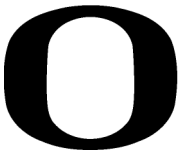
Presented to the Interdisciplinary Studies Program:



UNIVERSITY OF OREGON
APPLIED INFORMATION MANAGEMENT

Applied Information Management
and the Graduate School of the
University of Oregon
in partial fulfillment of the
requirement for the degree of
Master of Science

Identifying Cultural Changes Necessary in Traditional Plan-Driven Software Development Organizations when Preparing to Adopt Agile Principles

CAPSTONE REPORT

Name: Susan S. McElfish
Title: Project Manager
Organization: Intel Corporation

University of Oregon
Applied Information
Management
Program

February 2011

Continuing Education
1277 University of Oregon
Eugene, OR 97403-1277
(800) 824-2714

Approved by

_____

Dr. Linda F. Ettinger
Senior Academic Director, AIM Program

Identifying Cultural Changes Necessary in Traditional Plan-Driven Software Development Organizations when Preparing to Adopt Agile Principles

Susan S. McElfish

Intel Corporation

Abstract

Organizational culture plays a critical role in the acceptance and adoption of agile principles by a traditional software development organization (Chan & Thong, 2008). Organizations must understand the differences that exist between traditional software development principles and agile principles. Based on an analysis of the literature published between 2003 and 2010, this study examines nine distinct organizational cultural factors that require change, including management style, communication, development team practices, knowledge management, and customer interactions.

Keywords: agile principles, traditional software development, organizational culture, organizational change

Table of Contents

List of Tables

Introduction to the Literature Review

Purpose

The purpose of this study is to develop a set of organizational principles for traditional, plan-driven software development organizations to adopt when making the transition to agile (Vinekar, Slinkman, & Nerur, 2006). And while there are many aspects of traditional, plan-driven software organizations that could be examined, this study is focused on two: (a) identifying cultural factors that may be in conflict with agile, thus making it difficult for traditional organizations to accommodate some principles of agile development (Vinekar et al., 2006) and (b) identifying cultural changes necessary for traditional, plan-driven software development organizations preparing to adopt agile principles. In this study, organizational culture refers to values, norms, and assumptions embodied in organizational routine that influence the behavior and actions of people (Nerur, Mahapatra, & Mangalaraj, 2005).

According to Boehm and Turner, as cited by Vinekar, Slinkman, and Nerur (2006), the choice of traditional or agile methods for an organization is largely contingent upon compatibility with the prevailing culture. Traditional software development practices unlike agile methods have been dominated by process-centric engineering approaches since inception (Nerur et al., 2005). Process-centric software engineering follows the belief that there is a source of variation in the process that can be identified and eliminated by continually measuring and refining processes (Nerur et al., 2005). The focus of traditional, plan-driven software development is achieving efficiency by improving repeatable processes (Bose, 2008). Systems development in the traditional approach is guided by a life cycle model; examples include the waterfall model, spiral model, and some variations of these two (Nerur et al., 2005). A life cycle model, when used in software development, specifies the tasks to be performed, identifies what items the team is to deliver by phase, and assigns specific roles to those individuals that perform

development tasks (Nerur et al., 2005). Traditional software development organizations attempt to control change that may occur during "the course of a project through rigorous upfront requirements gathering, analysis, and design under a controlled schedule" (Vinekar et al., 2006, p.31). The impetus for traditional software development is "planning and control accomplished by a command and control management style" (Nerur et al., 2005, p. 74).

According to Nerur et al., (2005), the transition from this command and control aspect of traditional, plan-driven software development methodology to an agile focus of "adaptation and innovation" (Vinekar et al., 2006, p. 32) presents one of the most significant hurdles for organizations adopting agile. Agile methods refer to software development practices that are highly collaborative, favoring human interaction over processes and tools, a working piece of software over comprehensive documentation, customer collaboration over contract negotiation, and the ability to respond to change over following a detailed plan (De Cesare, Lycett, Macredie, Patel, & Paul, 2010). According to Bose (2008), agile methods overcome the limits of traditional, plan-driven software development by considering that requirements are not static but dynamic. Instead of development being limited by highly defined processes of traditional methods, agile development processes are minimally defined and adaptive (Bose, 2008). Agile assumes that change is inevitable and necessary during development, and that change is an opportunity for members of the development team to achieve innovation through individual initiative (Vinekar et al., 2006).

Organizational culture is a condition for the success of intended innovation and should be considered in order to successfully program changes (introduce new methods) necessary for agile adoption (Tolfo & Waslzwick, 2008). It is fundamental that some cultural aspects of agile be present in the working environment of a traditional, plan-driven software development organization preparing to adopt agile (Tolfo & Wazlawick, 2008). Nerur et al. (2005) assert "past research shows that software development process changes represent complex organizational

change phenomena and cannot be accomplished by merely replacing current tools and technologies" (p.74). McBreen, cited by Tolfo and Wazlawick (2005), warns that adaptation of a new software development method (SDM) is not an easy task, since organization members need to change attitude and values. De Cesare, Lycett, Macredie, Patel, and Paul (2010) state that successful software development practice "depends on a number of non-technical issues that are managerial, cultural, and organizational in nature" (p. 126). Companies with an organizational culture that is highly incompatible with agile values and principles may find adoption to be exhaustive (Tolfo & Waslzwick, 2008). In particular, Nerur et al., (2005) note that in order for organizations to successfully adopt agile methodologies they must rethink their goals and reconfigure their human and managerial components. They continue with the perspective that to be successful in adopting agile methods, traditional, plan-driven organizations need to address issues related to organizational culture (Nerur et al., 2005).

Problem Area & Significance

Black, Bock, Bowen, Gorman, and Hinchey (2009) state that the discipline of software engineering has evolved from hardware engineering in the 1950s to software crafting in the 1960s, formality and waterfall process in the 1970s, productivity and scalability in the 1980s, sequential processes in the 1990s, and agility and value in the 2000s. Since the industry-wide self-declared software development crisis in the late 1960s, there has been significant effort by the software engineering community to address problems related to cost, time, and quality of software development projects (De Cesare et al., 2010). Early efforts, according to De Cesare et al. (2010), resulted in implementation of prescriptive structured methods associated with traditional, plan driven development. The low level of success in the field of software development that resulted from these efforts "provided the impetus for the development of several new methods and practices" (Vinekar et al., 2006, p. 31). The new development practices, based on what is called the agile manifesto, includes methods such as "eXtreme

Programming, Scrum, Dynamic Systems Development Methods, Adaptive Software Development, Crystal, and Feature-Driven Development" (Vinekar et al., 2006, p. 31).

Traditional, plan-driven software development organizations tend to implement what are labeled as "heavy" development process approaches and quality systems, which are often too generic and complex for good development practices to occur (Lindvall et al., 2004). Corporate software development teams have traditionally used heavyweight waterfall methodologies for developing most software applications (Grossman, Bergin, Leip, Merritt, & Gotel, 2004). Traditional organizations follow a sequential development process in which requirements are gathered upfront, roles are assigned to the developers for coding, and software is tested and integrated with existing software (Bose, 2008). Although traditional software development methods work well for stable development projects, they are not sufficiently responsive when project requirements are not well understood or when requirements are changing and evolving (Grossman et al., 2004). According to Lindvall et al. (2004) these processes and systems limit what development practices the development team can and must use, which affects how quickly they can develop a piece of software.

Faced with the increasing pressures to produce software products at a lower cost while increasing productivity and maintaining quality, most software development organizations look for new ways to develop software (Lindvall et al., 2004). Finding alternate methods to develop software faster and more flexibly without compromising quality becomes essential and is the reason traditional software development organizations have turned their attention to agile methods (Lindvall et al., 2004). The primary goal of agile development methods is to deliver software quickly, and to adapt to changes in the process, product, and environment (Strode, Huff, & Tretiakov, 2009). Both large and small software development organizations have shown interest in agile methodologies because they seek alternatives to traditional software development methods which are too cumbersome, bureaucratic, and inflexible (Lindvall et al.,

2004). The introduction of agile methods is seen as a reaction from the software development industry to the cumbersome traditional methods which focus on formalizing requirements at the beginning of the development lifecycle and delivering a product at the end without much interaction with the customer in between (Black et al., 2009).

Organizations in today's business environment "need information systems that constantly evolve to meet their changing business requirements" (Nerur et al., 2005, p. 73). Traditional, plan-driven software development methodologies used by many organizations today lack the flexibility necessary to dynamically adjust the development process to meet the needs of a changing environment (Nerur et al., 2005). Agile development is a new generation of software development methodologies that claims to be better suited for dealing with a dynamic business environment than traditional software development methodologies (Chan & Thong, 2008). Agile development methods "accommodate change by employing a rapid iterative and incremental development process with high levels of communication and customer involvement" (Strode et al., 2009**,** p. 1). While traditional methodologies such as waterfall and object oriented continue to dominate the development arena, surveys and opinion pieces confirm that agile is growing in popularity (Nerur et al., 2005).

Audience

The audience for this study is the stakeholders who make up the organizational structure of a software development organization and have influence over the adoption of agile methodologies (Tolfo & Wazlawick, 2008). Stakeholders are defined as those individuals in the software development organization who affect or are affected by agile methodology including managers, company managers, developers, and customers (Tolfo & Wazlawick, 2008). Developers, represented by those individuals involved in software conception and development, include system analysts, system architects, programmers, and testers (Tolfo & Wazlawick, 2008). Chief Information Officers and project managers within the software development

organization "must be cognizant of the challenges they will face in their endeavors to embrace the agile philosophy of software development" (Nerur et al., 2005, p. 74). Programmers and analysts now work in a less structured team environment and need to learn news tools to successfully adopt agile (Nerur et al., 2005).

Outcome

The outcome of this study is a guide designed for use by the intended audience that identifies cultural factors that impact the adoption of agile principles and presents the actions necessary for making cultural change for the purpose of adopting agile principles by organizations steeped in traditional, plan-driven development methodologies.

Emerging evidence seems to indicate that agile development methodologies are gaining acceptance among traditional software development organizations (Vinekar et al., 2006). Many software development organizations are switching to agile development practices because of the attractive claims of success from the industry (Misra, Kumar, & Kumar, 2009). Nerur et al., (2005) believe that software development organizations that are "conducive to innovation may embrace agile methods more easily than those built around bureaucracy and formalization" (p. 78). The variations between traditional methods of software development and agile suggest that organizations must carefully assess their human, managerial, and technology components in order to successfully adopt agile methodologies (Nerur et al., 2005). According to Lindvall et al., as cited by Misra, Kumar, and Kumar (2009), "having the right corporate culture is almost unanimously perceived by agile experts to be a necessary factor determining the introduction of agile practices" (p. 1871). As stated by Beck, cited by Grossman et al. (2004), the biggest barrier to the success of an agile adoption is organizational culture. It is fundamental that certain cultural values and attitudes are present in traditional, plan-driven software development organizations when adopting agile, in order to achieve success (Tolfo & Wazlawick, 2008).

Delimitations

Time frame. In 2001, several prominent members of the agile development community introduced the Agile Manifesto made up of 4 core values and 12 principles of agile software development (Lee & Xia, 2010). Only literature published after the introduction of the agile manifesto is considered for review.

Focus. Adoption of agile methodologies requires a focus on key issues related to management practices, organization, people, process, and technology (Nerur et al., 2005). Under the category of management and organizations are the key areas of organizational culture, management style, organizational form, and reward systems (Nerur et al., 2005). Because "organizational culture has a significant impact on the social structure of organization, which in turn influences the behavior and actions of people" (Nerur et al., 2005) this research is focused specifically on the management and organizational factors impacting agile adoption.

Inquiry context. Literature selected for this study addresses the adoption of agile methods in various software development environments or addresses agile adoption issues in general, not specific to an agile type or environment. The traditional, plan-driven software development environment is not researched as a separate context because the selected literature in all cases addressed the subject in relation to adopting agile methodologies.

Author. Nerur, assistant professor of information systems at the University of Arlington, Texas, has published multiple references focused primarily on the organizational impacts of applying agile principles. Nerur places a special focus on organization change and cultural factors impacting the adoption of agile methods as cited in the following lines. Nerur et al., (2005) note that in order for organizations to successfully adopt agile methodologies they must rethink their goals and reconfigure their human and managerial components. Nerur et al., (2005) continue with the perspective that to be successful in adopting agile methods, traditional, plan-driven organizations need to address issues related to organizational culture. Nerur's focus and

perspective on organizational culture are of special interest to the researcher in relation to the subject of this study. Nerur's works are used as underpinnings to the research focus and to the research questions in this study and as such are referenced heavily.

Data Analysis Plan Preview

The approach to data analysis selected for use in this literature review is conceptual analysis as defined and described by Busch et al. (2005). The analysis begins with the development of pertinent research questions and the selection of key literature. Then a series of coding concepts are developed for application to selected references, including the level of analysis, the number of key concepts to be coded, the coding rules, etc. (Busch et al., 2005).

Writing Plan Preview

The writing plan refers to how the data identified during the coding process is presented in the Review of the Literature section of the paper. The pattern of organization selected for this review is thematic. The thematic pattern of organization allows the researcher to present the results of the data analysis process according to a theme or issue derived during analysis (Anson et al., 2007).

The writing plan is designed to identify the key cultural factors that make it difficult for traditional software development organization to adopt agile principles so that they are better prepared to take the necessary steps to begin the process.

Definitions

The definitions section of this study provides a comprehensive listing of all the terms found in this literature review. Providing a comprehensive list of definitions is necessary to establish a level of understanding for any reader outside the field of study of the terms that go beyond common language (Creswell, 2009).

- Agile development methodology is a term used to describe a highly iterative and incremental approach to software development. Agile methods include scrum, extreme programming, dynamic systems development method, lean development, feature-driven development, crystal, adaptive software development, and others that incorporate close, cross-functional collaboration, frequent planning, and regular project feedback fundamental to the evolution of a software system. The focus of agile development projects is frequent delivery of high quality, working software in the form of business valued functionality. Each agile method emphasizes ongoing alignment between technology and the business and is considered lightweight in nature in that they strive to impose a minimum of bureaucracy and overhead within the development lifecycle. Agile methods are adaptive in that they embrace and manage changing requirements and business priorities throughout the development process. Agile methods place considerable emphasis on empowering teams and collaborative decision making (AgileSherpa.com, 2010).
- Agile manifesto is a philosophy published in 2001, for approaching software development and describes the four core values and 12 principles of agile development (Kane et al., 2006).
- Agile principles are the "fundamental guidelines concerning software development activities" (Bozheva & Gallo, 2005, p. 6).
- Adaptive Software Development is a framework for managing software that is under intense time constraints with rapidly changing requirements (Strode et al., 2009).

- Crystal clear method is a member of the Crystal family of methodologies and is considered an example of an agile or lightweight methodology (Wikipedia, 2010).
- Dynamic Systems Development method is an iterative and incremental approach that emphasizes continuous user involvement during the software development process (Wikipedia, 2010).
- Extreme Programming method of development relies on simple design, refactoring, and test first development methods (Tate, 2006).
- Feature-Driven Development method is an iterative and incremental software development process (Wikipedia, 2010).
- Heavyweight software methods refer to method heaviness characteristic of traditional approaches requiring the production of non-software artifacts, mainly documentation, during development (Strobe, Huff, & Tretiakov, 2009).
- Object oriented methodology is a programming paradigm that uses objects, data structures consisting of data fields, and methods together with their interactions to design applications and computer programs (Wikipedia, 2010).
- Organizational culture according to Schein, cited by Chan and Thong (2008), is "a pattern of basic assumptions invented, discovered or developed by a given group as it learns to cope with its problems of external adaptation and integration that has worked well enough to be considered valid and therefore is to be taught to new members as the correct way to perceive, think, and feel in relation to those problems" (p. 809).
- Process-centric engineering follows the belief that there is a source of variation in the process that can be identified and eliminated by continually measuring and refining processes (Nerur et al., 2005).
- Scrum is an agile development method that focuses on project management practices (Kane et al., 2006).

- Spiral lifecycle model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts (Wikipedia, 2010).
- Software development methodology (SDM) is a documented collection of policies, processes, and procedures used by a software development team to improve the software development process (Chan & Thong, 2008).
- Traditional, plan-driven software development is an engineering-based approach "incorporating extensive planning, codified processes and rigorous reuse" (Dyba & Dingsoyr, 2009).
- Waterfall lifecycle model is focused on capturing detailed customer requirements at the beginning of the software development lifecycle and delivering a product at the end with very little customer interaction in between (Black et al., 2009).

Research Parameters

The research parameters section of this paper describes the overall research design of this literature review study. The concept of a literature review is defined as "a self-contained piece of written work that gives a concise summary of previous findings in an area of the research literature" (Hewitt, 2002, p. 5). The literature review is a "focus on empirical studies and seek to summarize past research by drawing overall conclusions from many separate investigations that address related or identical hypotheses" (Cooper, 1998, p. 3). The literature review reflects an author's knowledge and interpretation of an area of interest (Hewitt, 2002). The research parameters include descriptions of the detailed search strategy used to locate the literature, the documentation approach for recording the information extracted from the literature, the evaluation criteria used for the selecting the literature, and the data analysis and writing plans for the study.

Research Questions

Organizational culture is considered a key factor effecting the successful adoption of agile methods (Strode et al., 2009). Organizations that support a culture of hierarchical control typical of traditional, plan-driven organizations find it particularly difficult to accommodate some principles of agile development (Vinekar et al., 2006). Agile software development requires a suitable organizational culture that is very different from the organization culture of traditional software development organizations (Vinekar et al., 2006). Through preliminary examination of selected literature that addresses how cultural factors can impact a traditional software development organizations ability to adopt agile methods, the researcher developed the following questions in order to guide the study:

1. What cultural factors of traditional software development organizations are in conflict with the adoption of agile principles?

2. What parallel cultural changes are required for traditional software development organizations to successfully adopt agile principles?

Search Strategy Report

This inquiry is structured as a literature review based on information derived from peer reviewed software industry journal articles, books, symposium and conference proceedings, and websites relevant to the subject of agile software development.

Key search terms.

- Agile software development methodology

Subtopic search terms.

Agile software development methodology

- Cultural impacts of agile software development
- Adopting agile methods of software development
- Making the transition to agile methods
- Organizational culture and agile
- Plan-driven software development and agile

Selection and Evaluation Criteria

The literature for this study is retrieved from multiple computer science databases found at the UO Library website including ArXiv, Web of Science, Computer Source, EBSCOhost, ACM Digital Library, and IEEE Computer Society Database. Literature selected for this study is collected from books, professional software and computing journals, thesis and doctorate work, conference and symposium proceedings, references identified within selected articles, and professional and academic websites.

Literature selection is based on the following criteria:

- The author is affiliated with an accredited university or is a professional with industry level of expertise (Bell & Smith, 2009).

- The literature is published in industry-accredited software engineering journal, conference proceeding, or symposium proceeding (Bell & Smith, 2009).
- The literature is peer reviewed (Bell & Smith, 2009).
- The literature is current for the subject (Bell & Smith, 2009).
- The literature is highly relevant to the area being researched (Bell & Smith, 2009).

Literature evaluation is based on the following criteria.

Relevancy of the literature is determined by the following criteria.

- The literature addresses the research questions defined for the study (Bell & Smith, 2009).
- The content is derived from scholarly and fact based journal articles, books, and websites (Bell & Smith, 2009).

Quality of the literature is determined by the following criteria.

- The information is well organized, the main points are presented clearly, and the main ideas are unified (Bell & Smith, 2009).
- The information is complete and accurate demonstrated by results and facts that align with researcher's knowledge of the subject (Bell & Smith, 2009).
- Literature contains documented sources (Bell & Smith, 2009).
- Literature avoids questionable assumptions (Bell & Smith, 2009).

Documentation Approach

The documentation approach for this study is a manual extraction of the required content during the coding process from the selected literature, which is then placed into a Word document. Each piece of literature is reviewed in detail, critical content highlighted, and relevant information logged into Word. Reference information is captured at the time the literature is located including the abstract for the purpose of immediate reference should the literature source

be selected for the study. All reference detail is logged in a separate Word document and updated when new sources are located. Initial literature details including title, publication, peer review status, author affiliation, and author credentials is collected and stored in an excel file for the purpose of evaluation.

Data Analysis Plan

The data analysis plan selected for use in this literature review is conceptual analysis as defined and described by Busch et al. (2005). The analysis begins with the development of pertinent research questions and the selection of key literature. Then a series of coding concepts are developed and addressed, including the level of analysis, the number of key concepts to be coded, the coding rules, etc. (Busch et al., 2005).

Coding procedure. The conceptual analysis process framed for this study consists of the following eight defined coding steps.

1. Level of analysis – single words such as culture, values, attitudes, behaviors, and set of words such as organizational culture, organizational change, software development process or practice, software development process change or transition, traditional (or heavy) software development process or practice, and agile (or light) software development process or practice are coded.
2. Pre-defined set of concepts and categories – only words that are relevant to these concepts: organizational culture, organizational characteristics, software design process or practice, agile software development methods, traditional software development methods, and software design process change or transition are coded. New relevant concepts or categories that emerge during analysis may be added or used to modify pre-defined concepts and categories.
3. Frequency of a concept – emphasis is placed on coding for frequency as well as coding for existence. Therefore, the concept *organizational culture* is coded each

time the concept appears in the source being analyzed no matter how many times it appears. Variations of concepts are coded separately (e.g. organizational culture in relation to management practice or organizational culture in relation to values and beliefs).

4. Level of generalization – similar concepts and categories such as *organizational structure* and *organizational form* or *culture* and *organizational culture* are recorded as the same. In addition, the various methods of agile software development including *eXtreme Programming, Scrum, Dynamic Systems Development Methods, Adaptive Software Development, Crystal, and Feature-Driven Development* are coded as *agile methods.* Similar terms that have different meaning such as *organizational culture* and *cultural aspects* are coded separately.
5. Translation rules – translation rules are developed to ensure that terms and concepts are categorized consistently. For instance, *cultural aspects* are coded under *organizational characteristics*, *values* and *behaviors* are coded under *cultural aspects,* and *agile software development methods* are coded the same as *agile software development practices, processes, and principles*.
6. Irrelevant information – irrelevant information is discarded as long as the information does not influence the analysis results.
7. Code the text – coding is conducted manually; the first step is reading the printed article or book, recording the concept occurrences on post it notes, and attaching the notes to the literature source. Once the coding is completed, notes are transcribed into a Word document. Tracking of all coding is maintained in a Word document containing a source number, concepts, coding terms, title, publication year, and author.

8. Analyze results – at this stage the results recorded in Word are further analyzed by the researcher who draws all possible conclusions. Results are presented according to thematic organization scheme, described in the writing plan.

Writing Plan

The writing plan refers to how the data identified during the coding process is presented in the Review of the Literature section of the paper. Specifically, the writing plan is designed to present the key cultural factors that make it difficult for traditional software development organization to adopt agile principles so that they are better prepared to take the necessary steps to begin the process.

The pattern of organization selected for this review is thematic. The thematic pattern of organization allows the researcher to present the results of the data analysis process according to a theme or issue derived during analysis (Anson et al., 2007). The two anticipated themes are:

Theme one: Organizational cultural factors that impact the adoption of agile principles

- Cultural factors inherent to traditional software development organizations
- Cultural factors inherent to agile software development organizations
- Areas of cultural conflict between agile and traditional software development organizations

Theme two: Organizational cultural change within traditional software development organizations that is necessary for adopting agile principles

- Key areas of cultural change required for adopting agile principles
- Actions necessary for a traditional, plan-driven software development organizations adopting agile principles

Annotated Bibliography

The annotated bibliography is a list of the key references used to support the data analysis portion of this study. The following 20 references are selected for coding and extraction of content for the purpose of writing the review of the literature.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software, 22*(5), 30-39. doi.ieeecomputersociety.org/10.1109/MS.2005.129

Abstract. Agile software development processes have shown positive impacts on cost, schedule, and customer satisfaction. However, most implementations of agile processes have been in smaller-scale, software-only environments. In March 2004, a group of researchers and practitioners addressed the implementation of agile processes into large systems engineering-based projects that rely on traditional development processes and artifacts. They identified three management challenge areas. The authors discuss numerous ways in which to address them.

Comments. This article is relevant to the study because it addresses the management challenges a traditional development organization may face when adopting agile software development methods. The authors discuss the cultural barriers that managers face when adopting agile methods and provides strategies to help address them. The article is deemed credible because it is published in a peer reviewed accredited software engineering journal and co-authored by a professor and director of the center for software engineering at the University of Southern California and the director of the systems and software consortium. The article is published in 2005 making it current for the study and the content of the article is highly relevant to the research questions identified in this study.

Bose, I. (2008). Lessons learned from distributed agile software projects: A case-based analysis. *Communications of AIS*, *2008*(23), 619-632. Retrieved from Computer Source database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=cph&AN=41671742&login.asp&site=ehost-live&scope=site

Abstract. Agile software development in a distributed setting is challenging. The teams involved in the process face difficulties in communication, personnel selection, work culture, and knowledge management. The shortcomings associated with working in different time zones and the inabilities to develop trusting relationships between developers are well known. Companies often take recourse to agile software development methods in a distributed environment in search of reduced cost, higher efficiency, increased flexibility, and good customization. However, it is not clear whether agile methods can be successfully followed and their benefits realized in a distributed setting. This paper revisits and synthesizes the lessons learned from twelve case studies detailing successful implementation of distributed agile software projects. The cases are analyzed from the perspective of the agile manifesto to determine how closely they follow its values and principles and to what extent they realize the benefits of the agile methodology. The cases lead to the discovery of disparate and innovative solutions adopted by different companies for overcoming the challenges of distributed agile software development. Some solutions are commonplace and others are unique and their combination in the context of the challenges is enlightening. The list of solutions can suitably guide companies that plan to adopt the agile methodology in distributed software development environments in the future.

Comments. This article is important to the study as it describes some of the cultural challenges involved in adopting the principles of agile methodologies and provides possible strategies for addressing these challenges. The article is deemed highly credible

because it is peer reviewed and published in the industry-accredited journal Communications of the Association for Information Systems. The author Indranil Bose is associate professor of Information Systems at the University Of Hong Kong School Of Business and holds two MS degrees, a PhD, and is listed in four different versions of Marquis Who's Who. The subject of lessons learned from agile software projects is highly relevant to the area being researched and the publication date of December 2008 is current for the subject.

Bozheva, T., & Gallo, M. (2005). Framework of agile patterns. *Software Process Improvement, Proceedings, 3792*(147915531), 4-15.

Abstract. The variety of agile methods and their similarity could be a problem for software engineers to select a single or a number of methods and to properly execute them in a project. A pattern describes a problem, which typically occurs under certain circumstances and a basic approach to solve it providing opportunities to adapt the solution to the problem. The agile patterns, described herein, are based on the principles and practices of the best known agile methodologies. While individual practices included in any of these methods vary, they all have particular objectives and related activities. Therefore, every pattern is described as to show the core solution to a particular problem. Special attention is paid to the rationale for applying the agile patterns: what are the business drivers to adopting them; in what cases do they bring benefits; how could they be introduced in an organization.

Comments. This conference proceeding is highly relevant to this study because the content is a summary of work related to applying agile practices in different organizational contexts inspiring the creation of a framework of agile patterns for use by organizations adopting agile principles. This reference is deemed credible because it is published in a peer reviewed conference proceeding and co-authored by two industry

professionals from the European Software Institute. The proceeding published in 2005 is current for the study and highly relevant to the research questions identified for the study.

Chan, F., & Thong, J. (2008). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems, 46*(4)*,* 803-814. doi: 10.1016/j.dss.2008.11.009

Abstract. It is widely believed that systems development methodologies (SDMs) can help improve the software development process. Nevertheless, their deployment often encounters resistance from systems developers. Agile methodologies, the latest batch of SDMs that are most suitable in dealing with volatile business requirements, are likely to face the same challenge as they require developers to drastically change their work habits and acquire new skills. This paper addresses what can be done to overcome the challenge to agile methodologies acceptance. The authors provide a critical review of the extant literature on the acceptance of traditional SDMs and agile methodologies, and develop a conceptual framework for agile methodologies acceptance based on a knowledge management perspective. This framework can provide guidance for future research into acceptance of agile methodologies, and has implications for practitioners concerned with the effective deployment of agile methodologies.

Comments. This article is important to the study because it addresses what can be done to overcome the challenges a traditional software development organization may face when adopting agile methodologies. The authors provide a conceptual framework for acceptance of agile methods based on a knowledge management perspective. The article is deemed highly credible because it is co-authored by a doctoral student in information systems from the accredited Hong Kong University of Science and Industry and a university professor of the same institution with an extensive research background in technology adoption. The article published in the peer reviewed and accredited

Decision Support System journal in 2008 is current for the subject and the reference content is highly relevant to the research questions identified for this study.

Grossman, F., Bergin, J., Leip, D., Merritt, S., & Gotel, O. (2004). One XP experience: Introducing agile (XP) software development into a culture that is willing but not ready. *Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, Canada,* 242 – 254. Abstract retrieved from http://portal.acm.org.libproxy.uoregon.edu/citation.cfm?id=1034914.1034933&coll=DL&dl=GUIDE&CFID=4119751&CFTOKEN=14492728

Abstract. The main question to be asked is "Does Extreme Programming (XP) make sense as a development methodology in a diverse, multidisciplinary web development environment? This environment includes diverse, and perhaps, distributed teams requiring close coordination with multidisciplinary skills -- information architecture, visual design, XML, Java and others. The potential is to make the development process more responsive to users' needs and changing business requirements. This could have high impact on outcomes of the development process, decreasing cost, decreasing time to deployment, and increasing user satisfaction. The challenges are to adapt and reconcile the corporate and the agile culture processes and methodologies without seriously compromising either. The authors discuss their experience from conception into implementation of XP through the first release that incorporates several iteration cycles. They discuss the positive and negative cultural forces and how they have or have not been resolved to date.

Comments. This conference proceeding is highly relevant to this study because the authors address the transition of a large traditional based software development organization adopting agile methods. The authors consider the cultural environment and evaluate the organizations ability to adopt agile methods based on existing cultural

factors. This article is deemed credible because it is published in a peer reviewed conference proceeding and co-authored by four university professors specializing in software development engineering from the accredited Pace University and one industry professional specializing in internet application development. The article is published in 2004 making it current for the study and the content of the article is highly relevant to the study on adopting agile principles.

Kane, D., Hohman, M., Cerami, E., McCormick, M., Kuhlmman, K., & Byrd, J. (2006). Agile methods in biomedical software development: a multi-site experience report. *BMC Bioinformatics*, *7*273-12. doi:10.1186/1471-2105-7-273. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=28833806&site=ehost-live&scope=site

Abstract. Agile is an iterative approach to software development that relies on strong collaboration and automation to keep pace with dynamic environments. This paper reports on a qualitative study done by a biomedical development team using agile methods. The team found that agile methods are well suited to the exploratory and iterative nature of scientific inquiry. Agile provides a robust framework for reproducing scientific results and for developing clinical support systems. The agile development approach also provides a model for collaboration between software engineers and researchers. The authors present the experiences of the teams using agile methodologies in projects at six different biomedical software development organizations. The organizations include academic, commercial and government development teams, and included both bioinformatics and clinical support applications. The authors found that agile practices were a match for the needs of biomedical projects and contributed to the success of the organizations. In conclusion, the authors found that the agile development approach was a good fit, and that these practices should be applicable and

valuable to other biomedical software development efforts. Although they found differences in how agile methods were used, they were also able to identify a set of core practices that were common to all of the groups, and that could be a focus for others seeking to adopt these methods.

Comments. This article is important to the study because it captures the results of several software development projects adopting agile methods and describes the cultural characteristics of the organizations that contribute to a successful transition. The article is deemed highly credible because it is published in a peer reviewed industry accredited journal and co-authored by a team of university researchers from Northwestern, Vanderbilt, Fred Hutchinson cancer center, and Memorial Sloan-Keating cancer center as well as industry experienced software development engineers from commercial and government institutions. The content published in 2006 is current for the subject and highly relevant to the research questions identified for the study.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., et al. (2004). Agile software development in large organizations. *Computer*, *37*(12), 26-34. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=15586953&site=ehost-live&scope=site

Abstract. Developers need evidence that a new technology works in a certain context before they promote and deploy it on a larger scale. This need looms greater in large organizations because of their complexity and the need to integrate new technologies and processes with existing ones. To further evaluate agile methods and their underlying software development practices, several Software Experience Center member companies initiated a series of activities to discover if agile practices match their organizations' needs. Based on the experiences of these organizations, researchers concluded that agile

practices match the needs of large organizations, but integrating new practices with existing processes and quality systems that govern the conduct of software development requires further tailoring. The challenge here lies not in applying agile practices to a project, but in efficiently integrating the agile project into its environment.

Comments. This article is important to the study because it considers some of the cultural aspects of adopting and integrating agile practices into an existing traditional software development organization. The authors evaluate the results of several pilot projects and identify the need for organizations to consider tailoring of principles as an option to allow organizations to successfully adopt agile methods in order to align with existing traditional principles. This article is deemed credible because it is published in an accredited peer reviewed journal and co-authored by software industry engineers and researchers from several software industry development organizations. The article is published in 2004 making it current for the study and the content of the article is highly relevant to the study on adopting agile principles.

Misra, S., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869-1890. doi: 10.1016/j.jss.2009.05.052

Abstract. Agile software development (ASD) is an emerging approach in software engineering, initially advocated by a group of 17 software professionals who practice a set of "lightweight" methods, and share a common set of values of software development. In this paper, the authors advance the state-of-the-art of the research in this area by conducting a survey-based ex-post-facto study for identifying factors from the perspective of the ASD practitioners that influence the success of projects that adopt ASD practices. The authors describe a hypothetical success factors framework developed to address the research question, the hypotheses that conjectured the research

methodology, the data analysis techniques used to validate the hypotheses, and the results obtained from data analysis. The study is conducted using an unprecedentedly large-scale survey-based methodology, consisting of respondents who practice ASD and who have experience practicing plan-driven software development in the past. The study indicates that nine of the 14 hypothesized factors have statistically significant relationship with "Success". The important success factors found are: customer satisfaction, customer collaboration, customer commitment, decision time, corporate culture, control, personal characteristics, societal culture, and training and learning.

Comments. This article is critical to the study because it directly addresses the impact that cultural factors have on the successful adoption of agile software development practices. The authors take a research approach to identify the most important factors that influence the success of adopting agile practices based on input received from a large sample of companies that have transitioned from traditional, plan-driven development to agile. The article is deemed highly credible because it is published in a peer-reviewed accredited journal and co-authored by three highly accredited individuals working in academia and industry with extensive research experience in the area of software development. The article published in 2009 is current for the subject and the reference is highly relevant to the research questions identified for the study.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 73-78. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=16915874&site=ehost-live&scope=site

Abstract. The article articulates the challenges that chief information officers and project managers must be cognizant of in their endeavors to embrace the agile philosophy of software development. It focuses on constantly changing software development

methodologies owing to changing technologies and new demands from users. Several surveys demonstrate the growing popularity of agile methodologies in the field of software development. Past research shows that software development process changes represent complex organizational change phenomena and cannot be accomplished merely by replacing current tools and technologies with new ones. Uncertainties in the field of software development are further compounded by the diversity and unpredictability of people who engage in such tasks. A rationalized, engineering-based approach has dominated software development almost since its inception. Systems development in the traditional approach is guided by a life cycle model such as the waterfall model, the spiral model, or some variations of these.

Comments. This article is highly relevant to the study as it lays the framework for identify the key challenges a traditional software organization faces when adopting agile principles. The authors provide a clear compare and contrast between traditional and agile software development in order to provide organizations with the knowledge to assess their readiness in preparation for adopting agile methods. The article is deemed credible because it is published in a peer reviewed industry accredited journal and co-authored by two professors and one doctoral student from the accredited University of Texas at Arlington. The reference published in 2005 is current for the subject and highly relevant to the research questions identified for the study.

Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems & Software*, *82*(9), 1479-1490.

Abstract. Recent empirical studies have been conducted identifying a number of issues and advantages of incremental and agile methods. However, the majority of studies focused on one model (Extreme Programming) and small projects. To draw more general

conclusions we conduct a case study in large-scale development identifying issues and advantages, and compare the results with previous empirical studies on the topic. The principle results are that (1) the case study and literature agree on the benefits while new issues arise when using agile in large-scale and (2) an empirical research framework is needed to make agile studies comparable.

Comments. This article is highly relevant to the study because it addresses the decision of a traditional software development organization adopting agile methods, highlighting the organizational problems that have to be addressed prior to making the transition. The article is deemed highly credible because it is published in the peer reviewed research Journal of Systems and Software and co-authored by the Pro Vice Chancellor of the accredited Blekinge Institute of Technology, Sweden and an industrial Ph.D. student of the same institution. The article published in 2009 is current for the subject and the content is highly relevant to the research questions identified for the study.

Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems & Software*, *81*(11), 1899-1919.

Abstract. Agile methods are often seen as providing ways to avoid overheads typically perceived as being imposed by traditional software development environments. However, few organizations are psychologically or technically able to take on an agile approach rapidly and effectively. Here, we describe a number of approaches to assist in such a transition. The Agile Software Solution Framework (ASSF) provides an overall context for the exploration of agile methods, knowledge and governance and contains an Agile Toolkit for quantifying part of the agile process. These links to the business aspects of software development so that the business value and agile process are well aligned.

Finally, we describe how these theories are applied in practice with two industry case studies using the Agile Adoption and Improvement Model (AAIM).

Comments. This article is highly relevant to the study because it addresses the ability of traditional software development organizations to rapidly and effectively take on agile principles, describing a number of approaches to assist in the transition. The article is deemed highly credible because it is published in the peer reviewed and accredited research Journal of Systems and Software and co-authored by a professor and Ph.D. student of Information Technology from the accredited University of Technology, Sydney. The article published in 2008 is current for the subject and the content is highly relevant to the research questions identified for the study.

Robinson, H., & Sharp, H. (2005). Organisational culture and XP: three case studies. *Proceedings of the Agile Development Conference,* Denver, CO, 49-58. doi.ieeecomputersociety.org/10.1109/ADC.2005.36

Abstract. This article explores the nature of the interaction between organisational culture and XP practice via three empirically-based case studies. The case studies cover a spectrum of organisational cultures. The study findings suggest that XP can thrive in a range of organisational cultures and that the interaction between organisational culture and XP can be complex & subtle, with consequences for practice.

Comments. This article is critical to the study because it directly addresses the issue of organizational culture and the impacts organizational culture can have on an agile adoption. The authors observe three different organizational cultures in process of adopting agile principles and highlight the consequence each culture has on the adoption process. The article is deemed credible because the authors are well published in accredited peer reviewed journals and conferences based on research done in the area of software development engineering from the mid-1990s to present. The reference

published in 2005 is current for the subject and highly relevant to the research questions identified for the study.

Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: The agile adoption framework. *Software Engineering,* n.p. Retrieved from arXiv: 0704.1294v1

Abstract. Many organizations aspire to adopt agile processes to take advantage of the numerous benefits that it offers to an organization. Those benefits include but are not limited to, quicker return on investment, better software quality, and higher customer satisfaction. To date however, there is no structured process (at least in the public domain) that guides organizations in adopting agile practices. To address this problem we present the Agile Adoption Framework. The framework consists of two components: an agile measurement index and a 4-Stage process, that together guide and assist the agile adoption efforts of organizations. More specifically, the agile measurement index is used to identify the agile potential of projects and organizations. The 4-Stageprocess, on the other hand, helps determine (a) whether or not organizations are ready for agile adoption, and (b) guided by their potential, what set of agile practices can and should be introduced.

Comments. This article is highly relevant to the study because it identifies an organizational assessment approach for organizations interested in adopting agile methodology. The authors review a structured and repeatable agile adoption framework consisting of a measurement index and 4 stage processes for use by organizations to assess their agility and determine which agile practices might be introduced in order to achieve a successful agile adoption. The article is deemed credible because the three authors are well published in many accredited peer reviewed journals, conferences, and symposiums relating to software engineering practices from 1993 to the present. The

reference published in 2007 is current for the subject and highly relevant to the research questions identified for the study.

Soundararajan, S., & Arthur, J. (2010). *A structured framework for assessing the "goodness" of agile methods.* n.p.

Abstract. Agile methods are designed for customization; they offer an organization or a team the flexibility to adopt a set of principles and practices based on their culture and values. While that flexibility is consistent with the agile philosophy, it can lead to the adoption of principles and practices that can be sub-optimal relative to the desired objectives. The authors question, how can one determine if adopted practices are "in sync" with the identified principles, and to what extent those principles support organizational objectives? In this research, the authors focus on assessing the "goodness" of an agile method adopted by an organization based on (1) its adequacy, (2) the capability of the organization to provide the supporting environment to competently implement the method, and (3) its effectiveness. To guide their assessment, the authors propose the Objectives, Principles and Practices (OPP) framework. The design of the OPP framework revolves around the identification of the agile objectives, principles that support the achievement of those objectives, and practices that reflect the "spirit" of those principles. Well-defined linkages between the objectives and principles and between the principles and practices are also established to support the assessment process. The authors traverse these linkages in a top-down fashion to assess adequacy and a bottom-up fashion to assess capability and effectiveness. This is a work-in-progress paper, outlining the authors proposed research, preliminary results and future directions.

Comments. This article is highly relevant to the study because the authors address the application of agile principles in relation to organizational objectives and culture. Using a more comprehensive assessment process the authors propose a method that assesses an

organization not based on their level of agility but rather the people, process, project, and product characteristics of the organization adopting agile methods. The framework facilitates the identification of desirable objectives embraced by the agile philosophy and definitively links them to agile principles that support the achievement of those objectives. The article is deemed credible because the authors are affiliated with the computer science department at the highly accredited Virginia Tech University and one of the authors has published multiple works on the subject of software development engineering in several peer reviewed accredited journals and conference proceedings. The reference published in 2010 is very current for the subject and addresses the research questions identified for this study.

Srinivasan, J., & Lundqvist, K. (2009). Using agile methods in software product development: A case study. *Sixth International Conference on Information Technology: New Generations, Las Vegas, NV*, 1415-1420. doi.ieeecomputersociety.org/ 10.1109/ITNG.2009.334

Abstract. The mythos surrounding the use of agile methods emphasizes improved customer satisfaction, developer morale, and end-product quality. While the difficulty of adopting these methods is mentioned, it is often glossed over in the discussion. This paper presents an in-depth case study of agile methods adoption in a software product development firm. The choice of the firm as the unit of analysis enables the identification of organizational, social and technological challenges with respect to using agile methods. Using a mix of interviews, observation and archival data, the evolution of agile adoption within the firm is reconstructed. The data analysis highlights the importance of the four areas of requirements management, scrum implementation, organizational learning, and verification & validation activities.

Comments. This article is relevant to the study because the authors address several of the cultural challenges an organization may face when making the transition to agile methods. The reference is based on a case study following the evolution of a software development company's eight year transition to agile for the purpose of identifying and highlighting cultural challenges worthy of consideration by an organization adopting agile methods. The article is deemed credible because the authors are associated with the accredited Malardalen University in Sweden and have published multiple conference proceedings relating to software engineering practices from 1999 to the present. This reference published in 2009 is current for the subject and highly relevant to the research questions identified for the study.

Strode, D., Huff, S., & Tretiakov, A. (2009). The impact of organizational culture on agile method use. *Proceedings of the Hawaii International Conference on System Sciences, Waikoloa, HI, 42,* 1 – 9. doi.ieeecomputersociety.org/10.1109/HICSS.2009.952

Abstract. Agile method proponents believe that organizational culture has an effect on the extent to which an agile method is used. Research into the relationship between organizational culture and information systems development methodology deployment has been explored by others using the Competing Values Framework (CVF). However this relationship has not been explored with respect to the agile development methodologies. Based on a multi-case study of nine projects we show that specific organizational culture factors correlate with effective use of an agile method. Our results contribute to the literature on organizational culture and system development methodology use.

Comments. This article is critical to the study because it evaluates the impact of organizational culture on the adoption of agile methods. The authors evaluate nine organizations with varying cultures and identify the impacts culture has on the adoption

of agile methods. The article is deemed credible because the three authors are repeatedly published in several accredited peer reviewed journals and multiple conference proceedings on subjects relating to software engineering practices from 1985 to the present. The reference published in 2009 is current for the subject and highly relevant to the research questions identified for the study.

Tate, K. (2006). *Sustainable software development: An agile perspective.* Upper Saddle River, NJ: Addison-Wesley.

Abstract. This book describes the principles and practices required for change embracing technical excellence; these principles and practices promote sustainability and result in faster development with less effort through having a consistently low cost of change. For teams who are used to unsustainable development, the experience is best characterized as liberating because they are able to deal with change, not afraid of it or view it as a risk.

Comments. This book is relevant to the study because the author directly addresses the concept of applying agile principle and identifies the cultural factors that are desirable for an organization adopting agile principles. The author provides a set of practices for application by the organization adopting agile methods in support of a successful transition. The book is deemed credible because the author has more than 20 years of professional experience in the software development industry including development, methodology, product architecture, and technology strategy. The book published in 2006 is current for the subject and the reference content is highly relevant to the research questions identified in this study.

Tolfo, C. & Wazlawick, R. S. (2008). The influence of organizational culture on the adoption of extreme programming. *The Journal of Systems and Software, 8*, 1955-1967.

Abstract. The adoption of extreme programming (XP) method requires a very peculiar cultural context in software development companies. However, stakeholders do not always consider this matter and tend to stand to technical requirements of the method. Hence this paper aims at identifying aspects of organizational culture that may influence favorably or unfavorably the use of XP. In order to identify those aspects, this study analyzes dimensions of organizational culture under the perspective of practices and values of XP. This paper is based on the review of the literature of the area and empirical observations carried out with six software companies. This study does not intend to develop a tool for measurement of XP's compatibility with the organizational culture of each company. It intends to provide parameters (favorable and unfavorable aspects) for previous consideration of the convenience of XP implementation.

Comments. This article is critical to the study as it directly addresses the influences that organizational culture has on the adoption of agile principles. The authors evaluate the practices and behaviors of six software development teams in order to determine the factors of organizational culture that are favorable and unfavorable to adopting agile methods. The article is deemed highly credible because it is published in a peer reviewed and accredited software industry journal and co-authored by a PhD student and dean of undergraduate and graduate computer science program from the accredited Federal University of Santa Catarina in Brazil. The literature is published in January, 2008 making it current for the subject and highly relevant because it directly addresses the research questions identified for the study.

Vinekar, V., & Huntley, C. (2010). Agility versus maturity: Is there really a trade-off?. *Computer*, *43*(5), 87-89. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=51228468&site=ehost-live&scope=site

Abstract. The article focuses on the reasons why software developers are not following agile or formal methods as traditionally conceptualized. It cites the challenges faced by organizations in switching to a purely formal approach and a purely agile one. It mentions the increasing demand for the so-called agile tools to meet the needs of top management. According to emerging empirical evidence, most agile teams utilize some upfront design while most formal methods are iterative.

Comments. This article is very important to the study because it addresses the cultural nature of the structured, traditional mechanistic organization and the flexible organic organization in relation to the values and corresponding principles of agile. The authors review a new approach to address the cultural challenges different types of software development organizations face when adopting agile methods. This reference is highly relevant for the area of research and is valid for data coding because it addresses the research questions defined for the study. The article is deemed credible because it is published in a peer reviewed industry accredited journal and co-authored by two professors of information systems and operations management from the accredited Charles F. Dolan School of Business, Fairfield University. The article is published in May, 2010 making it very current for the subject.

Vinekar, V., Slinkman, C., & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management*, *23*(3), 31-42.

Abstract. Emerging evidence seems to indicate that most systems development organizations are attempting to utilize both agile and traditional approaches. This study aims to understand the reasons organizations feel the need for this unlikely juxtaposition and the organizational challenges in sustaining the opposing cultures. Drawing on the extensive literature in organizational theory and management, we advocate ambidexterity

as a viable solution to systems development organizations attempting to harness the benefits of both agile and traditional development.

Comments. This article is critical to the study because it defines a unique approach to the cultural challenges traditional organizations face when adopting agile methods. The authors address the fact that cultural change is a highly significant change for an organization that requires a high level of learning and can take years to complete. Instead of a full blown transition to agile the authors recommend a dual methodology approach based on project attributes. This article is deemed credible because it is published in a peer reviewed industry accredited journal and co-authored by three university professors from the accredited University of Texas at Arlington with academic credentials in the area of information systems development. The reference published in 2006 is current for the subject and highly relevant to the research questions identified for the study.

Review of the Literature

Cultural Factors within Organizations that Influence the Adoption of Agile Principles

The review of the literature provides a meaningful examination of the role played by organizational culture within a traditional software development organization when adopting agile software development principles and the cultural changes necessary for the organization to successfully adopt such principles. Organizational culture is defined as beliefs, attitudes, values, and behaviors, and affects a range of organizational elements, including (a) the organizational structure, which impacts the management style and practices, and (b) the physical setting, which impacts interactions of individuals and teams (Robinson & Sharp, 2005).

Organizational culture represents the organizational norms and pressures that facilitate the use of agile methods (Chan & Thong, 2008). According to Chan and Thong (2008) the important cultural factors that affect the adoption of agile methodologies can be summarized by the following categories: teamwork, individual ability, motivation, management support, communication, leadership, management style, management of software development knowledge, reward systems, and customer relationships. Factors are addressed in detail in this section under the headings of:

- Management style and practices
- Project management style and practices
- Communication practices
- Employee work habits and practices
- Development team practices
- Knowledge management practices
- Customer expectations

Cultural Factors in Traditional Software Development Organizations

Traditional software development is characterized by a prescriptive approach requiring the creation of many non-software artifacts, primarily documentation, during development (Strode et al., 2009). Traditional, plan-driven software development organizations have cultural aspects that put constraints on the development team, limiting the development practices they can and must use, which affects how quickly they can develop software (Lindvall et al., 2004).

Management style and practices inherent to the traditional software development organizational culture. Traditional software development characterized by methods of planning and control is accomplished by a command and control management style (Nerur et al., 2005). Traditional software development organizations follow a management style characterized by well documented plans accompanied by performance measures considered key to the success of their development practices (Misra, Kumar, & Kumar, 2009). Traditional software development management specifies tasks to be completed, desired outcomes by phase, and assigns roles to those individuals that perform the tasks (Nerur et al., 2005). Traditional managers associate employees to specific roles and expect employees to perform within the boundaries of their roles (Boehm & Turner, 2005). Traditional software development provides assurances by following compliance driven processes and activities that are measurement based (Nerur et al., 2005).

Project management style and practices inherent to the traditional software development organizational culture. Traditional software development organizations manage projects with a plan that follows a schedule with deadlines, milestones, and a budget (Vinekar & Huntley, 2010). The role of the traditional software development project manager is that of planner and controller directing the activities of the software development team (Nerur et al., 2005).

Communication practices inherent to the traditional software development organizational culture. Traditional software development organizations utilize large volumes of detailed documentation in order to overcome the requirements of communicating with a large number of members (Misra et al., 2009). According to Nerur et al. (2005) traditional software development "produces a large amount of documentation that codifies process and product knowledge" (p. 75) and supports the formal communication process used by project participants.

Employee work habits and practices inherent to the traditional software development organizational culture. Traditional software development methods assume that customers do not know their requirements and developers do (Chan & Thong, 2008) therefore there is little interaction between the development teams and customers in traditional software development organizations (Bose, 2008).

Development team practices inherent to the traditional software development organizational culture. Traditional software development teams attempt to minimize change through rigorous requirements gathering, analysis, and design (Vinekar et al., 2006). Developers rely on product requirements that are predictable and stable (Vinekar et al., 2006). Following traditional software development methods developers expect a detailed specification document from which to build the software (Chan & Thong, 2008). Developers in traditional software development organizations assume customers are short sighted and build in extra functionality to meet future needs (Chan & Thong, 2008). Traditional software developers follow a practice in which they write code upfront and test after the code is written (Nerur et al., 2005).

Knowledge management habits and practices inherent to the traditional software development organization. Traditional software development methods produce large amounts of documentation that contain knowledge about development processes (Bose, 2008). According to Nerur et al. (2005), documentation within traditional software development organizations "serves as useful artifacts for communication and traceability of design" (p. 76).

Customer expectations inherent to the traditional software development organization. Traditional software development organizations are highly dependent upon the customer's ability to identify requirements upfront (Vinekar et al., 2006). Traditional software development relies heavily on the customer during specification development with minimal participation during all other development activities (Nerur et al., 2005). User participation is minimal during most traditional software development activities except when specification development is in process (Chan & Thong, 2008).

Cultural Factors in Agile Software Development Organizations

Agile software development is an emerging software engineering approach constituting of a set of principles based on best practices derived from the success and failure of many software development projects (Misra et al., 2009). Agile software development is characterized by the following principles, as cited by Misra et al. (2009):

a. Close collaboration between the software development team and the business team
b. Face to face communications rather than extensive written documentation
c. Frequent delivery of a portion of software instead of the final delivery of a complete product
d. Acceptance of changing requirements rather than defining a fixed set of requirements
e. Adaptive capability of teams in response to changing business requirements

According to Lindvall, cited by Misra et al. (2009), "To be agile is a cultural thing. If the culture is not right, then the organization cannot be agile" (p. 1880).

Management style and practices inherent to agile software development organizational culture. Agile software development follows a management style that favors leadership and collaboration (Nerur et al., 2005). The agile development team manager spends considerable time "supporting, facilitating and orchestrating activity rather than explicitly managing or controlling activity" (Robinson & Sharp, 2005, p. 52). Agile software development management

practices place a significant emphasis on the role that people play in software development (Chan & Thong, 2008). Agile team members are empowered by their managers with more discretionary and decision making ability allowing them to organize according to interest and skill and not be confined by a specific role (Nerur et al., 2005). Agile software development is characterized by a management style of internalized plans and qualitative controls prepared and monitored by the team rather than external managers (Misra et al., 2009). According to Misra et al. (2009) agile organizations reflect a management style that is supportive, lacks organizational politics, has good customer relationships, and demonstrates adaptability to change.

Project management practices inherent to agile software development organizational culture. Agile software development project managers facilitate and coordinate the activities of the development team (Nerur et al., 2005). Project managers in an agile organization perform in the role of protector and coach creating a barrier between the organization and team to minimize disturbance and provide aide when technical help is needed (Boehm & Turner, 2005).

Communication practices inherent to agile software development organizational culture. Agile software development is an iterative and incremental process that relies on high levels of communication and regular interactions (Strode et al., 2009). Communication and cooperation is very important among the members of an agile software development team in order to establish accurate requirements and feedback from customers (Qumer & Henderson-Sellers, 2008). Agile software development organizations utilize rapid communication methods in order to make major decisions quickly and respond to change (Misra et al., 2009). The ambition of agile software development is to reduce the number of documents produced by the development team as "many documents are unnecessary because they are quickly outdated while other documents can be replaced by direct communication" (Petersen & Wohlin, 2009, pg. 1487).

Employee work habits and practices inherent to agile software development organizational culture. Agile software development principles rely on the importance of the people doing the work and their interactions (Bose, 2008). Relying on people and the interactions of those people is the cornerstone of agile principles (Sidky, Arthur, & Bohner, 2007). Agile emphasizes the work of the team and the intense dynamics of team interactions (Vinekar et al., 2006). Agile development teams thrive on the formal and informal interactions between the people involved in doing the development work, which is necessary for building trust (Bose, 2008). Agile software development team members are physically located together and communicate intensively during frequent face-to-face meetings increasing product understanding and knowledge significantly (Petersen & Wohlin, 2009). Collocated teams are recognized as an important vehicle for successful communications and are one of the most important factors for successful agile software development (Misra et al., 2009). Agile software "development is characterized by social inquiry in which extensive collaboration and communication provide the basis for collective action" (Nerur et al., 2005, p. 75). According to Sidky, Arthur, and Bohner (2007) collaboration is the dimension that sets the foundation for agile software development.

Development team behaviors inherent to the agile software development organizational culture. Agile software development work is done by small teams, less than 25 is considered small, in a collaborative fashion (Nerur et al., 2005). Agile developers program in pairs, insuring constant inspection of code quality (Bose, 2008). Agile development teams have frequent and often unplanned face-to-face interactions among team members (Bose, 2008). Agile team members are motivated, eager to learn from each other, honest, collaborative, and responsible (Misra et al., 2009). A key trait of agile software development teams is the ability to self-organize (Vinekar et al., 2006). Agile development teams follow pluralistic decision making that involves diverse stakeholders in a collaborative environment fostered by strong leadership (Vinekar et al., 2006).

Agile software development manages the unpredictability associated with developing software by relying on team members and their creativity rather than strict process (Nerur et al., 2005). Agile development teams work in short iterative cycles, make collaborative decisions about the work they will complete, and incorporate rapid feedback and change (Nerur et al., 2005). Agile software development teams do not measure progress but rather assess and adjust based on the remaining work to be completed (Nerur et al., 2005). Agile development practices are minimally defined and adaptive and require regular customer interaction from the beginning for providing feedback about the product and process (Bose, 2008).

Agile software development puts a premium on the practice of testing upfront and urges developers to test code upfront (Nerur et al., 2005). Agile methods promote early and frequent delivery of well-tested software (Kane et al., 2006). Utilizing agile development methods the developer is focused on getting the code written, tested, and accepted by the customer utilizing pair programming in small releases (Grossman et al., 2004). Agile software development teams rely on teamwork to complete the work required to deliver the finished product (Nerur et al., 2005).

Knowledge management habits and practices inherent to the agile software development organization. Qumer and Henderson-Sellers (2008) state that "agile software development is a knowledge-intensive process" (p. 1901) that requires the software development team to practice behaviors where knowledge is created and shared. According to Nerur et al. (2005) agile software development "encourages lean thinking and cutting down on overhead" (p. 76), which results in an organization creating far less documentation. Proper and lengthy documentation is less important because requirements are expected to be changing (Bose, 2008). Agile software development discourages documentation beyond the actual code which encourages the development and sharing of tacit product knowledge between team members expected to rotate between development roles (Nerur et al., 2005). Nerur et al. (2005) assert that

agile software development by nature fosters an environment of learning and adaptation as a result of "repeated cycles of thought-action-reflection" (p. 75) that occurs among diverse stakeholders, customers, developers, and end users.

Customer expectations inherent to the agile software development organization. Agile software development is done with customers who are actively engaged and knowledgeable about their product (Nerur et al., 2005). Customers are expected to play a much more active role throughout the agile software development process (Chan & Thong, 2008). Nerur et al. (2005) emphasize that agile software development customers must actively participate in the development process and are expected to be "collaborative, representative, authorized, committed, and knowledgeable" (p. 76). Customers play a critical role during agile development and the success of agile development hinges on finding users who actively participate in the development process (Chan & Thong, 2008). Agile software development is highly dependent on the on-site customer identifying and prioritizing requirements, providing feedback, and guiding change (Vinekar et al., 2006).

Areas of Conflict Between Agile and Traditional Software Development Organizations

Organizational culture is an important organizational characteristic found to affect the acceptance of agile methodologies (Chan & Thong, 2008). Integrating agile software development into an organization with a culture that favors more traditional software development can be difficult (Lindvall et al., 2004). There are significant differences between the cultures and practices of traditional and agile software development organizations which lead to numerous challenges for organizations transitioning from traditional to agile methods (Chan & Thong, 2008).

Agile methods conflict with traditional methods related to management planning and control, role assignments among the development team, the customer's role, and use of technology (Chan & Thong, 2008). Agile software development differs from traditional software

development in the sense that it is deeply focused on human relationships (Tolfo & Wazlawick, 2008). Agile software development is characterized by a high degree of uncertainty that arises from frequent and changing requirements (Vinekar et al., 2006). As a result, agile software development is often viewed by traditional software development organizations as chaotic and lacking in formal procedural rigor (Vinekar et al., 2006).

The right corporate culture is perceived by agile experts to be a fundamental necessary condition for agile adoption (Misra et al., 2009). The following section identifies the cultural factors that are in conflict with the culture of the traditional software development organization adopting agile principles.

Management style and practices in cultural conflict between agile and traditional software development organizations. One of the biggest hurdles for a traditional organization adopting agile methods is the perceived lack of stability in process inherent to the management style of traditional software development organizations (Nerur et al., 2005). Unlike agile, traditional software development is focused on well-defined plans and detailed documentation (Misra et al., 2009). Traditional software development provides assurances by following compliance-driven processes and activities that are measurement based (Nerur et al., 2005). Agile software development relies on speculation and planning that is based on uncertainty to "guide the rapid development of flexible and adaptive systems of high value" (Nerur et al., 2005, p. 77).

The agile principle requiring software development teams to be empowered can be seen as a threat to managers within traditional organizations, and requires sufficient training for manager to acquire the skill to change their management style and practice (Petersen & Wohlin, 2009). Traditional organizations making the shift from *command and control* to *leadership and collaboration* need to facilitate an organizational form that has the right blend of autonomy and

cooperation to achieve synergy while providing flexibility and responsiveness (Nerur et al., 2005).

Project management practices in cultural conflict between agile and traditional software development organizations. The biggest challenge for the traditional project manager moving to agile software development is giving up the decision making authority they previously enjoyed while managing traditional software development projects (Nerur et al., 2005). Traditional software development project managers are not accustomed to dealing with the technical issues associated with a shift in the product implementation schedule and must adjust their expectations and actions to support the change in practice (Petersen & Wohlin, 2009).

Communication practices in cultural conflict between agile and traditional software development organizations. Traditional software development teams made up of a large number of members post great hindrance to fast communications and decisions (Misra et al., 2009). Organizations with the right culture for agile adoption support "rapid communications, dynamicity in requirements changes, trusting people, and obtaining fast feedback from customers" (Misra et al., 2009, p. 1880).

Employee work habits and practices in cultural conflict between agile and traditional software development organizations. Unlike traditional methods of software development, agile puts a significant emphasis on the role that people play in software development (Chan & Thong, 2008). Agile software development places a premium on the people in the organization and their interactions (Vinekar et al., 2006). Agile principles are most successful in organizations where people are used to working collectively than in organizations where that is not the case (Misra et al., 2009). Agile is extremely suitable to dynamic organizations with a culture that is highly adaptive to change and supportive of working in a collaborative environment (Misra et al., 2009). It may take a traditional software development organization "enormous effort, time, and

patience to build a culture of trust and respect among its employees" (p. 76) necessary for the collaborative decision making of agile (Nerur et al., 2005).

Development team habits and practices in cultural conflict between agile and traditional software development organizations. Software development process change represents a more radical change for an organization than adopting new tools or techniques (Chan & Thong, 2008). According to Vinekar et al. (2006) "the roles of agile team members are interchangeable, and developers often choose roles that are not in their area of specialty" (p. 34). Merging agile processes with traditional processes is one of the most significant areas of difficulty for organizations (Boehm & Turner, 2005). The agile principle of pair programming is perceived by traditional software developers as exhaustive and inefficient (Petersen & Wohlin, 2009). Pair programming as a practice requires that developers be accepting of criticisms, suggestions, and have the ability to follow each other's instructions (Bose, 2008).

Knowledge management habits and practices in cultural conflict between agile and traditional software development organizations. Traditional organizations heavily dependent on documentation may struggle with the lack of explicit knowledge available with agile (Nerur et al., 2005). Knowledge management is of particular importance for agile and quite different than what is practiced among traditional software development organizations (Chan & Thong, 2008). Agile methods discourage the use of formal documentation used by traditional organizations to transfer product knowledge therefore product knowledge becomes tacit and its transfer relies on the rotation of team members (Chan & Thong, 2008). The capability to create and utilize knowledge among the development team is critical since agile methods "emphasize individual competence, constant communication, and close collaboration between developers and customers" (Chan & Thong, 2008, p. 808). The organizational culture that does not support teamwork will impede knowledge exchange necessary for a successful agile adoption (Tolfo & Wazlawick, 2008).

Customer expectations in conflict between agile and traditional software development organizations. Traditional customers take on a significantly more active role within the agile project team and must be prepared to make a commitment to the entire development process (Petersen & Wohlin, 2009). The relationship with the customer is an important consideration for organizations adopting agile methods (Chan & Thong, 2008). The more active customer participates throughout the agile software development process whereas customer participation is minimal in most activities except specification development in traditional software development methods (Chan & Thong, 2008). The role of the customer (the person representing the user community) changes with agile development from participating in requirements gathering to becoming an active member of the development team (Chan & Thong, 2008).

One of the adoption challenges for a traditional software development team will be to locate and find customers that are willing to make the commitment to the development process and that have the knowledge necessary to meet the requirements of agile (Nerur et al., 2005). According to Robinson and Sharp (2005) traditional software development organizations must rethink their expectations regarding how customers interact with the development team in order to successfully adopt agile principles.

Cultural Changes Necessary for a Traditional Software Development Organization When Adopting Agile Principles

Agile software development assumes change is inevitable and aims at achieving individual initiatives through adaptation and innovation (Vinekar et al., 2006). Where the culture is not innovative and is risk adverse, the traditional software development organization must foster these values necessary to support the change in work routines as a result of an agile adoption (Tolfo & Wazlawick, 2008). Traditional software development organizations with a culture conducive to innovation may embrace agile principles more easily than those "built around bureaucracy and formalization" (Nerur et al., 2005, p. 78). Cultural aspects, notably

individual and organizational characteristics, play a critical role in the acceptance and adoption of agile principles (Chan & Thong, 2008). Organizational culture and the deployment and effective use of agile software development principles are related (Strode et al., 2009).

Traditional software development organizations considering agile should be cognizant of the fact that adaptation is not an easy task and requires organizational members change attitude and values (Tolfo & Wazlawick, 2008). Traditional organizations require a shift from a culture that is primarily process driven to one that is people driven (Bose, 2008). The shift in focus from process to people is a significant departure for the traditional software development organization (Vinekar et al., 2006). Traditional organizations adopting agile principles must rethink their goals and reconfigure their human and managerial components to be successful (Nerur et al., 2005). The most important challenge for an organization adopting agile principles is the "development and encouragement of an agile culture and mindset" (Qumer & Henderson-Sellers, 2008, p. 1911). The following section identifies the cultural changes required for the traditional software development organization adopting agile principles.

Management style and practice changes required by the traditional software development organization adopting agile principles. Karlstroem and Runeson report, as cited by Misra et al. (2009), that despite some initial management resistance it is definitely possible to adopt agile principles in traditional software development organizations. To succeed at an agile adoption requires traditional software development organizations to change from a management style of plan and control to leadership and collaboration (Misra et al., 2009). Management practices for traditional organizations must shift from controlling to coaching and protecting in order to shelter the development team from the demands of the remainder of the organization (Boehm & Turner, 2005). Traditional, plan-driven organizations that choose to adopt agile principles have to adjust how daily business is conducted to allow for the greater levels of uncertainty and ambiguity that exist in evolutionary and iterative methods like agile (Boehm &

Turner, 2005). Traditional managers planning to adopt agile methods can no longer associate employees to specific roles and need to be open to the multitasking characteristics of agile development teams (Boehm & Turner, 2005).

Project management style and practice changes required by the traditional software development organization adopting agile principles. The traditional software development project manager must alter their role to that of a facilitator who now directs and coordinates the collaborative efforts of the development team (Nerur et al., 2005). Project managers become protector and coach creating a barrier between the organization and team to minimize disturbance and provide aide when technical help is needed (Boehm & Turner, 2005).

Communication practice changes required by the traditional software development organization adopting agile principles. According to Nerur et al. (2005) traditional organizations must develop a social process norm "characterized by communication and collaboration between a community of members who value and trust each other" (p. 76) to successfully adopt agile principles. Traditional software development teams should learn to identify changes and reflect on changes quickly by using a face-to-face communication and feedback approach (Qumer & Henderson-Seller, 2008).

Employee work habits and practice changes required by the traditional software development organization adopting agile principles. Nerur et al. (2005) emphasize that traditional software development programmers accustomed to solitary activities and work must adjust to the idea of "shared learning, reflection workshops, pair programming, and collaborative decisions" (p. 76) inherent with agile. Agile development depends on individuals working closely together requiring traditional organizations to create a workspace with pair-programming stations, plenty of wall space to track status and assignments, collaboration areas, and equipment to support product testing (Boehm & Turner, 2005). Agile development requires developers and customers work in teams and collaborate in order to achieve higher productivity (Chan & Thong,

2008). Agile development teams are smaller (Bose, 2008). It is suggested that large traditional teams adopting agile methods be broken down into a few smaller teams in order to efficiently coordinate and manage communications and interactions (Misra et al., 2009).

Development team habits and practice changes required by the traditional software development organization adopting agile principles. Traditional software development teams must overcome the practice of writing code upfront in order to institutionalize the practice of early and frequent testing (Nerur et al., 2005). Agile methods change the way developers learn about the software product (Chan & Thong, 2008). Agile methods emphasize simplicity and the art of maximizing unnecessary work not being done (Chan & Thong, 2008). Successful agile development teams are normally left on their own allowing them to make their own decisions and succeed (Misra et al., 2009). Traditional software development teams should be encouraged to organize and manage themselves but remain accountable and responsible for project deliverables (Qumer & Henderson-Sellers, 2008). Nerur et al. (2005) highlight that traditional software development teams adopting agile principles require "major alterations to work procedures, tools, techniques, communication channels, problem solving strategies, and the roles of people" (p. 77).

Knowledge management habits and practice changes required by the traditional software development organization adopting agile principles. Knowledge management is of particular importance for agile (Chan & Thong, 2008). Tacit and explicit knowledge about agile methods is important to the success of agile development; it has to be retained by the development team and transferred effectively among team members (Chan & Thong, 2008). Organizational culture is a motivation-related factor relevant to knowledge management aspects of software development in understanding individual acceptance of agile methodologies (Chan & Thong, 2008). Knowledge creation, retention, and transfer are interrelated and determine how successful development team members manage the knowledge required for using agile

methodologies (Chan & Thong, 2008). Organizational culture that emphasizes knowledge sharing can encourage knowledge management behaviors and improve the organizations acceptance of agile methods (Chan & Thong, 2008).

Customer expectation changes required by the traditional software development organization adopting agile principles. Factors associated with customers are crucial to the adoption of agile methods (Chan & Thong, 2008). Organizations need to prepare their stakeholders, especially customers, for the significantly changed role suggesting they be onsite interacting directly with the team, providing regular feedback, and participating in acceptance testing (Boehm & Turner, 2005). Boehm and Turner (2005) emphasize that "attention to process matching and customer education is necessary to smooth the transition" (p. 38) for the traditional software development customer. Delivering software early and continuously according to agile methods requires that customers are on-site with the development team, highly motivated, active, and consider themselves key to the success of the project (Misra et al., 2009).

Conclusion

Organizational culture should be carefully considered by the traditional software development organization in preparation for a successful agile adoption (Strode et al., 2009). The authors of agile software development methods clearly state that the organizational culture in which the agile adoption is planned could have an impact on its use (Strode et al., 2009). The differences between agile and traditional software development organizations raise a number of challenges for a successful transition (Chan & Thong, 2008). Merging agile software development principles with traditional software development principles is one of the most significant areas of difficulty for organizations (Boehm & Turner, 2005). Migrating to agile involves issues pertaining to management, people, process, and technology (Chan & Thong, 2008). Table 1, extracted from Nerur et al. (2005), highlights the key cultural differences between traditional and agile software development. The table is provided to assist the traditional software development organization with identifying the cultural factors that may impact the successful adoption of agile principles.

Table 1

Summary of Cultural Differences Between Traditional and Agile Software Development Organizations

	Traditional software development	Agile software development
Fundamental cultural assumptions	Software is specifiable, predictable, and can be built based on meticulous planning.	High quality software is developed by small teams using continuous improvement and testing based on rapid feedback and change
Control	Process focus	People focus

Management style	Command and control	Leadership and collaboration
Knowledge management	Explicit	Tacit
Role assignment	Individual and specialized	Self organizing with role interchangeability encouraged
Communication	Formal	Informal
Customer role	Important	Critical
Project cycle	Driven by tasks and activities	Driven by required product features
Organizational form and structure	Bureaucratic and highly formalized	Flexible, informal and participative

While varied software development methodologies have been employed in traditional organizations for years, the adoption of an agile software development method poses cultural challenges that demand change in management style, communication, collaborative practices among project members, and technological infrastructure (Chan & Thong, 2008). Following are the key changes related to management, people, and process that are recommended for the traditional software development organization when planning to adopt agile principles.

Changes in management style and practice required for adopting agile principles. The success of an agile adoption depends substantially on management (Qumer & Henderson-Sellers, 2008). To succeed at an agile adoption requires traditional software development organizations adopt a management style of leadership and collaboration (Misra et al., 2009) and follow management practices that are coaching and protecting (Boehm & Turner, 2005). Traditional software development management has to adjust how daily business is conducted to prepare the

organization for greater levels of ambiguity and uncertainty (Boehm & Turner, 2005). Traditional managers can no longer associate employees to specific roles and must be open to employee multitasking (Boehm & Turner, 2005).

The organization interested in adopting agile should be concerned with human capital, valuing policies that motivate employees to perform (Tolfo & Wazlawick, 2008). The traditional software development organization adopting agile should reevaluate reward systems to encourage collective goals over individual accomplishments (Vinekar et al., 2006). Traditional software development organizations should align their performance management in favor of teamwork and team accomplishments to successfully adopt agile principles (Nerur et al., 2005). Performance evaluation systems in the agile organization need to echo teamwork and evaluations should produce "a sense of union, collective effort, and concern with colleague difficulties, aiming at a good performance" (Tolfo & Wazlawick, 2008, p. 1963).

Changes in project management style and practice required for adopting agile principles. The traditional project manager performs in the role of facilitator and coach effectively managing the collaborative efforts of the team without stifling creativity (Vinekar et al., 2006). The traditional project manager following agile directs and coordinates (Nerur et al., 2005) acting as protector creating a barrier between the organization and team to minimize disturbance and provide aide when technical help is needed (Boehm & Turner, 2005).

Changes in communication required for adopting agile principles. The traditional software development team must learn to communicate fast and effectively among developers, operations, support, customers, management, and business areas in order to make and respond to changing requirements quickly (Misra et al., 2009). Qumer and Henderson-Sellers (2008) suggest that the traditional software development team overcome potential issues with the change in communication practices by utilizing a communication and cooperation protocol to

"specify and enable an effective face-to-face communication among the empowered, self-organizing and cross-functional team and their clients" (p. 1910).

Changes in employee attitudes required for adopting agile principles. Agile software development conception and evolution in the traditional software development organization depends on "acceptance, compromise and interactions among people" (Tolfo & Wazlawick, 2008, p. 1956). The competency of an agile development team is important since the goal of agile practices is delivering software fast (Misra et al., 2009). The processes followed by the development team should be flexible enough to be molded around the competencies of the people (Vinekar et al., 2006). Agile team members should have "real-world experience in the technology domain, have built similar systems in the past, and possess good interpersonal and communication skills" (Misra et al., 2009, p. 1872). Agile developers must be innovative, skilled, creative, and always prepared to manage the unexpected since they do not have a detailed long term plan to follow and it is expected that their customers may alter requirements at any time (Tolfo & Wazlawick, 2008). According to Misra et al. (2009) agile development team members must demonstrate "personal characteristics such as honesty, collaborative attitude, sense of responsibility, readiness to learn" (p. 1872), and a willingness to work closely with others. Agile software development requires careful developers who are detail-oriented and demonstrate pride in their work (Tolfo & Wazlawick, 2008). The people in traditional organizations are at the heart of the paradigm shift for a successful transition to agile, "the paradigm change is aimed at empowering individuals by supporting reasonable goals, shorter feedback cycles, ownership, and flexibility" (Boehm & Turner, 2005, p. 36).

Changes in development team design and practice required for adopting agile principles. Tolfo and Wazlawick (2008) highlight the importance of development team cohesion when adopting agile principles. Agile thrives on the formal and informal interactions between the people involved in doing the development work which is necessary for building trust (Bose,

2008). Collocation of the development team is necessary to support the required and frequent interactions of the customer and developers (Bose, 2008). Co-located teams are recognized as an important vehicle for successful communications and are one of the most important factors for successful agile software development (Misra et al., 2009). Robinson and Sharp (2005) highlight that the agile software development practice of pair programming is highly demanding and might be offset by providing developers with "gold card days where one could carry out some individually focused work that was of value to the company" (p. 56). According to Bossavit, cited by Tolfo and Wazlawick (2008), the organization adopting agile must enable the necessary conditions to support the methods such as the appropriate hardware and the adequate environment necessary to support pair programming.

Changes in knowledge management habits and practice required for adopting agile principles. Documentation can be reduced in software development organizations if there is a well-established communication-oriented culture where information is openly shared among team members (Qumer & Henderson-Seller, 2008). The development team has to acquire the knowledge of using agile methods before they can actually use the methods to guide the development of the software (Chan & Thong, 2008). Tacit and explicit knowledge about agile methods is important to the success of agile development; it has to be retained by the development team and transferred effectively among team members (Chan & Thong, 2008). Managing agile knowledge successfully increases team member confidence and removes the barriers to adopting agile methods (Chan & Thong, 2008).

Changes in customer expectation required for adopting agile principles. Customers that support agile software development principles are expected to participate actively in the software development process having direct contact with the development team (Tolfo & Wazlawick, 2008). Customers play a critical role in agile development and the success of agile development

hinges on finding customers who actively participate in the development process (Chan & Thong, 2008).

In summary, changing software development methods requires changing organizational culture; a cultural change of this type "can be harder than changing strategy, structures, processes, or tools and can take years" (Vinekar & Huntley, 2010, p. 87). Most discussions around the adoption of agile principles tend to gloss over the difficulties associated with the fundamental organizational change required to make the transition successful (Srinivasan & Lundquvist, 2009). Changing the culture of an organization is not trivial since it requires that members change the way they think, communicate, relate to each other, as well as alter work habits and working manners (Tolfo & Wazlawick, 2008). Tolfo and Wazlawick (2008) highlight that to successfully program changes in an organization; the dimensions of organizational culture should be considered.

Agile software development principles afford traditional software development organizations opportunities and benefits that make them attractive, but organizations must be circumspect when embracing or integrating them with existing practices (Nerur et al., 2005) to ensure a successful adoption. The existing culture of a software development organization has a potentially large impact on the adoption of agile software development principles (Qumer & Henderson-Sellers, 2008). According to Strode, Huff, and Tretiakov (2009), the greater the presence of the organizational culture factors considered necessary for an agile adoption, the higher the value achieved from utilizing agile software development principles.

References

The references section is a list of all works cited directly or used in any manner to support the development of the literature review (Ettinger, 2010). The list includes all references selected for use in the Annotated Bibliography and those that are used as support in support of the entire document.

AgileSherpa. (2010). What is agile development? Retrieved from http://agilesherpa.org /intro_to_agile/what_is_agile_development/

Anson et al. (2007). *Literature Reviews.* The Writing Center, University of North Carolina, Chapel Hill, NC. Retrieved from http://www.unc.edu/depts/wcweb/handouts/ literature_review.html

Bell, C., & Smith, T. (2009). Critical evaluation of information sources. Retrieved from http://libweb.uoregon.edu/guides/findarticles/credibility.html

Black, S., Boca, P., Bowen, J., Gorman, J., & Hinchey, M. (2009). Formal versus agile: Survival of the fittest? *Computer*, *42*(9), 37-45. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx? direct=true&db=aph&AN=44299242&site=ehost-live&scope=site

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software, 22*(5), 30-39. doi.ieeecomputersociety.org/10.1109/MS.2005.129

Bose, I. (2008). Lessons learned from distributed agile software projects: A case-based analysis. *Communications of AIS*, *2008*(23), 619-632. Retrieved from Computer Source Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true &db=cph&AN=41671742&login.asp&site=ehost-live&scope=site

Bozheva, T., & Gallo, M. (2005). Framework of agile patterns. *Software Process Improvement, Proceedings, 3792*(147915531), 4-15.

Busch, C., De Maret, P., Flynn, T., Kellum, R., Le, S., Meyers, B., Saunders, M., & White, R. (2005). *Content analysis*. Writing@CSU, Colorado State University Department of English, Fort Collins, Colorado. Retrieved from http://writing.colostate.edu/guides/research/content

Chan, F., & Thong, J. (2008). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems, 46*(4)*,* 803-814. doi: 10.1016/j.dss.2008.11.009

Chuan-Hoo, T., & Hock-Hai, T. (2007). Training future software developers to acquire agile development skills. *Communications of the ACM*, *50*(12), 97-98. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=27779974&site=ehost-live&scope=site

Cooper, H. (1998). Introduction. In C. Laughton, & A. Virding (Eds.), *Synthesizing research: A guide for literature reviews* (pp. 1-201). Thousand Oaks, CA: SAGE.

Creswell, J. (2009). Review of the literature. In V. Knight, S. Connelly, S. K. Quesenberry, & M. P. Scott (Eds.), *Research design: Qualitative, quantitative, and mixed method approaches* (23-46). Thousand Oaks, CA: SAGE.

De Cesare, S., Lycett, M., Macredie, R., Patel, C., & Paul, R. (2010). Examining perceptions of agility in software development practice. *Communications of the ACM*, *53*(6), 126-130.

Dybå, T., & Dingsoyr, T. (2009). What do we know about agile software development? *IEEE Software*, *26*(5), 6-9. Retrieved from Academic Search Premier Database.

Ettinger, L. (2010). Terminal project: Writing guides. Retrieved January 2, 2011, from University of Oregon, Applied Information Management Program: https://blackboard.

uoregon.edu/webapps/portal/frameset.jsp?tab_tab_group_id=_2_1&url=%2Fwebapps%2Fblackboard%2Fexecute%2Flauncher%3Ftype%3DCourse%26id%3D_295075_1%26url%3D

Fruhling, A., & de Vreede, G. (2006). Field experiences with extreme programming: Developing an emergency response system. *Journal of Management Information Systems*, *22*(4), 39-68.

Grossman, F., Bergin, J., Leip, D., Merritt, S., & Gotel, O. (2004). One XP experience: Introducing agile (XP) software development into a culture that is willing but not ready. *Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, Canada,* 242 – 254. Abstract retrieved from http://portal.acm.org.libproxy.uoregon.edu/citation.cfm?id=1034914.1034933&coll=DL&dl=GUIDE&CFID=4119751&CFTOKEN=14492728

Hasnain, E., & Hall, T. (2009). Introduction to stand-up meetings in agile methods. *AIP Conference Proceedings*, *1127*(1), 110-120. doi:10.1063/1.3146182.

Hewitt, M. (2002). Trent focus for research and development in primary health care: Carrying out a literature review. *Trent Focus Group,* 1-27. http://ce.uoregon.edu/aim/Capstone07/HewittLitReview.pdf

Kane, D., Hohman, M., Cerami, E., McCormick, M., Kuhlmman, K., & Byrd, J. (2006). Agile methods in biomedical software development: a multi-site experience report. *BMC Bioinformatics*, *7*273-12. doi:10.1186/1471-2105-7-273. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=28833806&site=ehost-live&scope=site

Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, *34*(1), 87-114. Retrieved from Computer Source Database. http://search.ebscohost.com.libproxy.uoregon.edu/

login.aspx?direct=true&db=cph&AN=48477462&login.asp&site=ehost-live&scope=site

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., & Kahkonen, T. (2004). Agile software development in large organizations. *Computer*, *37*(12), 26-34. Retrieved from Academic Search Premier database. http://search.ebscohost.com.libproxy. uoregon.edu/login.aspx?direct=true&db =aph&AN=15586953&site=ehost-live&scope=site

Mann, C., & Maurer, F. (2005). A case study on the impact of scrum on overtime and customer satisfaction. In *Proceedings of the Agile Development Conference* (ADC '05). IEEE Computer Society, Washington, DC, USA, 70-79. DOI=10.1109/ADC.2005.1 http://dx.doi.org/10.1109/ADC.2005.1

Miranda, E., & Bourque, P. (2010). Agile monitoring using the line of balance. *Journal of Systems & Software*, *83*(7), 1205-1215.

Misra, S., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869-1890. doi: 10.1016/j.jss.2009.05.052

Mohan, K., Ramesh, B., & Sugumaran, V. (2010). Integrating software product line engineering and agile development. *IEEE Software*, *27*(3), 48-55.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 73-78. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/ login.aspx?direct=true&db=aph&AN=16915874&site=ehost-live&scope=site

Obenzinger, H. (2005). What can a literature review do for me? http://ce.uoregon.edu/aim/ Capstone07/LiteratureReviewHandout.pdf

Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems & Software*, *82*(9), 1479-1490.

Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems & Software*, *81*(11), 1899-1919.

Robinson, H., & Sharp, H. (2005). Organisational culture and XP: three case studies. *Proceedings of the Agile Development Conference,* Denver, CO, 49-58. doi.ieeecomputersociety.org/10.1109/ADC.2005.36

Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: The agile adoption framework. *Software Engineering,* n.p. Retrieved from arXiv: 0704.1294v1

Soundararajan, S., & Arthur, J. (2010). *A structured framework for assessing the "goodness" of agile methods.* n.p.:

Srinivasan, J., & Lundqvist, K. (2009). Using agile methods in software product development: A case study. *Sixth International Conference on Information Technology: New Generations, Las Vegas, NV*, 1415-1420. doi.ieeecomputersociety.org/

Strode, D., Huff, S., & Tretiakov, A. (2009). The impact of organizational culture on agile method use. *Proceedings of the Hawaii International Conference on System Sciences, Waikoloa, HI, 42,* 1 – 9. doi.ieeecomputersociety.org/10.1109/HICSS.2009.952

Surendra, N. (2008). Using an ethnographic process to conduct requirements analysis for agile systems development. *Information Technology & Management*, *9*(1), 55-69. doi:10.1007/s10799-007-0026-6. http://search.ebscohost.com.libproxy.uoregon.edu /login.aspx?direct=true&db=aph&AN=28715975&site=ehost-live&scope=site

Tate, K. (2006). *Sustainable software development: An agile perspective.* Upper Saddle River, NJ: Addison-Wesley.

Tolfo, C. & Wazlawick, R. S. (2008). The influence of organizational culture on the adoption of extreme programming. *The Journal of Systems and Software, 8*, 1955-1967.

Vinekar, V., & Huntley, C. (2010). Agility versus maturity: Is there really a trade-off? *Computer*, *43*(5), 87-89. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=51228468&site=ehost-live&scope=site

Vinekar, V., Slinkman, C., & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management*, *23*(3), 31-42.

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011). The agile requirements refinery: Applying Scrum principles to software product management. *Information & Software Technology*, *53*(1), 58-70. doi:10.1016/j.infsof.2010.08.004. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=cph&AN=54879223&login.asp&site=ehost-live&scope=site

West, D. (2009). Agile processes go lean. *InformationWeek*, (1228), 32-40. Retrieved from Academic Search Premier Database. http://search.ebscohost.com.libproxy.uoregon.edu/login.aspx?direct=true&db=aph&AN=40090026&site=ehost-live&scope=site

Wikipedia. (2010). Definitions of agile software development methodologies. Retrieved from http://www.wikipedia.org/

Woodward, E., Bowers, R., Thio, V., Johnson, K., Srihari, M., & Bracht, C. (2010). Agile methods for software practice transformation. *IBM Journal of Research & Development*, *54*(2), 3:1-3:12. doi:10.1147/JRD.2009.2038749. http://search.ebscohost.com.libproxy.

uoregon.edu/login.aspx?direct=true&db=aph&AN= 51695358&site=ehost-live&scope=site