



Chapter IV

Lean, Light, Adaptive, Agile and Appropriate Software Development: The Case for a Less Methodical Methodology

John Mendonca
Purdue University, USA

Jeff Brewer
Purdue University, USA

ABSTRACT

Historically, the approach to software engineering has been based on a search for an optimal (ideal) methodology — that is, the identification and application of a set of processes, methods and tools that can consistently and predictably lead to software development success. This chapter presents the basis for pursuing a more flexible and adaptive approach to methodology. Less methodical methodologies, under a variety of names, take a contingency-oriented approach. Because of the limitations in the nature of methodology, the high failure rate in software development, the need to develop methodology within an environmental context and the pressures of fast-paced “e-development,” the authors argue that further exploration and definition of an adaptive, contingency-based approach to methodology is justified.

INTRODUCTION

Despite the high rate of failure in software development, the fundamental strategy for achieving quality in software engineering continues to be methodology — that is, discovery and application of that ideal set of processes and practices that lead to software products that are accurate, effective, delivered on time and within budget. The path to an optimal methodology leads theorists and practitioners toward increasingly refined sets of concepts, models, rules, project management strategies, descriptions of deliverables, tools, testing standards, test-case constructs and the many other components of a well-defined methodology. Perhaps because of its close identity with the “engineering” paradigm, ubiquitous failure seems not to have shaken faith in the methodical approach to software development. In fact, the response to failure seems often to be more methodology.

In recent years, due to the increasing complexity of the information technology (IT) arena and the furious pace of e-commerce and e-business development, a less methodical approach to software development management has gained attention. This approach has often been linked with Extreme Programming (XP) and has been called by a variety of names, including “lean” and “light” methodology (Yourdon, 2000b). Highsmith (2000) used the term “adaptive” in his book describing the basic concepts, but he and others prominent in XP theory and practice seem to have settled on “agile” as the preferred term. Earlier this year, with the support of XP proponents and others, the “Manifesto for Agile Software Development” (2001) was developed and published.

Regardless of the name, the approach embodies two characteristics. The first characteristic is that it is less methodical. It is not fixated on the search for an optimal methodology but is contingency oriented, allowing for adaptation and flexibility depending on environmental issues. The second characteristic is that it incorporates a concept of appropriateness. A methodology must not only adapt to its environment, it must also reflect an appropriate level of rigidity, the “just-right” level between no methodology and a heavily restrictive one that suffocates rather than informs.

This paper argues that because of the inherent limits to methodology, unrealized expectations and the fast-paced, complex and unpredictable environment, a less methodical contingency approach to software engineering is justified.

METHODOLOGY: EXPECTATIONS AND LIMITATIONS

As noted above, a software development methodology is a set of processes and techniques for the management of software development. The numerous formal documented methodologies and many more informal ones vary based on the many paradigms and variables that are part of the software development landscape. Ivaria, Hirschheim and Klein (2000/2001) suggest there are more than 1,000 information systems development methodologies and offers a schema for their characterization

9 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/lean-light-adaptive-agile-appropriate/28109

Related Content

On the Identification of Modeler Communities

Dirk van der Linden, Stijn J.B.A. Hoppenbrouwers and Henderik A. Proper (2014). *International Journal of Information System Modeling and Design* (pp. 22-40). www.irma-international.org/article/on-the-identification-of-modeler-communities/112040

Graph-Based Spam Image Detection for Mobile Phone Spam Image Filtering

So Yeon Kim and Kyung-Ah Sohn (2015). *International Journal of Software Innovation* (pp. 72-86). www.irma-international.org/article/graph-based-spam-image-detection-for-mobile-phone-spam-image-filtering/133116

Neural Network Control of a Laboratory Magnetic Levitator

J. Katende and M. Mustapha (2013). *Integrated Models for Information Communication Systems and Networks: Design and Development* (pp. 361-374). www.irma-international.org/chapter/neural-network-control-of-a-laboratory-magnetic-levitator/79673

Reengineering Structured Legacy System Documentation to UML Object-Oriented Artifacts

Terrence P. Fries (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 749-771). www.irma-international.org/chapter/reengineering-structured-legacy-system-documentation/77731

Method Using Command Abstraction Library for Iterative Testing Security of Web Applications

Seiji Munetoh and Nobukazu Yoshioka (2015). *International Journal of Secure Software Engineering* (pp. 26-49). www.irma-international.org/article/method-using-command-abstraction-library-for-iterative-testing-security-of-web-applications/136451