

## Chapter 36

# Quality Assurance in Agile Software Development

**Iwona Dubielewicz**

*Wroclaw University of Technology, Poland*

**Zbigniew Huzar**

*Wroclaw University of Technology, Poland*

**Bogumiła Hnatkowska**

*Wroclaw University of Technology, Poland*

**Lech Tuzinkiewicz**

*Wroclaw University of Technology, Poland*

### ABSTRACT

*Agile methodologies have become very popular. They are defined in terms of best practices, which aim at developing good quality software faster and cheaper. Unfortunately, agile methodologies do not refer explicitly to quality assurance, which is understood as a planned set of activities performed to provide adequate confidence that a product conforms to established requirements, and which is performed to evaluate the process by which products are developed. The chapter considers the relations of agile practices with software life cycle processes, especially those connected to quality assurance, and tries to answer the question of which agile practices ensure software quality. Next, agile practices associated with quality assurance are assessed from different perspectives and some recommendations for their usage are given. It is observed that modeling has a particular impact on quality assurance.*

### INTRODUCTION

Agile methodologies have become very popular in software companies. In spite of that they do not refer directly to model-driven approach, they help develop software faster and more effectively. Many reports confirm the advantages of an agile approach when applied to small and medium-size scale software projects. Unfortunately, agile methodologies do not refer explicitly to quality assurance. Practitioners of agile methodologies have not yet provided a convincing answer to the question “What is the quality of the software product?”

ISO/IEC/IEEE 24765:2010 (p. 287) provides some general definitions of quality assurance (QA):

*A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.*

DOI: 10.4018/978-1-5225-3422-8.ch036

*A set of activities designed to evaluate the process by which products are developed or manufactured.*

The definitions show that QA is process oriented and focuses on technical requirements (e.g. conformance to standards, procedures, etc.). From this perspective QA may be thought of as a defect-preventing set of activities.

On the other hand, ISO/IEC 12207:2008 (p. 6) defines QA as:

*The planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfill requirements for quality.*

Similar definition of QA is given in KPI Library (<http://kpilibrary.com/store/items/33>):

*Quality assurance is the systematic monitoring and evaluation of the various aspects of a project, service or facility to maximize the probability that minimum standards of quality are being attained by the production process. QA cannot absolutely guarantee the production of quality products.*

*Two principles included in QA are: “Fit for purpose” – the product should be suitable for the intended purpose; and “Right first time” – mistakes should be eliminated. QA includes regulation of the quality of raw materials, assemblies, products and components, services related to production, and management, production and inspection processes.*

These definitions take into consideration not only the technical requirements a product should fulfill but also the requirements for quality, so they have a broader meaning. From this perspective, QA is also thought of as a defect-detecting set of activities.

Sometimes, the activities that focus on defects detection are called Quality Control (QC): “A set of activities designed to evaluate the quality of developed or manufactured product” (ISO/IEC/IEEE 24765:2010, p. 287)).

In the chapter, software quality assurance (SQA) is understood in the broader sense, so this notion includes QC. There are two different purposes for SQA activities (ISO/IEC/IEEE 24765:2010, p. 287): (1) to convince the customer about the quality of the software which he is going to acquire (external perspective) and (2) to support an organization in developing high-quality software (internal perspective)

The first purpose is fulfilled by agile’s approach features e.g. sprint demos, retrospective meetings, acceptance and implementation of proposed changes and so on. In terms of the second aspect, SQA is cited as the one of the software quality management processes (the others two are: quality planning process and quality control process (PMBOK, 2004)). All these well-structured processes, together with some additional elements (procedures, quality standards, documentation), constitute the Quality Management System (QMS) inside the organization. QMS is usually adapted to any new project which is undertaken in an organization (i.e. appropriate quality standards, documentation, procedures are chosen). A QMS includes range of techniques and tools for quality management which are used when performing SQA.

In this chapter we are interested in the internal SQA perspective but at the same time we have restricted our considerations only to the execution phase of the project (i.e. software development) and only to the development team perspective. This means that, for example, QMS planning processes are not in the field of our interest (it is assumed that the quality planning process was finished before the development process started). Hence, the quality management system will not be considered further.

21 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/quality-assurance-in-agile-software-development/188236](http://www.igi-global.com/chapter/quality-assurance-in-agile-software-development/188236)

## Related Content

---

### Requirements Refinement and Component Reuse: The FoReVer Contract-Based Approach

Laura Baracchi, Alessandro Cimatti, Gerald Garcia, Silvia Mazzini, Stefano Puri and Stefano Tonetta (2014). *Handbook of Research on Embedded Systems Design* (pp. 209-241).

[www.irma-international.org/chapter/requirements-refinement-and-component-reuse/116111](http://www.irma-international.org/chapter/requirements-refinement-and-component-reuse/116111)

### Software Component Survivability in Information Warfare

Joon S. Park and Joseph Giordano (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 3381-3390).

[www.irma-international.org/chapter/software-component-survivability-information-warfare/29567](http://www.irma-international.org/chapter/software-component-survivability-information-warfare/29567)

### Optimal Operation of Multireservoir Systems by Enhanced Water Cycle Algorithm

Yanjun Kong, Yadong Mei, Weinan Li, Ben Yue and Xianxun Wang (2019). *International Journal of Software Innovation* (pp. 27-43).

[www.irma-international.org/article/optimal-operation-of-multireservoir-systems-by-enhanced-water-cycle-algorithm/217391](http://www.irma-international.org/article/optimal-operation-of-multireservoir-systems-by-enhanced-water-cycle-algorithm/217391)

### Finding Useful Features for Facial Expression Recognition and Intensity Estimation by Neural Network

Naoki Imamura, Hiroki Nomiya and Teruhisa Hochin (2020). *International Journal of Software Innovation* (pp. 68-84).

[www.irma-international.org/article/finding-useful-features-for-facial-expression-recognition-and-intensity-estimation-by-neural-network/248531](http://www.irma-international.org/article/finding-useful-features-for-facial-expression-recognition-and-intensity-estimation-by-neural-network/248531)

### Enterprise Resource Systems Software Implementation

Ganesh Vaidyanathan (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1780-1797).

[www.irma-international.org/chapter/enterprise-resource-systems-software-implementation/29477](http://www.irma-international.org/chapter/enterprise-resource-systems-software-implementation/29477)