

Chapter 8.6

Agile Software Methods: State-of-the-Art

Ernest Mnkandla

Monash University, South Africa

Barry Dwolatzky

University of Witwatersrand, South Africa

ABSTRACT

This chapter is aimed at comprehensively analyzing and defining agile methodologies of software development from a software quality assurance perspective. A unique way of analyzing agile methodologies to reveal the similarities that the authors of the methods never tell you is introduced. The chapter starts by defining agile methodologies from three perspectives: a theoretical definition, a functional definition, and a contextualized definition. Then an agile quality assurance perspective is presented starting from a brief review of some of the traditional understandings of quality assurance to the innovations that agility has added to the world of quality. The presented analysis approach opens a window into an understanding of the state-of-the-art in agile methodologies and quality, and what the future could have in store for software developers. An understanding of the analysis framework for objectively analyzing and comparing agile methodologies is illustrated by applying it to three specific agile methodologies.

INTRODUCTION

Agile software development methodologies have taken the concepts of software quality assurance further than simply meeting customer requirements, validation, and verification. Agility innovatively opens new horizons in the area of software quality assurance. A look at the agile manifesto (Agile Alliance, 2001) reveals that agile software development is not just about meeting customer requirements (because even process-driven methodologies do that), but it is about meeting the changing requirements right up to the level of product deployment. This chapter introduces a technique for analyzing agile methodologies in a way that reveals the fundamental similarities among the different agile processes.

As for now, there is a reasonable amount of literature that seeks to describe this relatively new set of methodologies that have certainly changed the way software development is done. Most of the existing work is from the authors of the methodologies and a few other practitioners. What lacks is

therefore a more balanced evaluation comparing what the original intents of the authors of agile methodologies were, to the actual things that have been done through agile methodologies over the last few years of their existence as a group, and the possible future applications.

While most of those who have applied agile methods in their software development projects have gained margins that are hard to ignore in the areas of product relevance (a result of embracing requirements instability) and quick delivery (a result of iterative incremental development), some have not joined this new fun way to develop software due to a lack of understanding the fundamental concepts underlying agile methodologies. Hence, this chapter intends to give the necessary understanding by comprehensively defining agile methodologies and revealing how agile methodologies have taken software quality assurance further than traditional approaches. The second concern resulted from more than three years of research into agile methodology practices where the author discovered that the individual agile methods such as extreme programming, scrum, and lean development etc. are not that different from each other. The apparent difference is because people from different computing backgrounds authored them and happen to view the real world differently. Hence, the differences are not as much as the authors would like us to believe. The evaluation technique introduced here will reveal the similarities in a novel way and address the adoption concerns of agile methodologies. This also reveals what quality in an agile context means.

CHAPTER OBJECTIVES

The objective of this chapter is to introduce you to the fundamentals of analyzing agile methodologies to reveal the bare bones of agile development. After reading this chapter, you will:

- Understand three approaches to the definition of agile methodologies (i.e., a theoretical definition, a functional definition, and a contextualized definition).
- Understand the state-of-the-art in agile methodologies.
- Understand the presented framework for objectively analyzing and comparing agile methodologies.
- Understand the meaning of software quality assurance in an agile context.

BACKGROUND

This section will start by defining agile methodologies based on what people say about agile methodologies, what people do with agile methodologies, and what agile methodologies have done to the broad area of software development.

DEFINING AGILE METHODOLOGIES

The agile software development methodologies group was given the name “agile” when a group of software development practitioners met and formed the Agile Alliance (an association of software development practitioners that was formed to formalize agile methodologies) in February 2001. The agile movement could mark the emergence of a new engineering discipline (Mnkandla & Dwolatzky, 2004a) that has shifted the values of the software development process from the mechanistic (i.e., driven by process and rules of science) to the organic (i.e., driven by softer issues of people and their interactions). This implies challenges of engineering complex software systems in work environments that are highly dynamic and unpredictable.

20 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/agile-software-methods/29561

Related Content

Highly-Accurate Temperature-Compensated Pneumatic Measurement Using Piezoresistive Pressure Sensor

Hsiung-Cheng Lin, Wei-Zhi Chen and Wei-Chieh Shih (2022). *International Journal of Software Innovation* (pp. 1-14).

www.irma-international.org/article/highly-accurate-temperature-compensated-pneumatic-measurement-using-piezoresistive-pressure-sensor/309733

Design and Evaluation of Automated Scoring: Java Programming Assignments

Yuki Akahane, Hiroki Kitaya and Ushio Inoue (2015). *International Journal of Software Innovation* (pp. 18-32).

www.irma-international.org/article/design-and-evaluation-of-automated-scoring/133112

Security Testing Framework for Web Applications

Layla Mohammed Alrawais, Mamdouh Alenezi and Mohammad Akour (2022). *Research Anthology on Agile Software, Software Development, and Testing* (pp. 453-479).

www.irma-international.org/chapter/security-testing-framework-for-web-applications/294478

An Approach for the Transformation and Verification of BPMN Models to Colored Petri Nets Models

Said Meghzili, Allaoua Chaoui, Martin Strecker and Elhillali Kerkouche (2020). *International Journal of Software Innovation* (pp. 17-49).

www.irma-international.org/article/an-approach-for-the-transformation-and-verification-of-bpmn-models-to-colored-petri-nets-models/243378

Systematic Model for Decision Support System

Ramgopal Kashyap (2019). *Interdisciplinary Approaches to Information Systems and Software Engineering* (pp. 62-98).

www.irma-international.org/chapter/systematic-model-for-decision-support-system/226396