

Chapter 8

Agile, Lean, and Service–Oriented Development, Continuum, or Chasm

Juha Rikkilä

Studios 4 Future Software, Free University of Bozen-Bolzano, Italy

ABSTRACT

A fad, hype, and a paradigm shift are often the words that are used about the agile, lean, and now also about service-oriented development. What starts as a step in evolution, grows into a mass movement in Internet and social media, and results in an avalanche of books, training, and consultancy services. Each proponent tries to differentiate from others with extreme statements of own superiority and blames of others' shortcomings. Only a next fad, hype, or paradigm shift seems to be able to override the previous. This chapter looks through the fad, hype, or paradigm shift statements and describes the principles of the agile and lean approaches to the software development. Then it introduces the service orientation that is expected to be the next major shift. If it will overshadow the agile and lean excitement in the software industry, is discussed at the end of the chapter. In addition, this chapter looks into the past in order to find a continuum between these topics till today and to the future. Further some chasms in this continuum are identified, where a new idea has made a major shift and consequently has become a major force in the field. The service orientation in the software development is in the early phase of its lifecycle. The question is: will it still go through some chasm until it settles for large use, or is it already through all of adaptations and ready to be the next wave of evolution, or the next fad, hype and paradigm shift in software industry? The last part of this chapter proposes one more adaptation that creates continuum from the agile and lean approaches but brings up also the revitalization of architecting and design methodologies.

DOI: 10.4018/978-1-4666-4301-7.ch008

INTRODUCTION

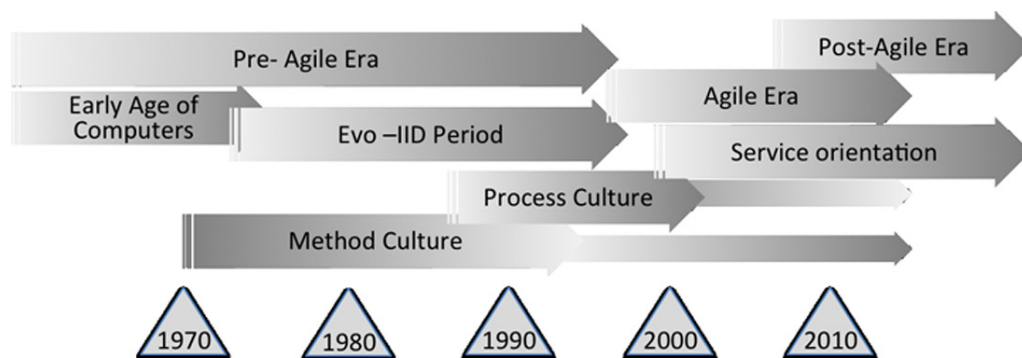
This first chapter of the book introduces the agile, lean and service-oriented approaches, and links them backwards into the history of the software development. This looking backwards enables readers to see how much of new thinking each evolutionary step contains and what radical shifts they introduce. And also see through the inevitable fad¹ and hype² that is included in the discussion of the new and existing things. Further this introduction helps the reader to assess the real existence and size of the possible paradigm shifts³ in this evolution. Understanding the technology adoption life cycle⁴ and the position on it of each of the approaches presented here gives basis to assess the stability or expected change of each.

However, the software engineering is not so much in the domain of natural science than it is on that of the social sciences (Runeson & Höst, 2009). This means that though many technology and engineering characteristics are applicable in technology adoption discussion, the overall view is that of social sciences. So a definite analysis of all aspects of agile, lean and service orientation cannot be done in technical terms and one single truth of the best or most feasible cannot be determined. Instead, multi paradigm nature of social science suggests that several paradigms or patterns can be found and have applicability in any defined scope (Bryman, 2000).

This chapter is ordered according to timeline where evolution of the agile approach sets the milestones, see Figure 1. However, as the study of the past indicates more evolution than revolution the milestones are not accurate. Though there are also some exact occasions, like the publishing of the agile manifesto or the book of the lean principles in manufacturing, they are not considered sufficient to characterize the life cycle of these approaches. Yet, they do have significant impact in the promotion of these approaches. Timelines are, by definition, always to some extent subjective. The exact dates of the events cannot be determined from the literature and the interpretations of dependences between different events and evolutionary paths may be differently viewed depending of the viewpoint. In this chapter, the proposed timeline is predominantly based on works of Larman and Basili (2003) and the author's experience in the industry since the 1970's, first in a multinational computer manufacturer and supplier, and later on in a multinational telecom vendor.

The timeline specifies the pre-agile, the agile, and the post-agile eras. The beginning of the pre-agile era is in the time when the first computers appeared and the first programs were made. The first period of the era, the early age of computers, is characterized by the batch computing, and writing of single programs. That lasted till 1970's. The second period of the pre-agile era is called Evo-IID since the debate of the evolutionary (Gilb,

Figure 1. The timeline of the agile evolution



30 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-lean-service-oriented-development/77703

Related Content

A Hybrid System for Heart Disease Diagnosis Based on HPCBE Method

Pooja Rani, Rajneesh Kumar and Anurag Jain (2022). *International Journal of Software Innovation* (pp. 1-14).

www.irma-international.org/article/a-hybrid-system-for-heart-disease-diagnosis-based-on-hpcbe-method/303582

Ontological Rules for UML-Based Conceptual Modeling: Design Considerations and a Prototype Implementation

Shan Lu and Jeffrey Parsons (2011). *International Journal of Information System Modeling and Design* (pp. 24-44).

www.irma-international.org/article/ontological-rules-uml-based-conceptual/51577

Masters of Imagination: From Hierarchies to Connected Swarms

Jaap van Till (2021). *Handbook of Research on Software Quality Innovation in Interactive Systems* (pp. 83-101).

www.irma-international.org/chapter/masters-of-imagination/273566

Implications of Markup on the Description of Software Patterns

Pankaj Kamthan (2011). *Knowledge Engineering for Software Development Life Cycles: Support Technologies and Applications* (pp. 136-160).

www.irma-international.org/chapter/implications-markup-description-software-patterns/52881

Multichannel Service Delivery Architecture: A Case Study

Randall E. Duran and Anh Duc Do (2015). *Handbook of Research on Innovations in Systems and Software Engineering* (pp. 589-601).

www.irma-international.org/chapter/multichannel-service-delivery-architecture/117942