

Chapter VI

IT Value of Software Development: A Multi-Theoretic Perspective

VenuGopal Balijepally

Prairie View A&M University, USA

Sridhar Nerur

University of Texas at Arlington, USA

RadhaKanta Mahapatra

University of Texas at Arlington, USA

ABSTRACT

Software development in organizations is evolving and increasingly taking a socio-technical hue. While empirical research guided by common sense reasoning has informed researchers and the software community in the past, the increasing social character of software development provides us with the context and the motivation to provide theoretical underpinnings to our empirical work. In this chapter we sample three theoretical domains that could serve our empirical research efforts: social capital, organizational learning and knowledge based view of the firm. We illustrate the utility of these theoretical perspectives by articulating a research model that captures the IT value created by software development teams practicing different methodologies.

INTRODUCTION

The field of software development has undergone significant changes in recent years. A hyper-competitive business environment characterized by change and uncertainty has prompted the software community to evolve new ways of building software. The emerging methodologies follow an evolutionary delivery model (Gilb, 1987) that allows developers to adapt to changing requirements. This is counter to the traditional plan-driven approach that uses a linear process in accomplishing the tasks involved in software development. In such an approach, an enormous amount of time and effort are spent in gathering requirements and evolving specifications with a view to preparing for all foreseeable changes during the lifecycle of the product. In contrast, emerging approaches such as agile methodologies rely on short iterative cycles with continuous stakeholder feedback and frequent planning to cope with and leverage change (Nerur, Mahapatra, and Mangalaraj, 2005).

Agile methodologies have attracted a lot of attention in the recent past. This new approach differs considerably from traditional software development in many ways. Foremost, there is an increased emphasis on self-organizing teams that enjoy greater latitude in goal-setting and decision-making with regard to setting priorities, deadlines, etc. Team members are encouraged to play multiple roles, such as developer, architect and analyst. A shared understanding and vision of the evolving product is facilitated by practices such as joint code ownership, role rotation and reflection workshops. This is in contrast to specialized roles assigned to developers in traditional methodologies. In the agile approach, specifications evolve through constant dialogue and feedback between developers and customers, while in the traditional approach, extensive specifications gathered upfront from customers guide the development process. Thus, there is a perceptible shift from a hierarchical, process driven, and command-and-

control based approach to one that emphasizes people-orientation, collaboration, and leadership (Highsmith, 2003).

The changes in methodologies highlight the underlying transformation of software development from a technical enterprise to a more socio-technical endeavor. The lack of theory-driven empirical work in software development may be attributed to the emphasis that was placed on technical aspects of software development. Theoretical grounding of empirical research is still not considered an essential requirement, especially in the software engineering domain. Though there is some evidence of increasing awareness of theoretical issues (Hannay, Sjoberg, and Dyba, 2007; Nerur and Balijepally, 2007), the dominant thinking is predicated on the primacy of common sense reasoning over generalizable theory (Lindblom, 1987). While use of theory is taken for granted in various business disciplines, including several areas of IS research, software development research is still grappling with the issue of whether theory should be used (Hannay et al., 2007). The current social “makeover” of software development provides us with the context and the opportunity to refocus empirical research in software development towards theory building and testing. The centrality and importance accorded to teams and collaboration among team members by emerging methodologies affords an opportunity to draw on the extensive body of knowledge in organizational and management theory.

The primary objective of this chapter is to demonstrate the applicability of theoretical perspectives in software development research. One of the critical problems that confront software managers today is the choice of methodologies. This poses an interesting research problem that can be theoretically investigated. Keeping this in mind, we articulate three theoretical streams, namely social capital, knowledge-based view of the firm, and organizational learning. Traditionally, human capital is considered as the main

13 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/value-software-development/4294

Related Content

A Novel Approach to Managing the Dynamic Nature of Semantic Relatedness

Youngseok Choi, Jungsuk Oh and Jinsoo Park (2016). *Journal of Database Management* (pp. 1-26).

www.irma-international.org/article/a-novel-approach-to-managing-the-dynamic-nature-of-semantic-relatedness/165160

Criteria for Comparing Information Modeling Methods: Informational and Computational Equivalence

Keng Siau (2005). *Advanced Topics in Database Research, Volume 4* (pp. 238-254).

www.irma-international.org/chapter/criteria-comparing-information-modeling-methods/4377

Modeling Fuzzy Information in the IFO and Relational Data Model

Z. M. Ma (2006). *Advanced Topics in Database Research, Volume 5* (pp. 273-293).

www.irma-international.org/chapter/modeling-fuzzy-information-ifo-relational/4397

Scalable Data Mining, Archiving, and Big Data Management for the Next Generation Astronomical Telescopes

Chris A. Mattmann, Andrew Hart, Luca Cinquini, Joseph Lazio, Shakeh Khudikyan, Dayton Jones, Robert Preston, Thomas Bennett, Bryan Butler, David Harland, Brian Glendenning, Jeff Kern and James Robnett (2014). *Big Data Management, Technologies, and Applications* (pp. 196-221).

www.irma-international.org/chapter/scalable-data-mining-archiving-and-big-data-management-for-the-next-generation-astronomical-telescopes/85457

CORBA-Based Run-Time Architectures for Workflow Management Systems

J. A. Miller, A. P. Sheth, K. J. Kochut and X. Wang (1996). *Journal of Database Management* (pp. 16-27).

www.irma-international.org/article/corba-based-run-time-architectures/51159