ISSN: 2366-1313

AN EVALUATION OF THE QUALITY OF SOFTWARE PRODUCTS CREATED BY STARTUPS

POLU NIROSHA

Assistant Professor, Dep of CSE, Narasimha Reddy Engineering College

polunirosha1122@gmail.com

ABSTRACT: *Software firms are seeing exponential expansion because of emergence of new sectors, readily accessible technology, and venture capital investment. The establishment of 476,000 new businesses accounts for over 20% of net monthly job creation in the United States. New businesses are mostly centered in the IT, software services, and e-commerce sectors. This would ultimately result in software startups becoming big economic actors. The future of software companies is promising, which should motivate academics to go into previously undiscovered territory: software engineering (SE) as practiced in startups. "Software startups" create novel, high-tech products with little to no operational experience in the aim of rapidly growing into sectors with plenty of space for growth. There has been no systematic study to increase the performance of software project management for startups, despite past empirical research recognizing that software development procedures for software startups differ from those of traditional software organizations. It is presumed that you understand the why and how of software development techniques. Startup studies were undertaken methodically. Nonetheless, several assessments have taken a business-centric perspective, focusing on open innovation processes and information management practices in organizations as key drivers of startup success. According to past software engineering research, software startups have less labor and information technology resources than established organizations; therefore, they must estimate the number of person-hours or days required to build their standard or customized product. Entrepreneurs must use extreme caution when attempting to estimate the amount of effort required for their ideas. Accurately calculating the effort necessary to construct a software product requires a thorough understanding of its scale.*

KEYWORDS: software Engineering, Software startup, ERP, CMM.

I. INTRODUCTION

The simplest definition of a software process is the processes required to turn a collection of inputs into an output. Inputs and outputs are the fundamental components of any software process. The findings of this study will undoubtedly influence the choices taken by the software development team to enhance the program's quality. Throughout a normal software development process, various factors might impact product quality and project costs. These include testing procedures, project complexity, debugging tools, and programmer or engineer ability. Given this observation, it is evident that each of these components must be evaluated. Previous software engineering [3] literature offered experimental research to establish relevance of these aspects and determine their relationships. The desired level of improvement in knowledge management can only be attained via experimental validation. It would be prudent to verify this hypothesis experimentally rather than relying just on theory, given academics believe that software engineering procedures are fundamentally childish. These tests are often designed to assist the development team in better evaluating, forecasting, and understanding the software development lifecycle and its ultimate result.

In software engineering, an experiment is a set of tests that are performed to validate a hypothesis. It is important to note that framework, model, and method hypotheses are evaluated and validated throughout the method or model development process. The information gathered from this procedure is then used to improve the prior hypothesis, either by suggesting a new one or refining the existing one. In software engineering, statistical process control, or SPC, is a typical approach for assessing what degree of variation is acceptable. Several commercial and industrial industries have utilized the methodologies described above to offer accurate quality evaluation and control. The techniques that emerge from SPC may be used to software processes in the same way that they are used in industrial processes. This would assist to assess software process control and predictability, as well as provide the groundwork for successful software process management and, ultimately, quality. To make the most use of the SPC, one must understand how separate processes influence the overall software quality criteria.

To ensure a high-quality ERP [7] system deployment, the implementation partner, hosting provider, software vendor, and business and technical teams must collaborate closely. This study used software quality indicators gathered from prior studies to a component of a custom enterprise resource planning system. The ISO/IEC 9126 model's quality features and sub-attributes may be assessed using [11] well-defined software quality metrics. The quality of startups' customized items was quantitatively tested in this study using the same criteria.

Following is a list of research questions that served as a guide for the investigation:

1. What kinds of modifications can be made to a software product that is already on the market?

2. How are software metrics used in quality initiatives?

3. What quality models does IT industry use?

4. How do the qualities of COTS software products and COTS products with customization compare?

Because it details the essential development and quality assurance tasks that must be completed throughout the project, the software process is based on the development process. According to the ISO 9126 standard, quality is defined as the total of a product's or service's characteristics that affect its capacity to meet certain standards.

II. REVIEW OF LITERATURE

Software metrics are standard numerical measurements that may be used to assess significance of different software quality aspects [11]. If the software product's quality was evaluated and presented to the development team as feedback, the estimates might be used to highlight areas of software projects that want improvement. According to surveys relevant published literature, studies on the management and control quality of a particular software product are sparse and poor. The software quality measures for ISO/IEC 9126 include function, reliability, usability, efficiency, maintainability, and portability. These metrics include the associated sub-characteristics [8,9]. For an ERP implementation to be successful, software quality is critical. A recent SAP research found that 25% of software ventures fail. Software dependability and inadequate software configuration management are two examples of software quality characteristics impacted

by technology and software tools. To deliver a software solution with the customer in mind, excellent code is essential [8]. In the literature on software engineering, the term "software quality" (abbreviated as "SQ") is defined in a number of ways. [1] define "quality" as "the totality of features and characteristics of a product or service that bear on its ability to satisfy given needs." Therefore, the term "quality" means "the totality of features and characteristics of a product or service that bears on its ability to satisfy given needs." [11] categorize five distinct components of software quality based on their previous research: viewpoints at the following levels: transcendental, manufacturing, product, user, and value-based.

In most circumstances, the literature on software engineering will provide one of two arguments about software quality. A person may look at things from two separate perspectives: internal and external. In this sense, "external quality" refers to how the intended product operates on a certain platform. In contrast, "internal quality" refers to the software development life cycle, including program architecture and complexity level [10]. Efficiency and portability are two of the several stated internal software quality characteristics. External criteria for software quality include usability, utility, reliability, and maintainability [7]. This gives us two more techniques that might be effective in ensuring the newly integrated software product fulfills our quality criteria. There are two approaches for doing this: one examines the quality of the finished output, while the other verifies and validates the software processes and activities used to build it.

You may discover ways for identifying significant quality characteristics in the books. Risk analysis and requirement uncertainty modeling are two of these methods [6]. In addition to the newly revised software architecture, the firm's demands must be addressed while developing software quality models [8]. According to research into the variables that influence software quality, requirements ambiguity during ERP deployment has a substantial impact [9].

III. DEFICIENCIES IN RESEARCH

Although there has been a considerable amount of research on ERP and software quality, a review of the literature suggests that the bulk of these studies focus on qualitative elements of ERP's external software quality characteristics. This is true even though ERP has been studied extensively [4,5]. For the purposes of this study, we employed [2,3] quality metrics to quantitatively assess the quality features and sub-aspects of the

ISO/IEC 9126 model. The 25000 series of standards, often known as SQuaRE, has supplanted the ISO/IEC 9126 standard as the gold standard for software product quality. Both ISO standards evaluate a product using the same set of quality criteria.

Importantly, the ISO/IEC 25010 standard enables the QA team to focus on the three most critical quality attributes of a software product: usability, security, and adaptability. The development process is at a crucial point. This research contributes to the literature in a number of different ways. The first step involves determining the procedure for assessing the ERP product's software quality in its original, unaltered form. We put the software quality measures into action and show you the expected outcomes. Personalizing the ERP software is the next phase in system deployment. Now we're utilizing the same metrics. The ERP installation team may monitor the bespoke ERP's performance against a variety of software excellence indicators. In contrast to earlier research, the data utilized in this study originated from software businesses that were developing and implementing COTS systems. Startups, which are very new firms, vary significantly from established IT service providers. Startups are businesses that are still in their early phases and have yet to withstand the storm of tremendous market and technological uncertainty.

IV. SUGGESTED METHODS

These are some most critical aspects of software development for startups, coupled with tasks such as periodically establishing requirements and testing freshly launched products. A software firm's product may fail due to requirements engineering, customisation approaches, or excessive component reusability, forcing the company to restart its software development cycle to address functionality-related issues. Similarly, competitor software businesses may gain an edge by using comparable customization and requirements engineering tactics. So, it's evident that software companies take shortcuts during requirements engineering to fulfil customers' requests for quicker product delivery or to fit their specific needs, sometimes without considering the repercussions.

One way a corporation may treat criteria is to see them as assumptions about target market and the nature product they would be producing. These assumptions are tested "early and often," despite receiving just a part of the necessary information. However, as more and more businesses offer customers configurable versions of their built-in items, customization has become an unavoidable component of doing business. On other hand, by using these tactics, they may reduce time required to finish the project, so speeding up

ISSN: 2366-1313

the processing of client orders. Software customisation is process of making modifications to an integrated software product so that it may meet a client's specific requirements after deployment, even if those needs are not previously incorporated into product. When customer needs and product requirements are matched, both success rate of software installation and the organizational benefits increase. To achieve such a great fit, bespoke software solution must be able to connect with both the company's processes and the client's special needs.

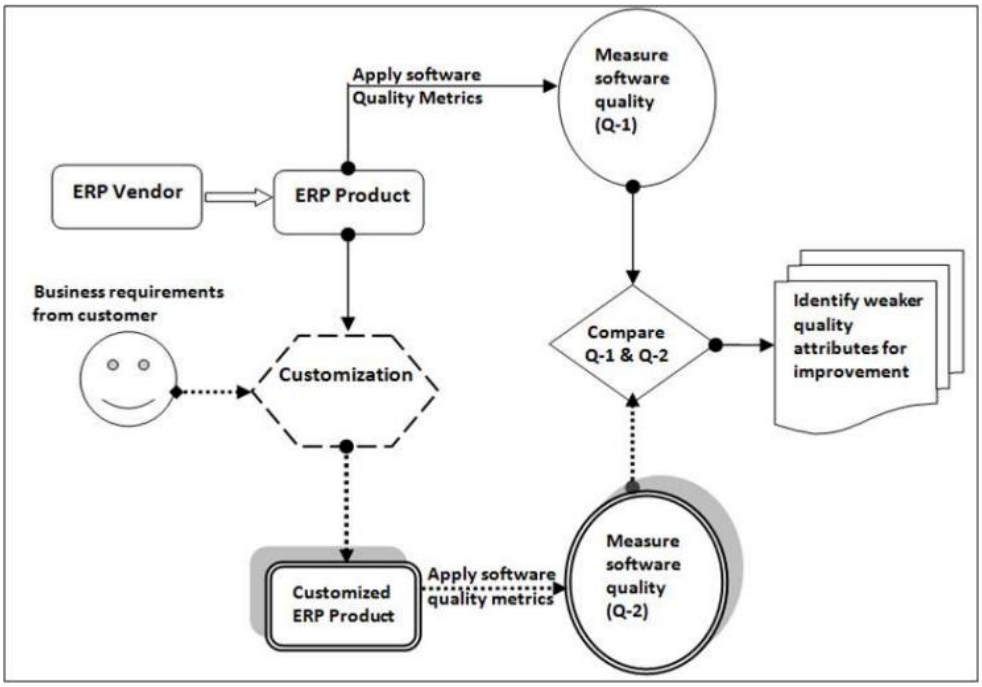

Figure 1 Research framework for quality measurement of customized ERP software

Within the context of this methodology, software quality measurements are employed in two cycles to evaluate how customization affects the quality of the customized ERP under consideration. We want to get a better understanding of the relationship between two via this investigation. The software quality measures are applied to the ERP as soon as it is received from the vendor, and after the first cycle of testing is done, the results are labeled 'Q-1' to show that they were successfully completed. During the second cycle, software quality measurements are applied to a customized ERP product, with the results identified with the 'Q-2' designation and shown. First, the client's business requirements will be collected during the second cycle. Next, the ERP product will be modified, and then the product's quality will be assessed using metrics. The framework makes it easier to compare the outcomes 'Q-1' and 'Q-2', which helps in identifying the quality components

of the program that need improvement and enables the ERP team to concentrate their efforts there in order to achieve a higher level of quality.

V. DATA ANALYSIS & RESULTS

The impact of ERP customisation on quality was examined. The ERP system of a private institution serves as an example of this method in action. A mid-sized software provider named "Dynax" has produced an enterprise resource planning (ERP) package for universities, and its adoption has been studied. Dynax Software is a young company based in India. To small and medium-sized enterprises (SMEs) in India and beyond, it manufactures and distributes commercial off-the-shelf (COTS) products. Every year, Dynax takes on a new clientele of 15 to 20 businesses, both large and small. There was pre- and post-ERP research plan development at client organization. The private institution (PI) that is the subject of this study has 265 faculty members teaching a wide variety of courses to 5,500 students. The school's choice to implement a module-based ERP system was primarily driven by need to streamline Teaching and Learning (TLP), Students' Examination (SE), and Students' Admission (SA) operations. The selected ERP was found to closely match the primary functions institution during first evaluation. Afterwards, in next step, consultants and ERP team members determined that some sub-processes in Teaching and Learning Process (TLP) module needed to be customized. This was in contrast to other two modules, Students' Examination (SE) and Students' Admission (SA), which were scanned for their visibility in selected ERP modules.

Table 2 Software quality measurement of TLP and customized TLP module of ERP

Software Quality Attributes	Sub-Attributes	Software Quality Estimation (in %)	
		TLP Module of ERP	Customized TLP Module of ERP
Functionality (F)	F1 Suitability	86	**95**
	F2 Accurateness	90	**77**
	F3 Interoperability	92	**88**
	F4 Compliance	88	**78**
	F5 Security	94	**74**
	R1 Maturity	89	**95**

Reliability (R)	R2-Fault tolerance	96	**88**
	R3-Recoverability	97	**89**
Usability (U)	– U1 Understandability	88	**96**
	U2 Learnability	79	**71**
	U3 Operability	88	**93**
Efficiency (E)	E1-Time behavior	81	**78**
	E2 Resource behavior	83	**77**
Maintainability (M)	M1 Analyzability	89	**87**
	M2 Changeability	79	**91**
	M3 Stability	89	**76**
	M4 Testability	87	**71**
Portability (P)	P1 Adaptability	97	**90**
	P2 Installability	96	**86**
	P3 Conformance	92	**90**
	P4 Replaceability	**94**	**88**

We gathered data across two cycles, with the support of the Dynax Software Quality Assurance (SQA) team, to analyze suggested research approach for establishing quality of ERP software that was tailored to our specific needs. When it comes to producing and delivering new versions of their products, information technology suppliers must follow certain protocols mandated by software quality standards such as the Capability Maturity Model (CMM) and International Organization for Standardization (ISO) [13]. According to [12], first cycle of this study project included an analysis of the ERP solution's TLP module in relation to previously defined software quality standards. This evaluation was conducted before formal start of system change. A questionnaire including these metrics was sent to every member of vendor's team, including two people from SQA group, three from development, and two from testing. Each of these persons oversaw determining results of each of these metrics together. "Q-1" refers to tabular format in which these findings are presented.

VI. CONCLUSION

According to the study's results, the process of altering the commercial off-the-shelf (COTS) product prior to its usage had a substantial influence on important elements of software quality. Given the scarcity of research on software quality for customized COTS goods, this study connects software quality for customized

COTS products and ERP to demonstrate how software metrics can be used to quantitatively assess the quality of customized ERP. This study's strategy included comparing quality of software for bespoke COTS systems such as ERP. This was done since few studies have been conducted on the quality of software for bespoke COTS items such as ERP. This is something that must be addressed to ensure that the customized ERP solutions given by suppliers meet high standards of quality. According to the results of this research, customization has a significant and high-level link with all quality metrics. The ERP modification has had a detrimental impact on certain sub-attributes, including learnability, time behavior, resource behavior, analyzeability, adaptability, compliance, and replaceability. However, several of the sub-attributes, such as appropriateness, maturity, understandability, operability, and changeability, functioned admirably even after customization. Compliance, accuracy, fault tolerance, and testability are sub-attributes most heavily influenced by the scenario.

REFERENCES

1. Alsaqaf, W, Daneva, M, & Wieringa, R 2019 Quality requirements challenges in the context of large-scale distributed agile: An empirical study Information and Software Technology, vol. 110, no 1, pp. 39-55.
2. Berg, V , Birkeland, J, Nguyen-Duc, A, Pappas, I & Jaccheri, L 2020, Achieving agility and quality in product development - an empirical study of hardware startups , Journal of Systems and Software, vol. 167, no. 110599, pp. 1-15.
3. Bandera, C & Thomas, E 2019, The role of innovation ecosystems and social capital in startup survival , IEEE Transactions on Engineering Management, vol. 66, no. 4, pp. 542-551.
4. Chang, JY , Jiang, JJ, K lein, G & Wang, ET 2019, Enterprise system programs: Goal setting and cooperation in the integration team Information & Management, vol. 56, no. 6, pp. 103-137.
5. Di Martino, S, Ferrucci, F, Gravino, C & Sarro, F 2020, Assessing the effectiveness of approximate functional sizing approaches for effort vol. 123, no. 106308, pp. 1-16.
6. Fotrousi, F, Fricker, SA & Fiedler, M 2018, The effect of requests for user feedback on quality of experience Software Quality Journal, vol. 26, no. 2, pp.

385-415.

7. Giraldo, FD, Chicaiza, Á J, Espana, S & Pastor, O 2021, Empirical validation of a quality framework for evaluating modelling languages in MDE environments Software Quality Journal, vol. 29, no. 2, pp. 275-307.
8. Hsu, C 2017 Complexity and flexibility in information systems design and implementation Journal of Information Technology Theory and Application, vol. 18, no. 2, pp. 1-21.
9. K aczmarek, J & K ucharski, M 2004, Size and effort estimation for applications written in Java , Information and Software Technology, vol. 46, no. 9, pp. 589-601.
10. Olsson, T, Wnuk, K & Jansen, S 2021, A validated model for the scoping process of quality requirements: A multi-case study Empirical Software Engineering, vol. 26, no. 2, pp. 1-29.
11. Prasadu Peddi (2023). AI-Driven Multi-Factor Authentication and Dynamic Trust Management for Securing Massive Machine Type Communication in 6G Networks. International Journal of Intelligent Systems and Applications in Engineering, 12(1s), 361–374.
12. Spender, JC, Corvello, V , Grimaldi, M & Rippa, P 2017, Startups and open innovation: A review of the literature , European Journal of Innovation Management, vol. 20, no. 1, pp. 4-30.
13. Wang, WT, Luo, MC & Chang, Y M 2022, Exploring the relationship between conflict management and transformational leadership behaviors for the success of ERP customization Information Systems Management, vol. 39, no. 2, pp. 177-200.