Chapter II
Agile Quality or Depth of Reasoning?
Applicability vs. Suitability with Respect to Stakeholders’ Needs

Eleni Berki
University of Tampere, Finland

Kerstin Siakas
Alexander Technological Educational Institution of Thessaloniki, Greece

Elli Georgiadou
University of Middlesex, UK

ABSTRACT

This chapter provides a basis for discussion and analysis of the quality characteristics of the agile information systems development (AISD) process, and exposes concerns raised regarding the applicability and suitability of agile methods in different organisational and national cultures. The arguments are derived from recent literature reviews and reports on the state-of-the-art in agile methodologies. We also reflect on our own research and experience in three European countries with different academic and work values, and information systems development (ISD) industrial practices based on diverse principles. Examining the different perceptions of quality in the agile software development paradigm by comparing and contrasting them to the quality perceptions in the established ISD methodological paradigms, we aim at: (i) exploring the relationship of traditional, agile, managed, and measured ISD processes, (ii) making the human dynamics that bear on the success and consensus acceptance of IS more explicit, and (iii) establishing a critical framework/approach for reflecting on the suitability and applicability of agile methods in the development and management of quality software systems.
INTRODUCTION

Agile methods have pervaded many application domains of software development and many claim that this has occurred because agile methods (DSDM, XP, Crystal, SCRUM, ...) advocate end-user participation and satisfaction by focusing on systems realisation, requirements change, and testing as the means to achieving a “correct” information system (Beck, 2000, 2003; Beck & Fowler, 2001). On the other hand, it is argued that the agility of these new methods might lead to more complex and not well-documented systems and to a fragmented software development process (Boehm & Turner, 2003a, 2003b; Marciniak, 1994). Our motivation to examine these arguments in this chapter derives from (a) scientific and (b) practical perspectives. Naturally, a fragmented, unpredictable, and non-measurable IS process does not add to the established scientific rules that must guide software development. In that respect, the use of computational principles combined with lightweight methods, which support continuous change, might be the answer for agile quality, particularly for software component-based development in a post-modern information society (Berki, Georgiadou, & Holcombe, 2004; Siakas, Balstrup, Georgiadou, & Berki, 2005b).

Considering, though, the post-technological state of the global software industry, an ISD method customisation to the needs of diverse and different organisational and national cultures points rather to further argumentation for the general applicability and suitability of the agile methods paradigm. There is an obvious need for further research in order to understand the requirements of quality and the requirements of agile quality in particular, within different cultural and social contexts and, perhaps, the need to identify controllable and uncontrollable quality factors for agile ISD (Georgiadou, Siakas, & Berki, 2003). IS quality requires knowledge of different organisational and national cultures on the methods and tools used, on the ways they are used and, most importantly, on the ways people perceive quality and quality assurance (Berki, 2006; Ross & Staples, 1995; Siakas, Berki, Georgiadou, & Sadler, 1997). Awareness and application of total quality management principles and the influence of human involvement and commitment are yet unresolved and largely non-researched issues in different cultural (organisational and national) contexts. Therefore, specialised knowledge is required in order to assess, assure, and certify the quality of agile software development.

It is, yet, questionable if risks and project failures that are frequently encountered in traditional software projects could be diminished or avoided (Siakas, Georgiadou, & Berki, 2005a) by adopting an agile approach. It is, however, argued that agile methods make the key business users a very strong partner in assuring quality (Beck, 2000, 2003; Beck et al., 2001). We presume that in a mature IS society, rather than completely leaving quality to the professionals, agile development projects will perceive the key ISD stakeholders, and end-users in particular, as co-responsible for ensuring that the application fits the purpose. At present, however, one needs to compare and contrast the agile development process (agile methods and the life cycle they support) to traditional methods. Based on the results of comparisons one could, afterwards, analyse, measure, validate, and verify the method suitability and applicability derived from the agile methodology deployment. From the software quality assurance’s point of view, new or/and older software quality properties could be the key attributes for constructing process and product quality principles when both traditional and agile development methods are employed in ISD.

Throughout this chapter, we proceed to a three axis quality examination of the agile methodology paradigm, outlined as follows: The chapter firstly considers a historical perspective and a discussion over process and product quality measurements in traditional software development (method and life cycle models). In order to provide an overview of
Related Content

Formal Semantics of Dynamic Constraints and Derivation Rules in ORM

Software Process Model using Dynamic Bayesian Networks
[www.igi-global.com/chapter/software-process-model-using-dynamic/52889?camid=4v1a](www.igi-global.com/chapter/software-process-model-using-dynamic/52889?camid=4v1a)

Abstract and Concrete Data Type Optimizations at the UML and C/C++ Level for Dynamic Embedded Software
[www.igi-global.com/chapter/abstract-concrete-data-type-optimizations/36338?camid=4v1a](www.igi-global.com/chapter/abstract-concrete-data-type-optimizations/36338?camid=4v1a)

Model-Driven Data Warehouse Automation: A Dependent-Concept Learning Approach
[www.igi-global.com/chapter/model-driven-data-warehouse-automation/78618?camid=4v1a](www.igi-global.com/chapter/model-driven-data-warehouse-automation/78618?camid=4v1a)