

Analyzing the enablers and challenges for Successful Methodological Transition

Jagpuneet Bajwa
Punjabi Uni. Regional Centre,
Mohali

Kawaljeet Singh
University Computer Center
Punjabi University, Patiala

Neeraj Sharma
Dept. of Computer Science
Punjabi University, Patiala

ABSTRACT

Agile methods are gaining popularity over a wider range of domains, predominantly in enterprise environments. Software methodologies are evolving due to the challenges imposed by ever changing software development scenarios. Agile methodologies particularly demand for an organizational culture which focuses on high responsibility and greater discipline of every individual in an organization. The transition from traditional methodology to agile is an overwhelming task and poses greater challenge for organizations which are striving for better customer satisfaction and producing higher quality products. This paper investigates the underlying reasons behind this 'methodological shift', discusses about the contrasting differences between the traditional or conventional software development methods and light weight or agile methodologies. Further in this paper we discuss the present need for methodological transitions. Third section provides an insight of 'preconditions' (core reasons), 'enablers' (changes required) and 'challenges' (risks involved) for a successful methodological transition in the form of a proposed framework.

General Terms

Software engineering, Software development methods (SDM), Methodological transition.

Keywords

Agile methods, XP, Scrum, DSDM, spiral model, waterfall model.

1. INTRODUCTION

Software development strategies/methods (SDM) have been an integral part of Software engineering from the last five decades. The credibility of these methods has always been a debatable issue. The proponents of traditional "Waterfall-type" methodologies of development consider "Light-Weight" methodologies such as "Agile" a 'non realistic hype'. But 'Agilators' (people who practice agile) believe that traditional methodologies are not able to match up the pace with rapidly changing technological and business requirements.

The process of improving the existing methodologies attained greater pace since the advent of WWW and Internet based applications. The spread of e-commerce brought in architectures that demanded software processes for supporting customer services at internet speed. Gradually, for software methodologists, it became imperative to shorten process

lifecycle for rapid product development. As a related consequence, business became more 'customer-centric'. Today, business houses have become more competitive as customer requires more services in lesser time and cost. Many organizations are sensing this business need and are transforming their traditional software development methods into more flexible, change oriented Agile Methods.

2. RELATED LITERATURE

'Software Crisis' of 1960's lead to an era which recognized the need for change in existing SDMs and acted as a stimulant in further development of numerous process models. 1970's saw the rise of 'Iterative enhancement' Model (Basili and Turner, 1975) which was a step towards iterative development. Major turning point in SDM evolution came in 1987. This year a report was published by 'The Defense Science Board Task Force on military Software' [5]. It created debate about the credibility of 'Traditional Software Process Models' as these process models were not able to accommodate the most useful and effective software development approaches such as 'Reuse' and 'Prototyping'. Also this report acted as a catalyst in the advancement of existing process models into much improved ones. The classic example was of 'Spiral Model' [Barry Boehm]. Boehm advocated the use of such process models which were risk-driven, capable of handling complex development projects and less document driven [2]. This was in contrasting comparison with the 'Waterfall Model' (Royce, 1987).

In 1995, Standish Group published a survey report. This report highlighted some hard facts about software industry. It is also called 'Chaos Report' as it revealed facts about the non completion of software projects in the United States mainly due to cost overruns and unclear user requirements [26]. These facts presented a grim picture of software development. But the fact is many projects are still failing. In spite of great planning and detailed preparations, projects still suffer from cost and time overruns.

Facing this daunting problem, the Standish Group gave some possible solutions for software development. *Standish Report* [26] further suggested some features to aid software development such as smarter time frame, suggested simple tools for Management understanding, laid stress on more and more customer involvement, smaller milestones, open work

culture and also advocated the concept of 'Growing Software' as opposed to developing software.

The changing business created need for researchers to experiment with alternative methodologies which could deliver software rapidly and had inherent simplicity. Software methods offering faster delivery in rapidly changing user requirements came into practice. For instance, *Dynamic systems development method* (DSDM, 1994), *Scrum development process* (Schwaber, 1995; Schwaber and Beedle, 2001), *Crystal family of methodologies* (Cockburn, 1998;2001), *Extreme programming g (XP)* (Beck, 1999), *Adaptive software development* (ASD) (Highsmith, 2000), Rational Unified Process (RUP) (Kruchten, 2000), *Feature-Driven Development* (FDD) (Palmer and Felsing, 2002), *Pragmatic Programming* (Hunt and Thomas, 2000) have emerged during the last fifteen years .

In 2001, seventeen software process methodologists signed a manifesto called '*Agile Manifesto*'. The above mentioned methods contributed towards the framing up of this manifesto and a new methodology called 'Agile Methodology'. Agile denotes "dexterity in motion, readiness for motion, the quality of being agile, nimbleness, and activity" (Oxford Advanced Learner's Dictionary). Software projects using agile methodologies report efficient handling of software development risks of cost overrun, extended time frame, failure to meet needs [28]. This methodology works efficiently if it gets conclusive environment.

Agile software development processes depict a drift from 'heavy weight' document driven software processes to 'light weight' processes. Glen B. Alleman differentiates between 'light weight' and 'Agile'. He explains that 'light weight' and 'Agile' are not interchangeable concepts as the term 'light weight' describes the non-requirement of those artifacts insignificant with respect to the final software product [7]. Abrahamsson et al. defines 'agile' to be an answer to business needs of ever growing Internet and mobile software applications [17].

3. TRADITIONAL AND AGILE SOFTWARE DEVELOPMENT

Traditional methods also called 'plan driven' methods differ from agile methods at methodology level. In traditional methods supports rigid structural frameworks where focus is on processes which require formal/ heavy documentation, upfront planning and supports very less customer to team and team to team interaction. Agile methods are in stark contrast to traditional methods as the focus is on improving quality of ongoing projects on daily basis. This accentuates the need to have a flexible process framework which relies on less process ceremonies, face to face communications, less documentation and focus on skilled work force which could provide higher responsiveness to rapidly changing user requirements, higher customer satisfaction and reduces defect rates. Traditional Methodologies employ development models like 'Waterfall Model', 'Spiral Model' etc which cannot handle 'rapidly changing requirements' and 'short product

cycles'. In order to cope up rapidly changing business scenario developers have innovated new development techniques which respond as well as embrace requirements change gracefully. They have developed 'Agile software development Methodologies' imbibing some concepts from earlier methodologies like 'iterative and incremental development' and prototyping.

With the arrival of numerous process models in the software development arena, a 'Methodology war' has also been reported in literature. Proponents of different process methods advocate their own process model as more effective and efficient. But there is ample evidence about successful agile projects reported in various survey reports, agile success stories, company whitepapers and agile enthusiast work groups [24,27,28].

4. CURRENT SCENARIO: TRANSITIONING FROM TRADITIONAL TO AGILE

The current product development scenario is going through a major overhaul. Older methodologies of development are becoming outdated as they are not able to match up their pace with rapidly changing technological and business requirements. Today, business is governed by 'Internet Economies'. So the rules of business are changing at a greater speed. Newer god on business horizon is 'Customer'; this makes today's fast-paced business more competitive as customer requires more services in lesser time and cost. 'Software' has become the key to successful business. So changes in business require an equivalent change in software development methods.

The current research in software development arenas is to study methodological transition. Because it has become imperative for software developers to adopt much sought after methodologies which provide an environment of responding to business needs faster than ever thus providing a competitive edge above others. To survive market competition, software developers are under immense pressure for developing better processes with which high quality products are released into market in lesser time. Adopting a new methodology is not as easy as it sounds. It's a mammoth task involving changes at various levels (mapping of size and scope, developmental process level, business process level, human resource level) of working hierarchy in a corporate house. Blending of agile processes into traditional existing processes requires either completely ignoring the agile principles or completely revamping the existing software processes. The latter option triggers series of problems because it abrogates the efforts spent on improving and refining the existing system. In this paper we present a conceptual framework where we have tried to identify focus areas where change is required during methodological change.

5. IDENTIFYING AGILE ENABLERS AND CHALLENGES FOR SUCCESSFUL METHODOLOGICAL TRANSITION

In this section, a conceptual framework (depicted in fig. 1) is presented that may provide insights during methodological transition. The frame work may help managers identify the pre-conditions for agile adoption, relating it to their own organization culture and perception. The focus is on identifying 'agile enablers' which help in guiding towards the required changes, critically needed for adopting agile practices and the risks or the challenges involved in this much needed transition. Identification of pre-conditions, enablers and challenges are based on existing literature [21, 23, 25, 29]. Presence of pre-conditions is important to initiate any sort of change in methodology and the list of enablers provided in framework symbolizes potential critical process activities for smooth methodological transition. If the enabler activities are

focused during transition, it may help create an environment suitable for gradual agile adoption. The transition of methodologies is basically transition of (work) cultures and the history of software engineering reveals that only those methodologies have survived 'methodology wars' which have efficiently tapped the change in economies and customer behavior ahead of turbulent market times. So we conclude that agile transition is important for majority of corporate domains irrespective of their working domains and our proposed framework is a very basic step towards solving the much complicated task of methodological transition.

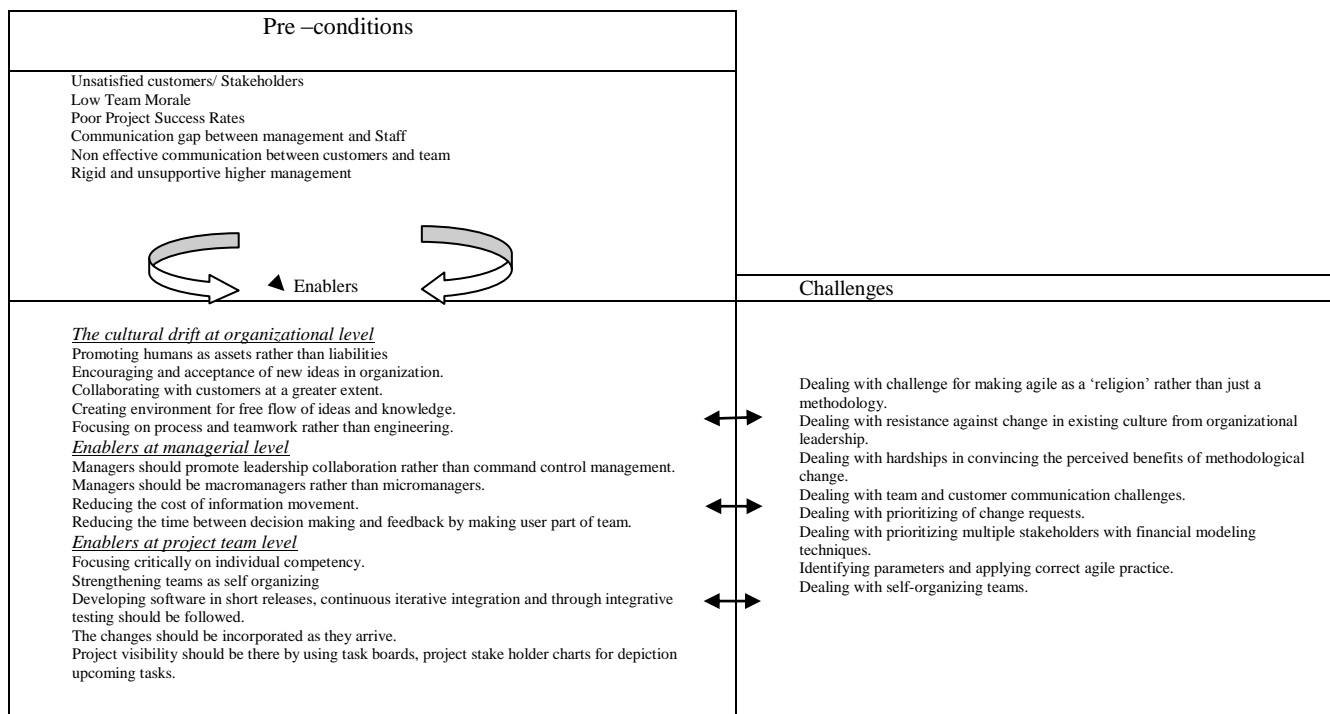


Fig 1: Conceptual Framework

6. FUTURE WORK

Currently, we are extending this list of preconditions, enablers and challenges through an extensive survey. The results will be empirically tested to identify the most distinguishing factors enabling methodological transition.

- [1] REFERENCES Buchalcevova. 2008. Research of the Use of Agile Methodologies in the Czech Republic. The Inter-Networked World: ISD Theory, Practice, and Education. Springer-Verlag: New York, ISBN 978-0387304038.
- [2] Bohem, W. B. 1988. A Spiral Model of Software Development and Enhancement. IEEE Computer, (May 1988), pp. 61-72..

- [3] Cockburn, A. and Highsmith, J. Agile Software development: The People Factor. Computer, Vol. 34 (11), pp. 131-133.
- [4] Digital Focus, "Agile 2006 Survey: Results and Analysis", Herndon, Command Information, <http://www.rallydev.com/agileblog/page/2/>, 2006, as accessed on 19.7.10.
- [5] Brooks, F. P., et al., 1987. Defense Science Board Task Force Report on Military Software, Office of the Under Secretary of Defense for Acquisition, Washington, DC 20301.
- [6] Berteig, M. 2008. Experience Report: Extremely short Iterations as a Catalyst for Effective Prioritization of work. In the proceedings of Agile 2008 conference, IEEE explore, pp. 265-268.

- [7] Alleman, G. B. 2002. Agile Project Management Methods for IT Projects, The story of Managing Projects: A global, Cross Disciplinary collection of Perspectives, Greenwood Press/ Quorum Books.
- [8] Schatz, B. and Abdelshafi, I. 2005. Primavera Gets Agile: A successful Transition to agile development. IEEE Software, May/June 2005, pp. 36-42.
- [9] Highsmith, J. and Cockburn, A. 2001. Agile Software Development: The Business of Innovation. IEEE Computer, Vol. 18(9), Sept 2001, pp. 120-122.
- [10] Livermore, A. Jeffrey, "Factors that Significantly Impact the Implementation of an Agile Software development Methodology", Journal of Software, Vol. 3 (4), pp. 31-36, 2008.
- [11] Jiang, L. and Eberlin, A. 2008. Towards a framework for understanding the relationships between classical software Engineering and Agile Methodologies. In proceedings of APSO' 08, ACM, May 2008, pp. 9-14.
- [12] Aoyama, M. 1998. Agile Software Process and its Experience. In proceedings of 20th ICSE, Kyoto, Japan, pp. 3-12.
- [13] Aoyama, M. 1998. Web-Based Agile Software development. IEEE Software, Vol. 15 (6), pp. 56-65.
- [14] Cohn, M. and Ford, D. 2003. Introducing an Agile Process to an Organization, Computer, pp. 74-78.
- [15] Griffiths, M. 2007. Developments in Agile Project Management. In proceedings PMI Global Congress, Atlanta Georgia.
- [16] Laanti, M., Salo, O and Abrahamsson, P. 2010. Agile methods rapidly replacing traditional methods at Nokia: a Survey of opinions on agile transformation. Information and Software Technology, Elsevier, 2010, doi: 10.1016/j.infsof.2010.11.010, in press.
- [17] Abrahamsson, P., Warsta, J., Simponen, M. T. and Ronkainen, J. 2003. New directions on Agile method : a comparative analysis. In the Proceedings of 25th International Conference on Software Engineering, Portland, Oregon, pp. 244-254.
- [18] Jalote, P., Palit, A., Kurien, P. and Peethamber, V. T. 2004. Timeboxing: a model for iterative software development. The Journal of Systems and Software, Elsevier, Vol. 70, pp. 117-127.
- [19] Rajamanickam. 2005. Successful Project Management using Agile Methodology. Journal of the Quality Assurance Institute, Vol. 19, pp. 15-18.
- [20] Ambler, S. 2007. Examining the Effectiveness of Agile Practices. Dr. Dobb's journal, <http://www.ddj.com/architect/200001986>, accessed on 4.5.10.
- [21] Misra, S. C., Kumar, V. and Kumar, U. 2009. Identifying some of the success factors in adopting agile software development practices. Journal of systems and software, Vol. 82 (11), pp. 1869-1890.
- [22] Cronholm, S. 2009. Using Agile method ? – Expected effects. Information Systems Development, Springer US, Chapter 96, pp 913-921.
- [23] Nerur, Mahapatra, R., Mangalaraj, G. 2005. Challenges of migrating to agile methodologies. Communications of the ACM, Vol. 48 (5), pp. 72-78.
- [24] Shine Technologies. 2007. Agile Methodologies Survey Results. <http://www.shinetech.com/download/attachments/98/shinetechnAgileSurvey2003-01-17.pdf>, accessed on 10.10.09.
- [25] Sidky, Arthur, J., Bohner, J. 2007. A disciplined approach to adopting agile practices: the agile adoption framework. Innovations in Systems and Software Engineering, Springer London, Vol. 3 (3), pp. 203-216.
- [26] The Chaos report, 1994, <http://www.projectsmart.co.uk/docs/chaos-report.pdf>, accessed on 8.1.10.
- [27] Version One, 4th Annual Survey 2009, www.versionone.com/agilesurvey, accessed on 2.4.11.
- [28] Version One, 3rd Annual Survey: The State of Agile Development. 2008. http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf, accessed on 2.4.11.
- [29] Rajlich, V. T. 2006. Changing the Paradigm of Software Engineering. Communications of the ACM, vol. 49 (8), pp. 67-70.