

Chapter 29

A Customized Quality Model for Software Quality Assurance in Agile Environment

Parita Jain

Amity Institute of Information Technology, Amity University, Noida, India

Arun Sharma

Department of Information Technology, Indira Gandhi Delhi Technical University for Women, Delhi, India

Laxmi Ahuja

Amity Institute of Information Technology, Amity University, Noida, India

ABSTRACT

The agile approach grew dramatically over traditional approaches. The methodology focuses more on rapid development, quick evaluation, quantifiable progress and continuous delivery satisfying the customer desire. In view of this, there is a need for measurement of the agile development process. In this respect, the present research work investigates the inter-relationships and inter-dependencies between the identified quality factors (QF), thereby outlining which of these QF have high driving power and dependence power, working indirectly towards the success of agile development process. This paper proposes a new agile quality model, utilizing an interpretive structural modeling (ISM) approach and the identified factors are classified using Matrice d'Impacts Croise's Multiplication Applique'e a UN Classement (MICMAC) approach. The research findings can significantly impact agile development process by understanding how these QF related to each other and how they can be adopted.

INTRODUCTION

From the past few years, the focus is more on to enhance software development practices by improving reusability, understandability of requirements, software delivery time and cost-effectiveness and many other characteristics. Quality being the most important aspect either in discrete production process or continuous production process, needs to be focused more to deliver a product that is acceptable by customers. According to International Standard Organization ISO 9000, quality is defined as the totality of characteristics of a product as a whole to satisfy the stated and implied needs in accordance with its capability. Here the stated needs mean the requirements that are given by the customer at the time of an agreement, and the implied needs are the needs that are identified by the developers as the necessary needs to be included while developing the product. Definition of quality has been perceived differently by various persons, but the one that has been given by the customer is the definition that counts the most. Quality corresponds to standards, cost of the product, conformance to requirements and value for performance (Juran & Gryna, 2010; Weinberg, 1992).

Overall quality is the much more complicated term than it appears. There are varieties of perspectives for consideration, for example, customer's perspective, developer's perspective, tester's perspective, specification based perspective, manufacturing based perspective, quality assurance based perspective and many more. In every single domain, quality is one of the most important factors for a product to survive in the market. Many more definitions are given by different researchers from the perspective of manufacturing the product and engineering the product respectively. Definition of quality has a contextual bias towards these two industries. A general definition is possible for all the areas; however, when applying to IT software products the context is slightly different. Some of the authors named as Meyer (2000), Pressman (2011), and Sommerville (2015) states' software quality differently but from the same perspective towards software product. The quality of the development process significantly affects the value received by customers and development teams. Hence, for high quality product it's important to concentrate more on development process as development of good quality software is usually an organizational effort, "something of higher quality has more value than of low quality".

Need of Quality Model for Agile

Over the years, developers found that for development the traditional software development methodologies proving to be a, "well defined process for development" work poorly in practice. Moreover, from the literature, it appears that developers are not more interested to adopt traditional processes and are finding ways to reduce efforts to adopt them either intentionally or unintentionally. With the escalation in quality software's, specifying relevant development process is a necessity. Assimilating agile methodology process for development is a good option to counter ever escalating software complexity. Software organizations are ready to accept the quick approach for developing software with the availability of resources, unable to do so due to uncertainty of proving agile process quality. Initially, when introduced agile methodology was applicable on small-scale projects where the methodology proved as a best practice for software development but with the increase in utility the agile methodology also applicable on large-scale projects.

It has been observed that agile methodology is the best practice so far for developing quality software but quantification of quality parameters is still a major challenge. Besides this, in current scenario, expectations of customers, stakeholders and need of quality changed significantly. The agile methodology

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/a-customized-quality-model-for-software-quality-assurance-in-agile-environment/294484

Related Content

State-of-the Art Concepts and Future Directions in Modelling Coordination

Abdelhamid Abdelhadi Mansorand Wan Mohd Nasir Wan Kadir (2014). *Handbook of Research on Emerging Advancements and Technologies in Software Engineering* (pp. 61-80).

www.irma-international.org/chapter/state-of-the-art-concepts-and-future-directions-in-modelling-coordination/108611

A Model Based on Data Envelopment Analysis for the Measurement of Productivity in the Software Factory

Pedro Castañedaand David Mauricio (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 792-818).

www.irma-international.org/chapter/a-model-based-on-data-envelopment-analysis-for-the-measurement-of-productivity-in-the-software-factory/294495

An Empirical Study on Novice Programmer's Behaviors with Analysis of Keystrokes

Dapeng Liu, Shaochun Xuand Huafu Liu (2013). *International Journal of Software Innovation* (pp. 68-87).

www.irma-international.org/article/empirical-study-novice-programmer-behaviors/77619

Analysis of Cognitive Levels of Questions With Bloom's Taxonomy: A Case Study

Ravi Lourdusamy, Poovizhi Magendiranand Clayton Michael Fonceca (2022). *International Journal of Software Innovation* (pp. 1-22).

www.irma-international.org/article/analysis-of-cognitive-levels-of-questions-with-blooms-taxonomy/297922

Managing Software Projects with Team Software Process (TSP)

Salmiza Saul Hamid, Mohd Hairul Nizam Md Nasir, Shamsul Sahibuddinand Mustaffa Kamal Mohd Nor (2012). *Software Process Improvement and Management: Approaches and Tools for Practical Development* (pp. 149-182).

www.irma-international.org/chapter/managing-software-projects-team-software/61214