. Lakhmi Prasanna, P., Rajeswara Rao, D., Meghana, Y., Maithri, K., &Dhinesh, T. (2018). Analysis of supervised classification techniques. International Journal of Engineering and Technology(UAE), 7(1.1), 283-285. Retrieved from www.scopus.com.

[11]. Lakshmi, K. N., & Rao, K. S. S. (2018). A study on role of emotional intelligence on employee performance. International Journal of Civil Engineering and Technology, 9(3), 440-448. Retrieved from www.scopus.com.

[12]. J. Cahill, J. M. Hogan, and R. Thomas, "Predicting Fault-Prone Software Modules with Rank Sum Classification," pp. 211-219, 2013.

[13]. R. C. and andF.Kemerer, "A Metrics Suite for Object Oriented Design " IEEE Transactions on Software Enginneering vol. 20, 1994.

[14]. RUdiger L, Jonas L, and W. L, "Comparing Software Metrics Tools," ACM, July 2008. Monday, August 01, 2016). Analyst4j Find Using Metrics Available: www.codeswat.com Syntegrico. Analyst 4j Find Using Metric. [24] (Friday October 30, 2015). CCCC Software metrics Available: http://www.sourceforge.net/projects/cccc/.

[15]. Chidamber and Kemerer Java Metrics Available: http://www.spinellis.gr/sw/ckjm/ [26] (Friday, October 30, 2015). Eclipse Metrics Plug-in 1.3.6. Available: http://sourceforge.net/projects/metrics/ Eclipse Metrics Plug-in 3.4 Available: : http://eclipsemetrics.sourceforge.net/.

[16]. Jarallah S, Raimi A, and Rufai and Sohel M, "OOMeter: A Software Quality Assurance Tool " Proceedings of the Ninth European Conference on Software Maintenance and Reengineering 2005 SciTools Source Code Analysis and Metrics, Understand for Java. Available: www.scitools.com.

[17]. D. a. Tom, Controlling Software Projects. New York: Yourdon Press, 1986.

[18]. Campbell , Luke, and B. K, "Software Metrics: Adding Engineering Rigor to a Currently Ephemeral Process " 1995.

[19]. Pooja P and D. A. Phalke, "Survey on Software Defect Prediction Using Machine Learning Techniques," International Journal of Science and Research, vol. 3, December 2014.

[20]. Malkit S and D. S, "Software Defect Prediction Tool based on Neural Network " International Journal of Computer Applications, vol. 70 2013.

[21]. H. R. Bhatti, "Automatic Measurement of Source Code Complexity," MASTER'S THESIS Computer Science and Engineering Luleå University of Technology.

[22]. Vikas V and S. M, "Applications of Software Testing Metrics In Constructing Models Of The Software Development Process " Journal of Global Research in Computer Science, vol. 2 pp. 96-98 May 2011.

[23]. Fernando, Wijayarathne, M. D. Fernando, and I. Guruge, "The Importance of Software Metrics: Perspective of A Software Development Projects In Sri Lanka," SAITM Research Symposium on Engineering Advancements, 2014.

[24]. C. Z. a. Q. SHICHAOZHANG, "Data Preparation for Data Mining " Applied Artificial Intelligence vol. 17, pp. 375–381 2003.

[25]. S. Shivaji, "Efficient Bug Prediction and Fix Suggestions " PhD A dissertation UNIVERSITY OF CALIFORNIA 2013.

[26]. Chayanika S, Sangeeta S, and R. S, "A Survey on Software Testing Techniques using Genetic Algorithm," International Journal of Computer Science vol. 10, pp. 381-393, January 2013.

[27]. D. Kafura, "A survey of software metrics " presented at the ACM annual conference on The range of computing New York 1985.

[28]. Ming Chang Lee and T. Chang, "Software Measurement and Software Metrics in Software Quality," International Journal of Software Engineering and Its Applications, vol. 7, 2013.

[29]. Poornima Gupta and P. Sahai, "A Review on Artificial intelligence Approach on Prediction of Software Defects," International Journal of Research and Development in Applied Science and Engineering vol. 9, February 2016.

[30]. Kunal Chopra and M. Sachdeva, "EVALUATION OF SOFTWARE METRICS FOR SOFTWARE PROJECTS " INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY vol. 14, April 2015.

[31]. S. D. Conte , a. V. Y. Shen, and W. M. Zage, "A Software Metrics Survey " in Technical Reports P. U. C. Science, Ed., ed, 1987.