

Chapter 11

A Framework for Analyzing Structural Mechanisms Deployed to Support Traditional and Agile Methods: Making Sense of “Democratization” in the Software Development Workplace

Michal Dolezel

 <https://orcid.org/0000-0002-5963-5145>

University of Economics, Prague, Czech Republic

Alena Buchalceva

 <https://orcid.org/0000-0002-8185-5208>

University of Economics, Prague, Czech Republic

ABSTRACT

People rely on structures to make their worlds orderly. This chapter conceptually probes into the problem of the differences between organizational structures deployed in traditional and agile environments. The authors develop an argument that all common forms of organizational entities can be classified by involving a two-dimensional classification scheme. Specifically, they constructed a typology to examine the issues of formal vs. informal authority, and disciplinarity vs. cross-functionality in terms of their significance for traditional and agile software development workplaces. Some examples of concrete organizational forms—including traditional project team, independent test team, self-organizing agile team and developers’ community of practice—are discussed. In sum, they argue that by employing this classification scheme, they can theorize the nature of the on-going structural shift observed in conjunction with deploying agile software development methods. They acknowledge that the structures have fundamentally changed, terming the move “democratization” in the software development workplace.

DOI: 10.4018/978-1-7998-4165-4.ch011

INTRODUCTION

Over the last two decades, Agile Software Development Methods (ASDMs) have rapidly grown in popularity. While ASDMs were originally coined by a group of “organizational anarchists” (Fowler & Highsmith, 2001) who wanted to “undo the damage that waterfall had done to ... [the software development] profession” (Schwaber, 2013), the methods seem to be nowadays widely accepted as *the* mainstream way of software development. The shift from the plan-driven methods to ASDMs can be characterized as a move from the rigorous and well-defined processes towards rapid, people-oriented and customer-centered software practices (Cohn, 2010). Indeed, the shift was supposed to also bring some fundamental changes in the overall philosophy and value-orientation with regard to software development activities. These changes were exemplified in the Agile manifesto and its twelve principles (Fowler & Highsmith, 2001). However, despite the huge attention which the Agile manifesto attracted in the past decade, the question whether it is yet making a real, on-going impact remains open. In our view, this may be due to the essential fluidity of the Agile manifesto, despite the fluidity being intentional (Hohl et al., 2018).

In contrast to being fluid and uncertain, in many aspects of their everyday lives people naturally tend to look for tangible guiding principles and structures. In that sense, social psychologists argue that “structure provides people with the means to achieve their desired ends”, as “[w]hen the world is orderly and structured, people can make sense of events” and predict the future (Laurin & Kay, 2017). Notably in the world of work, structures play an important role. In this context, a structure means a generic mechanism allowing workers to get their work done. In particular, organizational structures, which are typically over-simplistically represented by corporate organizational charts, are frequently used in practice and examined by management scholars (Daft et al., 2010; Pugh et al., 1969). It is reasonable to expect that software teams are no exception in their need for a structure.

Apparently, due to ASDMs in general, and scaled agile frameworks in particular, the commonly used approaches to organizational structuring within the software world have changed during the last decade or so. It appears that software practitioners hold that the structures which previously worked in plan-driven settings (e.g. silo-ed functional teams) are now partly or fully dysfunctional in agile settings (Noordeloos et al., 2012). Also, emerging structuring mechanisms such as cross-functional teams, communities of practice (Cohn, 2010), agile centers of excellence (Knaster & Leffingwell, 2019), agile chapters and guilds (Smite et al., 2019) etc. have appeared in the organizational repertoires of software development enterprises. One of key drivers for the adoption of these structures seem to be the growing popularity of scaled agile frameworks. These frameworks now play a dominant role by demonstrating how to conduct and structure software development activities in an agile manner at a large scale (Ebert & Paasivaara, 2017; Kalenda et al., 2018). In some of these frameworks, it is possible to find a prescription of the role delineation to be implemented (Smite et al., 2019), which is an important structural aspect of modern organizations (Daft et al., 2010).

In sum, we believe that the issues such as how practitioners structure the fluid world of ASDMs, and whether they do so at all, are of great practical and theoretical relevance. Moreover, the current “restructuring” trend, i.e. the move from certain type of structures to others, is an interesting phenomenon *per se*. In particular, a deeper understanding of the emerging agile organizational structures may help software leaders and managers to understand the nature of the shift from traditional to agile methods. Also, it may help them to choose an appropriate structure by finding a fit between their context and available organizational options. In addition, we propose that the present theory-founded analysis can stimulate further research on this currently under-researched topic. Therefore, the question we pose herein is:

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/a-framework-for-analyzing-structural-mechanisms-deployed-to-support-traditional-and-agile-methods/259179

Related Content

An Image Processing and Machine Learning Approach for Early Detection of Diseased Leaves

Sowmya B.J., Chetan Shetty, S. Seemaand Srinivasa K.G. (2019). *International Journal of Cyber-Physical Systems* (pp. 56-73).

www.irma-international.org/article/an-image-processing-and-machine-learning-approach-for-early-detection-of-diseased-leaves/247483

Migration Goals and Risk Management in Cloud Computing: A Review of State of the Art and Survey Results on Practitioners

Shareeful Islam, Stefan Fenz, Edgar Weippland Christos Kalloniatis (2016). *International Journal of Secure Software Engineering* (pp. 44-73).

www.irma-international.org/article/migration-goals-and-risk-management-in-cloud-computing/160712

An Ontology Based Representation of Software Design Patterns

Jens Dietrichand Chris Elgar (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 528-545).

www.irma-international.org/chapter/ontology-based-representation-software-design/29408

Assisting Learners to Dynamically Adjust Learning Processes Through Software Agents

Weidong Panand Igor Hawrysiwycz (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1345-1357).

www.irma-international.org/chapter/assisting-learners-dynamically-adjust-learning/29449

Fault Recognition for Mechanical Arm by Using Relative Margin SVM

Dongzhe Yang (2022). *International Journal of Information System Modeling and Design* (pp. 1-10).

www.irma-international.org/article/fault-recognition-for-mechanical-arm-by-using-relative-margin-svm/313576