Journal of Engineering and Applied Sciences 14 (15): 5111-5122, 2019
ISSN: 1816-949X

A Survey on Software Quality Assurance

Mua'ad Abu-Faraj
Department of Computer Information Systems, Faculty of Information Technology and Systems,
University of Jordan, 77110 Aqaba, Jordan m.abufaraj@ju.edu.jo, +962 32090450 Ext. 36087

Abstract: This study presents an overview of one of the software engineering topics in general and one of the software quality topics in specific which is Software Quality Assurance (SQA). This survey provides an extensive review to a number of studies and approaches done in software quality assurance also it includes a classification of these studies according to, its targeted proposed work. There are many studies done on software quality in different fields such as: education, economics, healthcare and software products. This survey focuses on the models of software quality and frameworks for measuring quality using metrics. Also, a number of previously done surveys and literature reviews on software quality assurance were included in this survey in order to provide a complete picture on this topic.

Key words: Software quality, software quality assurance, survey, targete, software, metrics

INTRODUCTION

Software quality is considered as one of the very important topics in software engineering. Software quality origins goes back to the 1970's Wadsworth *et al*. (2002). The outcomes of software quality affect the financial concerns in which if a software does not possess good or high quality, it will lead to increase in budgets required to maintain that software. And it also may not serve its purposes. Some safety critical systems are more affected by quality, since, it may lead to severe losses (human losses).

Software quality is a generic term in software engineering that referred to as what are the needed characteristics of a software product whether, if a software product does possess a specified characteristic. Another important aspect is the tools, processes and techniques needed to accomplish these characteristic (Bourque and Fairley, 2014).

Many definitions exist for software quality, for example, Humphrey (1989) defined it as achieving excellent levels of fitness for use. Crosby (1980) defined it as conformance to requirements.

In recent studies, new definitions of software quality appeared (OIN, 2011) defined it as the capability of software product to satisfy stated and implied needs under specified conditions (Anonymous, 2009, 2013). Defined it as the degree to which a software product meets established requirements, however, quality depends upon the degree to which those established requirements accurately represent stakeholder needs, wants and expectations.

Software quality assurance is part of software quality in which it is not just totally a testing activity but it is more than one activity that measures how adequate are the processes of the software in order to make sure that the appropriate processes are chosen and done in order to produce a good quality software that serves its purpose (Bourque and Fairley, 2014). There are lots of frameworks and models proposed in the literature that contributes to improving the process of software quality assurance, ranging from frameworks to assure software quality in education, models to improve software quality assurance in a number of fields such as organizations and in the industry, frameworks that present a way to use metrics for measuring how good a quality of software.

In this study, the scholar provides a survey on a number of studies focused on software quality assurance. Also, the scholar provides a classification of these studies, according to, what is targeted domain. In addition, to that some similar studies done on surveys of software quality assurance have been presented by the scholar too.

MATERIALS AND METHODS

Methodology of the survey: This study presents the method of doing this survey and how it is broken down. How we classified the approaches included in this study.

First of all, the survey includes data that has been collected from studies ranging from the years between 1980-2018. These studies contain a number of targeted domains that are related to software quality assurance, so, scholar breaks down these studies into a number of subsections according to, the targeted domain where each domain contains its relate d studies. The following subsections presents the targeted domain together with their related studies in a generic way.

Table 1: Papers published in the education domain with its contribution

Researchers	Contribution
Amin and Salih (2017)	Model
Owlia and Aspinwall (1996)	Framework
Chong and Crowther (2005)	Framework
Martinez-Caro *et al.* (2015)	Model
Chua and Dyson (2004)	Tools
Skalka *et al.* (2012)	Framework
Grifoll *et al.* (2010)	Model
Masoumi and Lindstrom (2012)	Mode
Welsh and Dey (2002)	Standards and guidelines
Laporte and April (2013)	Framework
European Association for Quality Assurance in Higher Education	Standards and guidelines
European Association for Quality Assurance in Higher Education	Standards and guidelines
Laporte *et al.* (2007)	Standards and guidelines

Table 2: Papers published in the economics domain with its contributions

Researchers/years	Contribution
Wagner and Meisinger (2006)	Model
Hampp (2012)	Model
Dao-Phan *et al.* (2014)	Model
Elberzhager *et al.* (2011)	Model
Alberts in (1976)	Standards and guidelines
Yau *et al.* (1990)	Framework
Nikolik (2012)	Standards and guidelines

Table 3: Papers published in the healthcare domain with its contributions

Researchers/years	Contribution
Nguyen *et al.* (2011)	Standards and guidelines
Linberg (1993)	Framework
Ihlenfeldt (1988)	Standards and guidelines

Education: The beginning of computing education researchers and stockholders start pay attention to quality of education process from the customers IDE to the technical IDE, recently many researchers start to digging in e-Learning of higher education as a field of the future of education. Table 1 shows the studies that were published and were targeted to the education domain, including the researches and year of publication.

Economics: Table 2 shows the studies that were published and were targeted to the economics domain, including the researchers and year of publication.

Healthcare: In the last decade, a lot of medical devices transformed into computing machines as an example heart regulating devices. Because of that many researchers start to digging in this area which demand high quality due to one result wrong could lead to a fatal energy in human being. There are some papers which discuss the quality assurance due to this reason. Table 3 shows the studies that were published and were targeted to the healthcare domain including the researchers and year of publication.

Studies and software products: Table 4 and 5 shows the studies that were published and were targeted to the studies and software products domains including the researchers and year of publication.

Table 4: Papers published in the studies domain with its contributions

Researchers/years	Contribution
Day and McVey (1986)	Standards and guidelines
Hedberg *et al.* (2007)	Standards and guidelines
Liu (2012)	Standards and guidelines
Ho-Stuart and Thomas (1996)	Standards and guidelines
Riabov (2011)	Tools
Illgen and Ortmann (2005)	Standards and guidelines
Boehm *et al.* (2008)	Standards and guidelines
Goodenough and McGowan (1980)	Standards and guidelines
Runeson and Isacsson (1998)	Standards and guidelines
Zuser *et al.* (2005)	Model
Nandakumar *et al.* (2014)	Standards and guidelines
Prasad (1994)	Standards and guidelines
Li *et al.* (2010)	Model
Grinthal in (1985)	Standards and guidelines
Saif *et al.* (2010)	Model
Davis *et al.* (1993)	Standards and guidelines
Murugan and Prakasam (2013)	Standards and guidelines

Table 5: Papers published in the education domain with its contributions

Researchers/(Years)	Contribution
Janus *et al.* (2012)	Tools
Yilmaz *et al.* (2005)	Model
Gill (2005)	Standards and guidelines
Lee (2014)	Framework
Hongying and Cheng (2011)	Model
Chen and Sorenson (2008)	Model
Goeb and Lochmann (2011)	Model
Xu *et al.* (2005)	Model
Alsultanny and Wohaishi (2009)	Model
Mishra and Mishra (2006)	Model
Wong *et al.* (2005)	Standards and guidelines
Rosenberg and Sheppard (1994)	Standards and guidelines
Goeb and Lochmann (2011)	Model
Sharma and Srivastava (2014)	Framework
Sangeetha *et al.* (2010)	Model
Pande *et al.* (2013)	Standards and guidelines
Khalane and Tanner (2013)	Standards and guidelines
Lee (2014)	Standards and guidelines
Shigeta *et al.* (2013)	Tools 2013
Abdi *et al.* (2012)	Standards and guidelines
Chulani *et al.* (2016)	Standards and guidelines

A survey on software quality assurance: This study presents a survey on the software quality assurance studies in the literature. The section breaks down into sections according to, the domain that the study is targeted to. Each study will be presented in the following sections. The domains as mentioned in the previous section were: education, healthcare, economic, software products.

Education

A framework for the dimensions of quality in higher education: By Owlia and Aspinwall (1996) presents a conceptual framework that targets dimensions of quality in higher education and how to improve the quality of higher education. The framework is based on interpreting dimensions of quality in contexts that are non-educational and reviewing proposed quality factors that were published for higher education. In order to examine the framework to validate it, it needs an empirical study to be performed which is the next step in the researcher's research program.

A new framework for measuring the quality of outcomes-based engineering education: By Chong and Crowther (2005) proposes a conceptual framework that is based on theory, the framework is targeted towards the transnational engineering education. The integrity of the proposed framework is tested using surveys by representatives of employing agencies, staff and enrolled students. The validity of the preliminary conceptual framework was high with some limitations. The results of the surveys enabled the development of a revised conceptual framework of the transnational engineering education. The revised framework has two comprised dimensions: contributory and outcomes. A 13-dimension revised survey instrument taken from the transposed framework consists of 25 outcome items, these items were divided into 5 dimensions and 49 contributory items divided into 8 contributory dimensions.

An application of the performance-evaluation model for e-Learning quality in higher education: The purpose of this study (Martinez-Caro *et al*., 2015) is to manage the practices of software quality to increase user satisfaction and software continuous improvement, to achieve that the researchers provides evaluation model. The main goal of this model is to improve student satisfaction.

Applying the ISO 9126 Model to the evaluation of an e-Learning system: By Chua *et al*. the researcher suggest that ISO 9126 Model standard of quality can improve tool of evaluating software quality product educational system, it also shows how the proposed model in this study can detect design flaws in e-Learning system.

e-Learning courses quality evaluation framework as part of quality assurance in higher education: By Skalka *et al*. (2010) the Researcher produces the concept proof to course of e-Learning which contains an implementation and evaluation of elearning frameworks in addition, to that the researchers presents a summary of recent issues related to software quality assurance.

Quality assurance of e-Learning: By Grifoll *et al*. (2010) the researchers presents a workshop that provides a platform which discusses the e-Learning based system specially the internal challenges. The researchers also discusses that a common language was needed and an approach that integrated group of approaches which helps with improving the provision in e-Learning system.

Quality in e-Learning; A framework for promoting and assuring quality in virtual institutions: By Masoumi and Lindstrom (2012) the researchers produce a framework of e-Quality in e-Learning system which considering the advantages and disadvantage from the previous framework and model. Also, in this study, the researchers provide a tool work based on benchmarks and factors of e-Quality model.

Quality measurement and quality assurance in higher education: The researchers Welsh and Dey (2002) provide discussion of QMS 2000 which a technology of continuous measurement of quality, it is a study that include 273 students, faculty and employer the purpose of this study was to measure the satisfaction of this sample. The user of QMS 2000 can generate reports and applying a statistical analysis from university database.

Software quality assurance in an undergraduate software engineering program: By LaPorte and Aprile (2013) presents a discussion done at the ceole de Technologie Suprieure (TS) in Canada, it targets the positive impact of the practices of software quality assurance for delivering a software that possesses high quality within budget and on time. The study has two parts. The first overviews the software engineering undergraduate program offered at the TS. After that a software quality assurance course detailed description is presented with some focus on the sessions held at the laboratories.

Economics

Integrating a model of analytical quality assurance into the V-Model: The researchers Wagner Meisingeret (2006) discuss an integration between systems development process such as V-Model XT with stochastic model of quality assurance in economics level. Modular is the base of integration in this process module, this concept is derived directly from the V-Modell XT for purposes of analyzing quality assurance. Providing a new process model and compare it with the existing V-Model elements, the approach, here, concentrates on work products, activities and roles.

A cost-benefit model for software quality assurance activities: By Hampp (2012) presents COBE, a quantitative model that is validated and developed by the researchers. Quantification is done through using historical data and the model is built with relationships that have specific details. Features of the model include allowing the decision on what tests and reviews should be done and how to do them also when a defect is fixed it shows how to retest this fix. The outputs of the model are the benefits and cost for the activities of quality assurance that can be used in the development activity and after delivering the

project. The results are represented as three terms: staff, time and effort. These three are then weighted and summed in financial manner to find an optimal trade-off between benefit and cost. Data from real world has been applied to the model to validate it, data from 21 projects from students were used. Two projects from the industry were also used. The accuracy of the results of the model was sufficient. The application of the model in a specific environment needs more calibration.

Developing method for optimizing cost of software quality assurance: By Dao-Phan *et al.* (2014) proposes a methodology for the cost optimization of the software quality assurance based on proposed regression model. Regression analysis is used to measure the defects number in software. The proposed method is an optimal software quality assurance method that is based on constraint conditions and with the use of linear programming techniques. The study also presents, the results of the analysis in two sides, a theoretical side and an empirical side. Both sides are also evaluated.

Optimizing cost and quality by integrating inspection and test processes: By Elberzhager *et al.* (2011) contains the new version of the approach that focuses on prioritizing defect types for testing. A previous version of the approach was focused on prioritizing code classes. The result is derived from a case study in the code level which assumes that prioritization could happen before the testing. Last section contains the related work and the future research that can be created.

The economics of software quality assurance: By Alberts examines the economy of the software quality assurance to determine at which phase it will be the most beneficial by analyzing the number of errors that occur is each phase of project life cycle aiming to increase the quality of the software product by using structured programming, automated tools, management techniques and top down designs.

An integrated expert system framework for software quality assurance: By Yau *et al.* (1990) uses knowledge-based engineering technology to propose a framework for software quality assurance. It helps in providing software quality assurance assistance based on knowledge during the software development lifecycle. Throughout the development lifecycle, software metrics are used for the prediction and measurement of the software product quality to make sure that the software possesses high quality and the software maintenance and development achieve cost-effectiveness.

Software quality assurance economics: By Nikolik (2012) proposes a case study targeted on an industrial data, it is concerned on what is the relation between the value of test cases and their cost and how the value and cost sensitiveness on artifact and passage of time can change.

Healthcare

A view of 21st century healthcare industry and software quality improvement practices: By Nguyen *et al.* (2011) highlights that how the development of software affects the healthcare field in the current and future state. It also discusses the challenges and the current state of software quality and provides possible solutions to these challenges, solutions targets improving and assuring the software quality practices.

Defining the role of software quality assurance in a medical device company: By Linberg (1993) discusses how a medical device company defines its program of software quality assurance using a proposed quality framework. This framework can also be used by other companies and organizations that need to start their software quality assurance program. The frame-work also helps in improving product quality and provides consistent support for many projects.

Quality begins at home; The role of project leader in software quality assurance: Expertise of the researcher Ihlenfeldt (1988) is used to provide quality assurance in new projects from the leader of the project role view. The purpose of this study is to cover two aspects: software and hardware. The software includes development and design and the hardware includes microprocessor of heart and lung machine. A description for project techniques were implemented successfully and recommendations were proposed to project leader for improving such projects.

Studies

A survey on software quality assurance: By Day and McVey (1986) presents the outcomes of a survey done on software quality assurance. A collection of 52 sets of questionnaires was collected and analyzed. The survey discusses the data processing, survey methodology and analysis. The main points of this survey is highlighting the encountered quality assurance problems, software quality investments trends, the software quality assurance activities and the people that are involved in it and the used tools and their weaknesses.

Assuring quality and usability in open source software development: By Hedberg *et al.* (2007) presents a literature

review on usability and quality assurance in open source software development. The focus is on software that is targeting large user population. One of the things taken into account is the non-computer professional users and nave users to help improving the usability. The study also shows a trade-off between usability and quality in terms of two fields: human computer interaction and software engineering. A summary of the practices currently used is also presented to make sure quality and the usability is realized in the open source software development. In addition, to this the recommendations of proposed practices to be used are also presented. Finally, the study highlights the current research limitations and how to improve them for future research.

Formal engineering methods for software quality assurance: By Liu (2012) contains a brief survey on using engineering formal methods and its challenges and how you can face and solve these challenges through the use of a mathematical notation. The researcher suggested that the use of formal methods could improve the reliability and controllability of the software and decrease the complexity of the project.

Laboratory practice with software quality assurance: By Stuart and Thomas is an experimental study done in queens land university on practical work in combining between in house and outsource project. The researcher discusses the interaction between the student quality assurance with student project in two aspects: reviewing and walkthroughs based on document. The outputs of this study are some tentative drawing conclusion finished by analyzing the experience with an experiment.

Methodologies and tools for the software quality assurance course: To choose good tool sand methods in software quality assurance course you can choose from: object oriented approach, predicate-logic and metric theory. The researcher provides an executive summary of SQA Software tools, lab manuals, project reports, homework assignments and research papers of students who attend the SQA course. The advantages and disadvantages of using online tools and hybrid tools are discussed by Riabov (2011).

Process oriented software quality assurance an experience report in process improvement OEM perspective: By Illgen and Ortmann (2005) produces two views about quality assurance: constructional in the design level of view and the verification in destructional level of view. Destructional level is part of customer point of view. In the software lifecycle the effort changes dynamically in an organization of each role in the system also for each view the researcher propose.

Sixth workshop on software quality: By Boehm *et al.* (2008) views the quality from different perspective, the point of view for all stakeholders involved in the quality aspects and tries to understand their view which is affected by many factors such as cultural issues and outsourcing issues. The study discusses all these challenges by gathering academic, industrial and commercial communities interested in software quality in the sixth workshop of software quality.

Software quality assurance; Testing and validation: By Goodenough and McGowan (1980) aimed to apply the software quality assurance concepts to hardware by building a high quality software product then analyzing the common properties and characteristics between both. Because implementing the hardware is more mistakable than the software .

Software quality assurance-concepts and misconceptions: By Runeson and Isacsson (1998) the researcher presented different software quality assurance perspective and determine the difference between SQA in software product and SQA in management. The idea of the SQA in software product is the ability to build mature SQA system which can control the whole quality of the software product by implement it in each level of procedure as needed (Fig. 1).

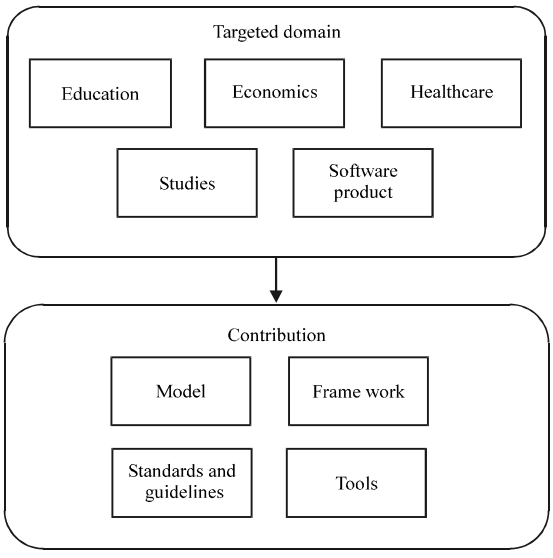

Fig. 1: Mapping between targeted domains and contribution

Software quality development and assurance in RUP, MSF and XP-A comparative study: By Zuser *et al.* (2005) compared three of the most popular software development process models in industrial filed regarding the quality of its support. Considering the support as one of the software quality assurance aspects, this study based on the result of real-life standards quality support in software development process models.

State of the art in software quality assurance: The researchers by Nandakumar *et al.* (2014) present their point of view regarding the word "quality" which appeared in the literature review of the last 5 years and talking about some other aspects of software quality assurance. The study cover the following eight aspects: quality models, timely QA feedback, quantitative approaches to predicting software quality and the effectiveness of Software QA, optimal choice of QA methods, design level QA, impact of parallel development options on software quality, continued QA efforts even after the operational deployment of the software product, and the use of CASE tools and perceived value for QA at NASAJPL. The studys also view a summary for three ongoing researches regarding the quality.

Towards a zero-defect product; the end-to-end test process: By Prasad (1994) the researcher presents the quality assurance from the tester point of view to develop zero defects software considering all stakeholders and factors (customers, lines of code., etc) as a part of end-to-end test which lead to high quality product with no defects.

Transition from a plan-driven process to Scrum 5A longitudinal case study on software quality: By Li *et al.* (2010) presents a longitudinal study which followed a project over a 3 years period and compared software quality assurance processes and software defects of the project between a 17 months phases with a plan-driven process, then by a 20 months phase with Scrum. The results of the study did not show a significant reduction of defect densities or changes of defect profiles after Scrum was used. The iterative nature of Scrum resulted in constant system and acceptance testing and related defect fixing which made the development process more efficient in terms of fewer surprises and better control of software quality and release date.

Tutorial software quality assurance for CAD: By Grinthal presents a tutorial that provides an overview of Software Quality Assurance (SQA) especially when applied to Computer AiM Design (CAD) systems. The study also, discusses software quality assurance basic concepts and the current state of the art and a theoretical background is also discussed. The study also demonstrates a software quality assurance pragmatic approach in the CAD world through over viewing the VLSI layout editor and the CAD application itself. Other topics highlighted in the study include: robustness, reliability, test strategies, regression testing, verification, software change control, generation of test data and how to handle some of the problems encountered when many people work on the same program.

An analysis of a comprehensive planning framework for customizing SQA: By Saif *et al.* (2010) presents an analysis of Frank Elberzhanger and Christian dengers proposed framework. It also discusses the limitations and shortcomings of these frameworks. And it provides a proposed improvement to refine this framework to a better version. The study also presents a new framework/model based on the outcomes of the discussed framework.

Industrial acceptance of software quality assurance standards: By Davis *et al.* (1993) the researcher presents an investigation research of quality assurance in the last decade to improve the acceptance of Quality Assurance Standards (QAS) in the industry. Results were derived from researches and surveys related to major findings and software production. This needs to analyze the quality assurance methods and group of QAS. The researcher also provides a formal proposal that relate to QAS acceptance process.

A literal review of software quality assurance: By Murugan and Prakasam (2013) highlights software quality assurance concepts that are of high importance, these concepts are used in software development in order to produce a software that is concerned with time consumption and has error free system.

Software product

The 3C approach for agile quality assurance: By Janus *et al.* (2012) the researcher presents the 3C approach as an extension for the agile method integration by adding a continuous measurements subsequent activities to made metrics base quality gates. This approach was developed on Java web based applications including open source tools and it was developed and proven using the agile maintenance for an industrial German company.

Main effects screening

A distributed continuous quality assurance process for monitoring performance degradation in evolving software

systems: By Yilmaz *et al.* (2005) discusses performance assessments that can be improved across large configuration space to achieve that they produced a process in effects screening. That is used to improve the formality of the designed experiments which helps in reducing the configuration space. So, this would improve the effectiveness to perform targeted within the organization by applying feasibility studies on large scale projects and well known projects. Using this study, the researcher evaluates this process. After analyzing the data, the researchers indicate that using conventional techniques are less efficient than performing main effect screening on large scale systems.

Factors affecting effective software quality management revisited: By Gill (2005) aimed at examining and revising some software development process aspects that may affect the process of software quality management. Aspects include ISO approach applicable to software quality, reliability measurement and some aspects related to software testing improvement. It is not guaranteed that software evaluation and testingtools/methods/techniques ensure high quality software and achieve effective testing. One method for improving testing effectiveness is by improving the software developer's attitude towards testing. The above-mentioned factors that affect software quality management were also discussed and highlighted in this study. So, all the possible ways to improve them is also suggested. The outcome of the study might help researchers to quantify specific tools for measurement of such quality attributes.

Software reliability assurance using a framework in weapon system development a case study: The Lee *et al.* (2009) proposed a quality model to insure the reliability of weapon systems in Korean defense domain. The framework provides guidelines on how to measure the reliability in software organizations and follow the enhancements of the software engineering process that support activates and indicators. Also, the study presents an empirical study on the application of the proposed framework and analyzes the results.

A customizable agile software quality assurance model: The researcher by Hongying and Cheng (2011) presents a new Agile Quality Assurance Model (AQAM). The model can be used in the software industry for incorporating new changes, it is flexible in this manner. The model also provides templates and detailed guidelines for customizing and implementing it in the real world. The model can be used in agile software development teams that range from medium to small sized teams. Iit benefits teams for improving the capabilities of their quality assurance skills. The model contains 20 key process areas where area contains maturity levels, guidelines, processes, benefits, templates, customization and best practices.

A QoS-based service acquisition model for IS services (customer point of view): By Chen and Sorenson (2008) demonstrates the definition of quality of service for information systems. It also examines the ways for discovering and selecting high quality information systems services from the customer's point of view. A model of quality of service for information systems service acquisition was developed with the introduction of the quality assurance party and two well-known terms of quality. Software as a Service (SAAS) and provider/customer service.

A unifying model for software quality: By Goeb and Lochmann (2011) presents a general quality model. The model provides a possible way for describing quality related concepts in a different way. The researchers of the study provide evidence of the capability of their model of integrating different concepts from guidelines, quality models, standards and static code checker rules. They also highlight the ability of their model in describing the interrelations of disciplines such as software testing, software requirements and software quality. The quality model also provides an establishment for software quality related concepts which enables continuity and consistency of information related to quality during the development of software.

Metric based software quality assurance system: By Xu *et al.* (2005) produces a SQA system abbreviated (PM-SQA) which contains a generic process modeling tool which that can support product models, resource models, functional models and role models. After executing the process, the useful data will be automatically collected and some suitable metric entities can be chosen. The main goal of implementing the system is to support collaborative working and distributed development.

Requirements of software quality assurance model: The researcher Alsultanny and Wohaishiet (2009) provides a model of software quality to evaluate the quality factors which affect on software quality and to increase the software productivity focusing on complexity of software design that faces the implementer. Based on ISO 9126 the proposed approach by the researcher of this study provides a reliable, safe and high quality product.

Software quality assurance models in small and medium organizations a comparison: By Mishra and Mishra (2006) presents a software quality model for small and medium projects organizations by extracting the main characteristics, advantage and disadvantages for the model used to enhance the quality in small and medium software projects organizations. As the researcher assumed, the most of outsourcing projects are considered small or medium projects, then when apply the existing models (models for huge projects) on these projects the cost of implementation will be high.

Third workshop on software quality: By Wong *et al.* (2005) is talking about the third workshop for quality which gathering the industrial commercial. People interested in software quality assurance topics to discuss the quality from different perspective in order to reach to a high level of quality standards which satisfy both industrial and commercial levels.

Metrics in software process assessment, quality assurance and risk assessment: Two concepts you must take in your consideration when you develop software processes: you must understand it better and treat maintenance more carefully to obtain high degree of reliability. The researchers by Rosenberg and Sheppard (1994) consider the software process model in the context of its maintainability, quality of the code and risk assessment. A software process assessment analyzed in this study also depict a data collection set of 4 years. The last section of the study discusses the preliminary results and defines the metric that the researcher used.

A software quality model for SOA: By Goeb and Lochmann (2011) presents a unified meta-model that describes the service oriented systems quality, it is an enhanced version of the Quamcometa-Model. The study also presents examples of software quality based systems quality model. This model is an initial model that is based on empirical results from the software quality assurance quality community sources. The contradictions and similarities between and within the software quality assurance quality models become transparent because of the integration of sources of information. This is considered as a baseline in the manner of denying the comprehension of software quality assurance quality model. This approach also presents features of software quality assurance that distinguishes the quality modeling.

Generalized software quality assurance technique for maintenance parameter evaluation: An approach presented by Sharma and Srivastava (2014) that concern of estimation of cost presented in this study in addition, to reengineering assessment ROI computation.

Improved software quality assurance techniques using safe growth model: By Sangeetha *et al.* (2010) produced a description of approach related to software testing. This approach used Bayesian models of graphics also, the researcher provides a mechanism that presented the formal logic to structure of the software. The researcher also discusses that the model provides dynamic representation to solve testing software problem.

Software quality assurance activities of ITERCODAC: By Khalane and Tanner (2013) the researcher presents the activities of software quality assurance defined by CODAC. There are four main phases of software quality assurance development, maintenance, verification and validation teams are responsible of analyze conformance of product, conformance of process, data process of record and processes improvements performed.

Software quality assurance in scrum The need for concrete guidance on SQA strategies in meeting user expectations: By Khalane and Tanner (2013) presents a way to identify what are the most important concerns that matters in terms of the relation of the software quality assurance to the project stakeholders in a scrum environment.

Software quality factors and software quality metrics to Enhance software quality assurance: By Lee (2014) proposes a solution to the problem of customer value evaluation. It proposes a framework that combines criterion of software quality and it provides a description of software metrics, builds an application that pro-vides software quality metrics in terms of software quality assurance with quality lifecycle.

Tool use within NASA software quality assurance: By Shigeta *et al.* (2013) presents an investigation on the usage of tool support in organizations of assurance in NASA. It describes a long-term work done at the Jet Propulsion Laboratory in order to provide the assurance organizations with tools information that they might need in order to use these tools in an effective way.

Using security metrics in software quality assurance process: By Abdi *et al.* (2012) presents a method that focuses on security metrics in various phases during software development and does a validation on them using criteria of standards. With the use of this proposed work during the life cycle of development, it will ensure that qualified and secure final product will be produced.

RESULTS AND DISCUSSION

Challenges: The challenge in software quality assurance depends on the type of product and software such as economic, healthcare and education. healthcare quality software demand high accuracy, safety and performance, for economic the main challenges is applying high accuracy, software such as education demand a high user acceptance.

From our point of view the most common and widespread domain of quality assurance are education, healthcare and economic. Software quality assurance consists of four contribution type framework, model, method and tools. By analyzing and studying above four domain, we conclude that accuracy, user acceptance, performance and safety are the most important quality factors to achieve high quality assurance.

The most contribution produced in our review related to education domain is framework contribution. The most contribution produced in our review related to healthcare domain is guideline and standards contribution. The most contribution produced in our review related to economy domain is model contribution. The most contribution produced in our review related to study domain is guidelines and standards contribution. The most contribution produced in our review related to software product domain is model contribution.

Quality assurance factors related to healthcare are satisfaction, accessibility, effectiveness and safety. The quality assurance factors related to economic domain are productivity, efficiency, effectiveness, satisfaction.

The quality assurance factors related to education domain are teachers satisfaction, weak supervision of daily performance, poor quality of students joining secondary education, space or distance between candidates and effectiveness, security.

The quality assurance factor related to software quality product, there are many factor related to this domain, we will mention the most important of them security, safety effectiveness, efficiency, accuracy, productivity and user satisfaction.

CONCLUSION

In this study, scholar presented a survey on software quality assurance. The survey is based on a number of studies and approaches done in this field, scholar classifies these studies according to what is the targeted domain is also scholar presents some similar studies done on surveys of software quality assurance.

There are studies done on software quality for education, economics of software, quality assurance in healthcare software systems and quality assurance in software products.

ACKNOWLEDGEMENT

The researcher would like to thank the University of Jordan for their support to this research project.

REFERENCES

Abdi, A., A. Souzani, M. Amirfakhri and A.B. Moghadam, 2012. Using security metrics in software quality assurance process. Proceedings of the 2012 6th International Symposium on Telecommunications (IST), November 6-8, 2012, IEEE, Tehran, Iran, ISBN:978-1-4673-2072-6, pp: 1099-1102.

Alsultanny, Y.A. and A.M. Wohaishi, 2009. Requirements of software quality assurance model. Proceedings of the 2009 2nd International Conference on Environmental and Computer Science (ICECS '09), December 28-30, 2009, IEEE, Washington, DC, USA., ISBN:978-0-7695-3937-9, pp: 19-23.

Amin, F.M. and N.K. Salih, 2017. New model to achieve software quality assurance in E-learning application. Intl. J. Comput. Sci. Issues, 14: 65-69.

Anonymous, 2009. Standards and guidelines for quality assurance in the European higher education area. European Association for Quality Assurance in the European Higher Education, Helsinki, Finland.

Anonymous, 2013. P730/D9, Nov 2013-IEEE approved draft standard for software quality assurance processes. IEEE, New York, USA. https://ieeexplore.ieee.org/document/6781526

Boehm, B., S. Chulani, J. Verner and B. Wong, 2008. Sixth workshop on software quality. Proceedings of the Companion of the 30th International Conference on Software Engineering (ICSE Companion '08), May 10-18, 2008, ACM, Leipzig, Germany, ISBN:978-1-60558-079-1, pp: 1035-1036.

Bourque, P. and R.E. Fairley, 2014. Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0. 3rd Edn., IEEE Computer Society Press, Los Alamitos, California, ISBN:9780769551661, Pages: 348.

Chen, X. and P.G. Sorenson, 2008. A QoS-based service acquisition model for IS services. Proceedings of the 6th International Workshop on Software Quality (WoSQ '08), May 10, 2008, ACM, New York, USA., ISBN:978-1-60558-023-4, pp: 41-46.

Chong, B.K. and F. Crowther, 2005. A new framework for measuring the quality of outcomes-based engineering education. Proceedings of the 35th Annual International Conference on Frontiers in Education (FIE'05), October 19-22, 2005, IEEE, Indianopolis, Indiana, USA., pp: F1E-22.

Chulani, S., B. Boehm, J. Verner and B. Wong, 2006. Workshop description of 4th workshop on software quality (WOSQ). Proceedings of the 28th International Conference on Software Engineering (ICSE '06), May 20-28, 2006, ACM, Shanghai, China, pp: 1019-1020.

Crosby, P.B., 1980. Quality is Free: The Art of Making Quality Certain. The New American Library, New York, USA., ISBN:9780451624680, Pages: 270.

Dao-Phan, V., T. Huynh-Quyet and V. Le-Quoc, 2014. Developing method for optimizing cost of software quality assurance based on regression-based model. Proceedings of the International Conference on Nature of Computation and Communication, November, 24-25, 2014, Springer, Cham, Switzerland, pp: 243-253.

Davis, C.J., J.B. Thompson and P. Smith, 1993. Industrial acceptance of software quality assurance standards. Proceedings of the 1993 International Symposium on Software Engineering Standards, August 30-September 3, 1993, IEEE, Brighton, UK., ISBN:0-8186-4240-8, pp: 199-208.

Day, R. and T. McVey, 1986. A survey of software quality assurance. IEEE. Aerosp. Electron. Syst. Magaz., 1: 21-26.

Elberzhager, F., J. Munch, D. Rombach and B. Freimut, 2011. Optimizing cost and quality by integrating inspection and test processes. Proceedings of the 2011 International Conference on Software and Systems Process (ICSSP '11), May 21-22, 2011, ACM, Waikiki, Honolulu, Hawaii, USA., ISBN:978-1-4503-0730-7, pp: 3-12.

Gill, N.S., 2005. Factors affecting effective software quality management revisited. ACM. SIGSOFT Software Eng. Notes, 30: 1-4.

Goeb, A. and K. Lochmann, 2011. A software quality model for SOA. Proceedings of the 8th International Workshop on Software Quality (WoSQ '11), September 04, 2011, ACM, Szeged, Hungary, ISBN:978-1-4503-0851-9, pp: 18-25.

Goodenough, J.B. and C.L. McGowan, 1980. Software quality assurance: Testing and validation. Proc. IEEE., 68: 1093-1098.

Grifoll, J., E. Huertas, A. Prades, S. Rodriguez and Y. Rubin *et al*., 2010. Quality Assurance of E-learning.ENQA,Brussels,ISBN:978-952-5539-51-6, Pages: 50.

Hampp, T., 2012. A cost-benefit model for software quality assurance activities. Proceedings of the 8th International Conference on Predictive Models in Software Engineering, September 21-22, 2012, ACM, New York, USA., pp: 99-108.

Hedberg, H., N. Iivari, M. Rajanen and L. Harjumaa, 2007. Assuring quality and usability in open source software development. Proceedings of the 1st International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS'07), May 20-26, 2007, IEEE, Minneapolis, Minnesota, USA., pp: 2-2.

Ho-Stuart, C. and R. Thomas, 1996. Laboratory practice with software quality assurance. Proceedings of the International Conference on Software Engineering: Education and Practice, January 24-27, 1996, IEEE, Dunedin, New Zealand, pp: 220-225.

Hongying, G. and Y. Cheng, 2011. A customizable agile software quality assurance model. Proceedings of the 2011 5th International Conference on New Trends in Information Science and Service Science (NISS) Vol. 2, October 24-26, 2011, IEEE, Macao, China, ISBN:978-1-4577-0665-3, pp: 382-387.

Humphrey, W.S., 1989. Managing the Software Process. Pearson Educationm, Uttar Pradesh, India, ISBN:9788177583304, Pages: 512.

Ihlenfeldt, L.D., 1988. Quality begins at home: The role of project leader in software quality assurance. Proceedings of the Symposium on the Engineering of Computer-Based Medical Systems, June 8-10, 1988, IEEE, Minneapolis, Minnesota, USA., pp: 147-151.

Illgen, T. and S. Ortmann, 2005. Process oriented software quality assurance-an experience report in process improvement-OEM perspective [automotive applications]. Proceedings of the International Conference on Design, Automation and Test in Europe, March 7-11, 2005, IEEE, Munich, Germany, ISBN:0-7695-2288-2, pp: 536-537.

Janus, A., A. Schmietendorf, R. Dumke and J. Jager, 2012. The 3C approach for agile quality assurance. Proceedings of the 3rd International Workshop on Emerging Trends in Software Metrics (WETSoM '12), June 3, 2012, IEEE Press, Piscataway, New Jersey, USA., ISBN:978-1-4673-1762-7, pp: 9-13.

Khalane, T. and M. Tanner, 2013. Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. Proceedings of the 2013 International Conference on Adaptive Science and Technology (ICAST), November 25-27, 2013, IEEE, Pretoria, South Africa, ISBN:978-1-4799-3067-8, pp: 1-6.

Laporte, C.Y. and A. April, 2013. Software quality assurance in an undergraduate software engineering program. Proceedings of the International Conference on Canadian Engineering Education Association (CEEA13), June 17-20, 2013, Quality Control, Montreal, Canada, pp: 1-6.

Laporte, C.Y., A. April and K. Bencherif, 2007. Teaching software quality assurance in an undergraduate software engineering program. Software Qual. Profess., 9: 1-14.

Lee, D., J. Baik and J.H. Shin, 2009. Software reliability assurance using a framework in weapon system development: A case study. Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2009), June 1-3, 2009, IEEE, Shanghai, China, ISBN:978-0-7695-3641-5, pp: 989-995.

Lee, M.C., 2014. Software quality factors and software quality metrics to enhance software quality assurance. Bri. J. Appl. Sci. Technol., 4: 3069-3095.

Li, J., N.B. Moe and T. Dyba, 2010. Transition from a plan-driven process to Scrum: A longitudinal case study on software quality. Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10), September 16-17, 2010, ACM, New York, USA., ISBN:978-1-4503-0039-1, p: 1-10.

Linberg, K.R., 1993. Defining the role of software quality assurance in a medical device company. Proceedings of 6th Annual IEEE International Symposium on Computer-Based Medical Systems, June 13-16, 1993, IEEE, Ann Arbor, Michigan, USA., ISBN:0-8186-3752-8, pp: 278-283.

Liu, S., 2012. Formal engineering methods for software quality assurance. Front. Comput. Sci., 6: 1-1.

Lochmann, K. and A. Goeb, 2011. A unifying model for software quality. Proceedings of the 8th International Workshop on Software Quality, September 4, 2011, ACM, New York, USA., ISBN:978-1-4503-0851-9, pp: 3-10.

Martinez-Caro, E., J.G. Cegarra-Navarro and G. Cepeda-Carrion, 2015. An application of the performance-evaluation model for E-learning quality in higher education. Total Qual. Manage. Bus. Excell., 26: 632-647.

Masoumi, D. and B. Lindstrom, 2012. Quality in E-learning: A framework for promoting and assuring quality in virtual institutions. J. Comput. Assisted Learn., 28: 27-41.

Mishra, A. and D. Mishra, 2006. Software quality assurance models in small and medium organisations: A comparison. Intl. J. Inf. Technol. Manage., 5: 4-20.

Murugan, C.S. and S. Prakasam, 2013. A literal review of software quality assurance. Intl. J. Comput. Appl., 78: 25-30.

Nandakumar, R., A. Lal and R. Parmar, 2014. State of the art in software quality assurance. ACM. SIGSOFT Software Eng. Notes, 39: 1-6.

Nguyen, T.D., H. Guo, R.N. Naguib and N. Wickramasinghe, 2011. A view of 21st century healthcare industry and software quality improvement practices. Intl. J. Netw. Virtual Org., 9: 155-168.

Nikolik, B., 2012. Software quality assurance economics. Inf. Software Technol., 54: 1229-1238.

OIN., 2011. ISO-IEC 25010: 2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-system and software quality models. International Organization for Standardization, Geneva, Switzerland.

Owlia, M.S. and E.M. Aspinwall, 1996. A framework for the dimensions of quality in higher education. Qual. Assur. Educ., 4: 12-20.

Pande, S., F. DiMaio, C. Kim, J. Kim and W.D. Klotz *et al.*, 2013. Software quality assurance activities of ITER CODAC. Fusion Eng. Des., 88: 1437-1440.

Prasad, R.K., 1994. Towards a zero-defect product-the End-To-End test process. Proceedings of the 1st International Conference on Software Testing, Reliability and Quality Assurance, December 21-22, 1994, IEEE, New Delhi, India, pp: 137-147.

Riabov, V.V., 2011. Methodologies and tools for the software quality assurance course. J. Comput. Sci. Coll., 26: 86-92.

Rosenberg, L.H. and S.B. Sheppard, 1994. Metrics in software process assessment, quality assurance and risk assessment. Proceedings of the 2nd International Symposium on Software Metrics, October 24-26, 1994, IEEE, London, England, UK., ISBN:0-8186-5865-7, pp: 10-16.

Runeson, P. and P. Isacsson, 1998. Software quality assurance-concepts and misconceptions. Proceedings of the 24th International Conference on Euromicro Vol. 2, August 27, 1998, IEEE, Vasteras, Sweden, pp: 853-859.

Saif, S., A.A. Khan and F. Arif, 2010. An analysis of a comprehensive planning framework for customizing SQA. Proceedings of the 2010 International Conference on National Software Engineering, October 4-5, 2010, ACM, New York, USA., ISBN:978-1-4503-0026-1, pp: 1-7.

Sangeetha, M., C. Arumugam, K.S. Kumar and K. Akila, 2010. Improved software quality assurance techniques using safe growth model. Intl. J. Comput. Sci. Eng., 2: 2125-2132.

Sharma, S. and S. Srivastava, 2014. Generalized software quality assurance technique for maintenance parameter evaluation. Proceedings of the 2014 International Conference on Computing for Sustainable Global Development (INDIACom), March 5-7, 2014, IEEE, New Delhi, India, ISBN:978-93-80544-10-6, pp: 613-618.

Shigeta, D., D. Port, A.P. Nikora and J. Wilf, 2013. Tool use within NASA software quality assurance. Proceedings of the 2013 46th Hawaii International Conference on System Sciences (HICSS), January 7-10, 2013, IEEE, Wailea, Maui, Hawaii, USA., ISBN:978-1-4673-5933-7, pp: 4938-4947.

Skalka, J., M. Drlik and P. Svec, 2012. E-learning courses quality evaluation framework as part of quality assurance in higher education. Proceedings of the 2012 15th International Conference on Interactive Collaborative Learning (ICL), September 26-28, 2012, IEEE, Villach, Austria, ISBN:978-1-4673-2425-0, pp: 1-5.

Wadsworth, H.M., K.S. Stephens and A.B. Godfrey, 2002. Modern Methods for Quality Control and Improvement. 2nd Edn., Wiley, Hoboken, New Jersey, USA., ISBN:9780471299738, Pages: 683.

Wagner, S. and M. Meisinger, 2006. Integrating a model of analytical quality assurance into the V-Modell XT. Proceedings of the 3rd International Workshop on Software Quality Assurance (SOQUA '06), November 6, 2006, ACM, New York, USA., pp: 38-45.

Welsh, J.F. and S. Dey, 2002. Quality measurement and quality assurance in higher education. Qual. Assur. Educ., 10: 17-25.

Wong, B., J. Verner, S. Chulani and B. Boehm, 2005. Third workshop on software quality. Proceedings of the 27th International Conference on Software Engineering (ICSE '05), May 15-21, 2005, ACM, St. Louis, Missouri, USA., pp: 688-689.

Xu, D., Z. Liu, B. Zhu and D. Xing, 2005. Metric Based Software Quality Assurance System. In: Current Trends in High Performance Computing and its Applications, Zhang, W., Z. Chen, R. Glowinski and W. Tong (Eds.). Springer, Berlin, Germany, ISBN:9783540279129, pp: 551-555.

Yau, S.S., Y.W. Wang, J.G. Huang and J.E. Lee, 1990. An integrated expert system framework for software quality assurance. Proceedings of the 14th Annual International Conference on Computer Software and Applications (COMPSAC 90), October 31-November 2, 1990, IEEE, Chicago, Illinois, USA., ISBN:0-8186-2054-4, pp: 161-166.

Yilmaz, C., A.S. Krishna, A. Memon, A. Porter and D.C. Schmidt *et al*., 2005. Main effects screening: A distributed continuous quality assurance process for monitoring performance de gradation in evolving software systems. Proceedings of the 27th International Conference on Software Engineering (ICSE '05), May 15-21, 2005, ACM, New York, USA., pp: 293-302.

Zuser, W., S. Heil and T. Grechenig, 2005. Software quality development and assurance in RUP, MSF and XP: A Comparative Study. ACM SIGSOFT Software Eng. Notes, 30: 1-6.