

Chapter 68

The Influence of the Application of Agile Practices in Software Quality Based on ISO/IEC 25010 Standard

Gloria Arcos-Medina

Escuela Superior Politécnica de Chimborazo, Ecuador; Universidad Nacional Mayor de San Marcos, Perú

David Mauricio

 <https://orcid.org/0000-0001-9262-626X>

Universidad Nacional Mayor de San Marcos, Perú

ABSTRACT

Agile practices are activities or procedures that are applied during the software development process in order to improve its quality and productivity. The objective of this study is to determine the influence of agile practices on software quality. For this purpose, a model composed of 4 groups of agile practices and 8 quality characteristics according to the ISO/IEC 25010 standard has here been proposed. The results of 146 questionnaires addressed to people involved in the software development process show that the application of agile engineering and project management practices have a significant positive influence on the quality attribute functionality. On the other hand, project management practices have a low impact on the quality characteristics of compatibility, portability, security, and usability.

1. INTRODUCTION

While software is essential in all areas of the modern world, software development itself has yet to become a perfect process. Despite efforts to employ software engineering methodologies, software development has not been consistently successful, as evidenced by the high rates of delayed, abandoned, or rejected software projects. Several investigations indicate that the success factors of a project can be linked to the

DOI: 10.4018/978-1-6684-3702-5.ch068

cost, delivery time, scope, and quality (Agarwal & Rathod, 2006; Chow & Cao, 2008; de Wit, 1988). Therefore, lack of quality directly contributes to the failure of a project.

Agile methodologies were disseminated in order to improve software quality and respond more easily to changes. Unfortunately, the results verify that this goal has not yet been achieved and that the lack of quality in software continues to be worrisome. Quality is undoubtedly the most important element of a software project (Xu, 2009). Agile software processes, such as eXtreme Programming (XP) and Scrum, rely on best practices that are expected to improve software development quality. It can be said that best practices aim to introduce software quality assurance (SQA) into a project (Sagheer, Zafar, & Sirshar, 2015).

There are various studies that have contributed to identifying the agile practices and quality characteristics that influence agile development (Bermejo et al., 2014; Gorla & Lin, 2010; Opelt & Beeson, 2008; Subramanyam & Prasad, 2013; Versionone, 2017; Xu, 2009). In addition, research has been carried out to analyze the relationship between agile practices and quality (Bougroun, Zeaaraoui, & Bouchentouf, 2014; Curcio, Malucelli, Reinehr, & Paludo, 2016; Gorla & Lin, 2010; Santos, 2011). In such research, the concept of quality has been analyzed as a whole. However, individual quality characteristics and how agile practices affect each of them have not been considered. That information is vital for project managers to be able to prioritize the practices that contribute to software quality.

In this paper, a conceptual model is introduced to identify the practices that influence the quality characteristics of the software development process by applying agile methodologies. The proposed model consists of four categories of agile practices, eight quality characteristics based on the ISO/IEC 25010 standard (ISO/IEC, 2005), and 13 relationships between agile practices and quality characteristics. The four categories of agile practices that were included are teamwork, project management, engineering, and test. The model was validated through surveys administered to 146 people who implement agile methodologies in the software development process.

The remainder of this paper is organized into 4 sections. Section 2 includes a theoretical framework of agile methodologies, agile practices, and software quality, and it climaxes with the motivation for this work. Section 3 details the proposed model, as well as the strategy employed for the collection of information. Section 4 presents the results of the study. In Section 5, we realize a discussion of the results, and finally, the conclusions of the work are presented.

2. BACKGROUND

2.1 Agile Methodologies and Practices

In February 2001, the term “agile” was applied for the first time to software development. A group of 17 experts from the software industry participated in the meeting where that software term was born. Its objective was to propose the values and principles that would allow teams to quickly develop software that can respond to changes that may arise throughout the project (Muñoz, Velthuis, & Rubia, 2010). After the meeting, The Agile Alliance was created. The Agile Alliance is a non-profit organization dedicated to promoting concepts related to agile software development and helping organizations to adopt those concepts. The starting point was the Agile Manifesto, a document that summarizes the agile philosophy. Although the creators and promoters of the most popular agile methodologies subscribe to the Agile Manifesto and principles, each methodology has its own characteristics and emphasizes one

26 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/the-influence-of-the-application-of-agile-practices-in-software-quality-based-on-isoiec-25010-standard/294525

Related Content

Requirements for the Testable Specification and Test Case Derivation in Conformance Testing

Tanja Toroiand Anne Eerola (2007). *Verification, Validation and Testing in Software Engineering* (pp. 136-156).

www.irma-international.org/chapter/requirements-testable-specification-test-case/30750

A Framework for Understanding the Open Source Revolution

Jeff Elpernanand Sergiu Dascalu (2009). *Designing Software-Intensive Systems: Methods and Principles* (pp. 439-454).

www.irma-international.org/chapter/framework-understanding-open-source-revolution/8244

Advanced Temporal Constraints for Business Processes Modelling and Execution

Eleanna Kafeza (2018). *International Journal of Systems and Service-Oriented Engineering* (pp. 1-25).

www.irma-international.org/article/advanced-temporal-constraints-for-business-processes-modelling-and-execution/231505

Ranking and Risk Factor Scheme for Malicious applications detection and Classifications

Kiran Khatterand Sapna Malik (2018). *International Journal of Information System Modeling and Design* (pp. 67-84).

www.irma-international.org/article/ranking-and-risk-factor-scheme-for-malicious-applications-detection-and-classifications/218172

Designing an Efficient and Scalable Relational Database Schema: Principles of Design for Data Modeling

Rajesh Kanna Rajendranand T. Mohana Priya (2023). *The Software Principles of Design for Data Modeling* (pp. 168-176).

www.irma-international.org/chapter/designing-an-efficient-and-scalable-relational-database-schema/330495