

Chapter 5

Managing Contradictory Tensions in Agile Distributed Development: An Ambidextrous Approach

Ashay Saxena

Indian Institute of Management, Bangalore, India

Shankar Venkatagiri

Indian Institute of Management, Bangalore, India

Rajendra K Bandi

 <https://orcid.org/0000-0002-1872-3587>

Indian Institute of Management, Bangalore, India

ABSTRACT

Increasingly, agile approaches are being followed in a distributed setup to develop software. An agile approach is characterised by the need to regularly welcome change requests and update the software artefact accordingly whereas distributed teams prefer to work towards following a plan to fulfil project objectives defined upfront. This results in contradictory tensions when agile is practised with teams operating in a globally distributed format. This chapter focuses on exploring the central conflict and discuss approaches to manage the conflicting forces in an agile distributed development setup. Furthermore, it presents an industry case study to provide more clarity on conflict management in such settings.

INTRODUCTION

Over the last decade, agile approaches have become the *de facto* standard owing to their promise of iterative and incremental software development. Unlike traditional approaches, Agile recognizes the need for software requirements to evolve with and thrive on frequent changes to the product definition

DOI: 10.4018/978-1-7998-4165-4.ch005

(Paetsch et al., 2003). In essence, agile offers *flexibility* to continuously update the software artefact and deliver relevant functionality, with work being coordinated by self-managing teams (Boehm & Turner, 2003; Moe et al., 2010). Heretofore, this necessitated the co-location of the software team, a mode which promoted extensive face-to-face communication so that developers could collaborate efficiently on project-related tasks.

To meet the demands of globalized development, software teams are increasingly operating in a distributed setup. Their main objective is to develop high-quality software in multiple time-boxed iterations by optimizing resources that are available globally (Jiménez et al., 2009; Shrivastava & Date, 2010). In direct contrast to flexibility, teams working in a distributed setup seek *stability* to predictably meet the objectives of a project plan. They prefer to operate with clarity on what needs to be delivered upfront in order to ensure that the project progresses in a smooth manner.

There is, therefore, a conflict between flexibility and stability when the promise of agile meets the reality of a distributed setup.

In an Agile Distributed Development (ADD) model, projects are implemented by teams located across multiple geographies and following an agile approach (Ramesh et al., 2012). Depending upon the contingencies presented by different variants of ADD setup, which are described in the next section, the conflicting forces between flexibility and stability surface in a nuanced form. A holistic view of the contingencies experienced by software teams in varied setups has received scant attention within existing literature (Jalali & Wohlin, 2011). This chapter elaborates on the potential drivers of the flexibility-stability conflict across different types of ADD setup.

Dealing with such a conflict is not unique to the ADD setting. Researchers (e.g., Gibson & Birkinshaw, 2004) have examined other domains that involve demands of similar nature across an entire business unit. Invariably, organizational settings have demonstrated the need for the simultaneous pursuit of trade-off decisions rather than strictly selecting one over another. In this regard, the capability of organizational ambidexterity provides guidance in managing the conflicting forces. This chapter dwells on the potential drivers of ambidexterity, which enable teams to manage the tensions posed in the ADD setting. We also present findings from a case study to provide evidence from the field.

BACKGROUND

This section outlines the basic principles of agile software development and the fundamentals of distributed teamwork. We focus on the foundational aspects of agile approaches (i.e., theoretical underpinnings) and the constraints induced when teams operate in a distributed setup. These elements are essential to uncover conflicting experiences in ADD settings.

Agile Software Development

Agile practices picked up pace since the enunciation of a manifesto in 2001. The academic community began to take note of this new approach towards developing software starting 2005, and their contributions have steadily increased since then (Dingsøyr et al., 2012). Over the last decade, the domain of agile software development (ASD) has generated a significant amount of literature uncovering theoretical underpinnings, among many other themes (Hoda et al., 2018).

12 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/managing-contradictory-tensions-in-agile-distributed-development/259173

Related Content

Automated Ranking of Relaxing Query Results Based on XML Structure and Content Preferences

Wei Yan, Li Yanand Z. M. Ma (2013). *Mobile and Web Innovations in Systems and Service-Oriented Engineering* (pp. 44-62).

www.irma-international.org/chapter/automated-ranking-relaxing-query-results/71990

Multiple Multimodal Mobile Devices: Lessons Learned from Engineering Lifelog Solutions

Daragh Byrne, Liadh Kellyand Gareth J.F. Jones (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 2014-2032).

www.irma-international.org/chapter/multiple-multimodal-mobile-devices/77788

Dealing With Noise and Partial Volume Effects in Alzheimer Disease Brain Tissue Classification by a Fuzzy-Possibilistic Modeling Based on Fuzzy-Genetic Initialization

Lilia Lazliand Mounir Boukadoum (2019). *International Journal of Software Innovation* (pp. 119-143).

www.irma-international.org/article/dealing-with-noise-and-partial-volume-effects-in-alzheimer-disease-brain-tissue-classification-by-a-fuzzy-possibilistic-modeling-based-on-fuzzy-genetic-initialization/217396

Predicting Bug Priority Using Topic Modelling in Imbalanced Learning Environments

Jayalath Bandara Ekanayake (2021). *International Journal of Systems and Service-Oriented Engineering* (pp. 31-42).

www.irma-international.org/article/predicting-bug-priority-using-topic-modelling-in-imbalanced-learning-environments/272543

Temporal Interaction Diagrams for Multi-Process Environments

T. Y. Chen, Iyad Rahwanand Yun Yang (2003). *Practicing Software Engineering in the 21st Century* (pp. 143-155).

www.irma-international.org/chapter/temporal-interaction-diagrams-multi-process/28115