SAIS 2006 Proceedings

Southern (SAIS)

3-1-2006

Towards a Theoretical Framework of Information Systems Development Strategy: The Contingent Effects of Organizational Culture and Project Uncertainty

Vishnu Vinekar
vvinekar@uta.edu

TOWARDS A THEORETICAL FRAMEWORK OF INFORMATION SYSTEMS DEVELOPMENT STRATEGY: THE CONTINGENT EFFECTS OF ORGANIZATIONAL CULTURE AND PROJECT UNCERTAINTY

Vishnu Vinekar
University of Texas at Arlington
vvinekar@uta.edu

Abstract

Although practitioners are facing increasing complexities and high rates of failure in IS development, research in the field has not yet developed adequate theory to address the issues involved. This study first analyzes the benefits and shortcomings of existing strategies of IS development, agile and traditional development, and then theorizes four underlying dimensions along which these strategies differ: change responsiveness, knowledge tacitness, people-driven, and customer collaboration. The strategy contingency perspective is utilized to theorize the effects of organizational culture and project uncertainty of IS development strategy. Using this perspective, a framework is developed that identifies two new strategies of IS development, in addition to the existing traditional and agile strategies. These two strategies of IS development are labeled the responsive process strategy and the creative optimization strategy. This may help explain practitioner surveys that report that most organizations adopt agile practices for some projects and not others, and that

Keywords: Information systems development, agile, contingency theory

Introduction

Information systems development has faced very low rates of success by several different measures (Lindstrom & Jeffries, 2004; The Standish Group, 2003). This has caused several IS development organizations to abandon traditional development methodologies in favor of 'lightweight' or agile methodologies. Traditional methodologies for information systems development follow an engineering approach, with sequential phases of planning, analysis, design, implementation and maintenance. This methodology involves detailed requirements gathering, followed by a design which includes all specifications. Development then proceeds according to the plan. Agile methodologists assume that it is impossible, or at least, very unlikely that requirements can be captured upfront, and that business requirements change during the course of information systems development. Several organizations developed varied methodologies to address the issue of responding to changing requirements over the course of IS development. Although several of these methodologies were developed separately, the pioneers of the then-labeled 'lightweight' methodologies formed the Agile Alliance, agreeing over several shared values and principles (Cockburn & Highsmith, 2001). Under the common umbrella of 'agile' methodologies, these pioneers put forth an 'agile manifesto'. In this manifesto, they averred that traditional development has been hindered by an over-reliance on processes, tools and following a plan, tended to focus more on documentation and negotiation of contracts. Agile methodologists professes that they focus instead on the creative abilities of their developers, and collaborations with their customers to develop working code and respond to change over the course of development. Essentially, agile methodologies do not attempt Big Design Up Front (BDUF), and instead, collaborate with the customers to prioritize the functionality of the information system as per business and technical requirements, and initially focus on a relatively small set of features with the highest business priorities. This is developed in a rapid iteration, ranging from between one and six weeks, resulting in a deliverable. The deliverable is released to the customer in order to elicit user feedback, required changes, and emerging and added functionality, which is then prioritized for the next iteration. In this manner, the information system is developed in an evolutionary manner rather than by following a fully specified design. The traditionalists believe that agile methodologies are too undisciplined and meant primarily for 'hackers'.

Background

Both traditional and agile methodologies have several benefits as well as concerns. Requirements may not be uncertain for every IS development project, and it may be possible to obtain a reasonably unchanging set of requirements at the commencement of a project. With these projects, traditional methodologies have the advantage of a design plan, which serves as a guide to the entire development effort. They also enable the exploitation of existing knowledge of IS development, and a refinement of pre-existing processes, techniques and tools. Following a traditional methodology may result in a well-documented system. Traditional methodologies also stimulate the cultivation of 'best practices' and are focused on creating an optimum solution for the given set of requirements.

Agile methodologies, on the other hand, have the ability to respond to change over the course of development. It encourages active customer participation, allows the developers greater creativity and flexibility, and results in information systems that are leaner and without software pollution, i.e. without unnecessary functionality. Agile methodologies have been shown to improve developer productivity, improve quality, and increase business satisfaction (Shine, 2003). Using incremental development with releases of minimum marketable features may be able to improve a projects return on investment, net present value, and decrease the investment on the project (Denne & Cleland-Huang, 2004). However, an IS development organization that is attempting to migrate towards agile methodologies may face several challenges. Such migration represents complex organizational change that cannot be accomplished merely by replacing existing tools with new ones. Technological change can affect an organization at 5 levels (Adler & Shenhar, 1990; Adler, 1989) skills, procedures, structure, strategy and culture. The main challenges of migrating from traditional to agile development, as outlined by Nerur, Mahapatra, and Mangalaraj (2005) affect all five levels.

In addition, organizations that manage to successfully migrate from traditional development to agile development as well as newer organizations that embrace agility from their inception face other challenges that arise due to agility. Agile development involves tacit, implicit knowledge where developers adapt to each project individually. As a result, codified, explicit knowledge is severely lacking due to this form of development. A consequence of this is that repeatable, optimized processes may be neglected, and developers would be necessitated to 'reinvent the wheel' with each project. An organization may fail to take advantage of the benefits that identifying, refining, standardizing and adhering to 'best practices' processes has to offer. In addition, the focus on change may make an agile development team become over-responsive and create unnecessary change (Boehm, 2002). Furthermore, agile development may result in a less than optimum information system for a project where requirements are fairly stable and the system is life-critical (Ambler, 2003; Boehm, 2002).

Despite the challenges faced in migration to agile development and the shortcomings of agility, it is becoming increasingly unfeasible for an IS development organization to follow traditional development methodologies. An average software project is associated with a twenty five percent change in requirements, which may be larger change than a purely traditional methodology can manage without substantial time and cost overruns (Boehm, 2002). Consistent with this, large nationwide surveys report that ‘incomplete and changing requirements’ is among the leading critical factors associated with unsuccessful projects, while user involvement, agility, smaller teams and smaller project durations are associated with successful projects (The Standish Group, 1994, 1995, 1999, 2001, 2003). Organizations that have migrated to agile indicate that the change has increased productivity of the developers, quality of the information system, and increased business satisfaction (Shine, 2003). Traditional development practices are also associated with high amounts of software pollution, with eighty percent of the provided features being seldom, rarely or never used (Larman, 2004). In addition, a traditional development project loses the opportunity to gather early revenues from releases of minimum marketable features that incremental development allows (Denne & Cleland-Huang, 2004).

Contingency Perspective of Organizational Strategy

The majority of information systems development organizations may not follow a purely traditional or purely agile methodology, in part due to the concerns with the shortcomings of each methodology outlined earlier. A survey by Shine Technologies shows that 96% of organizations that are knowledgeable of agile methodologies intend to adopt or continue using agile methodologies (Shine, 2003). However, the majority of these respondents do not believe that it is suitable for all projects. This leads to the first research question: what determines the suitability of an agile development strategy to a project or an organization?

Some practitioners also believe in combining agile and traditional methodologies (Boehm, 2002). Along these lines, the Methods and Tools newsletter found that of the organizations that are aware of agile methodologies, there are more organizations that are 'partial implementers' (i.e., they have adopted some agile practices and not others) than organizations that have either decided not to adopt, fully adopt, or adopt on a few projects. This leads to our second research question: what are these partial implementation? In other words, which practices would organizations adopt? Do all partial implementers have similar practices or do they vary? If the do vary, what may predict which practices an organization may adopt?

In addition, both practitioners and academics agree that agile development requires a suitable organizational culture (Lindvall et al., 2002; Nerur et al., 2005), and that changing an organization's culture takes several years (Adler & Shenhar, 1990). This raises our third research question: how are some organizations able to move between agile development for some projects and traditional development for others if a cultural change requiring years is necessary to enable this? Is it possible that partial implementation strategies enable this phenomenon?

While these research questions are important to our field, there is a noted lack of empirical studies. One of the major reasons for this lack of empirical research is the lack of suitable theoretical foundations to base empirical research on. This paper therefore makes an important contribution to the field of information systems by developing a suitable theoretical framework on which future empirical studies may be developed, the contingency approach to strategy. Contingency theory assumes that there is no single best strategy for all organizations, and that for each organization, a single strategy may not be equally effective under all circumstances. An approach to strategy from this perspective posits that organizational conditions and environmental conditions guide an organizations choice of strategy (Ginsberg & Venkatraman, 1985).

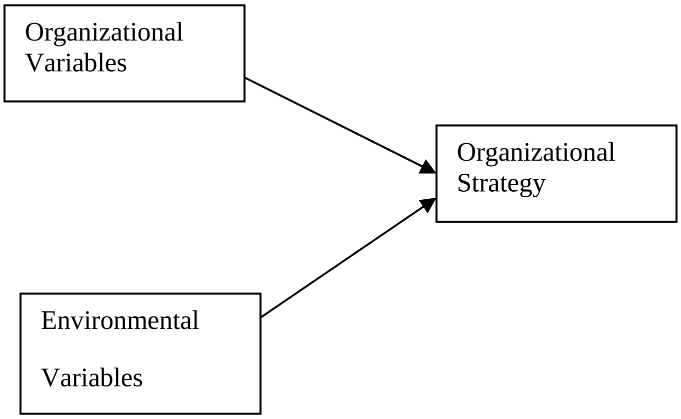

Figure 1. The Contingency Theory of Organizational Strategy (Adapted from Ginsberg & Venkatraman, 1985)

Contingency Perspective of IS Development Strategy

In applying this framework to information systems development, we need to first define the dimensions along which the dependent variable may vary. Traditional methodologies and agile methodologies may vary primarily along four dimensions, according to a definition of agile methodologies developed by a large group of practitioners of the methodologies (Lindvall et al., 2002). The *linear/iterative* dimension: traditional methodologies progress linearly, from planning, to analysis, design, implementation and maintenance; agile methodologies progress cyclically, in short iterations each of which involve all the phases of traditional development. The *constant/incremental* dimension: traditional development begins by requirements gathering with subsequent stages addressing all the functionality gathered in the initial phase; agile development starts with a prioritized set of requirements and proceeds by adding functionality, and is hence incremental. The *predicted/emergent* dimension: In traditional development, all functionality is predicted at the beginning; in agile development, requirements are allowed to 'emerge' from active user collaboration, participation, and feedback. The *process/people* dimension: Traditional development is process-driven, it follows fixed plans and procedure; agile development is people-driven, consisting of self-organizing teams that have a large amount of flexibility and loose processes.

In comparing these four dimensions to the values of agility as professed in the agile manifesto, we can see some amount of overlap. The agile manifesto states that agile methodologies value "individuals and interactions over processes and tools, working software over comprehensive documentation; customer collaboration over contract negotiation; responding to change over following a plan" (Boehm, 2002). It appears that the process/people dimension resembles the first value, 'individuals and interactions over processes and tools'. It also appears that the remaining four dimensions are closely related to the last value, 'responding to change over following a plan'. Therefore, we posit that linear/iterative, constant/incremental and predicted/emergent are three sub-dimensions of a single factor that we term 'change responsiveness'. These three sub-dimensions are what allow agile methodologies to respond to change over the course of development, and impedes traditional development from doing so. From the remaining two values, two more factors appear to emerge. 'Customer collaboration over contract negotiation' seems to indicate a *user segregation/collaboration* factor: traditional development takes user input only at the commencement of the development, while agile development needs active user collaboration throughout the project. Lastly, 'working software over comprehensive documentation' indicates a more subtle factor. Although the goal of both methodologies is to produce working software, the lack of documentation indicates a lack of codified knowledge. Hence we posit that the final factor is *explicit/tacit knowledge*: traditional development relies on a codified knowledge and documentation, while agile development relies on tacit, implicit knowledge.

Propositions

Now that we have defined the dimensions, we may proceed to theorizing the effects of organizational and environmental variables. A recurring theme in literature is the effect of organization culture. As discussed earlier, the shift from traditional to agile development involves significant changes in the skills, procedures, structure, strategy and culture. While changes in skills and procedures are small and can be accomplished in weeks, changes in structure and strategy are large, requiring several months to implement (Adler & Shenhar, 1990). However, it is the change in organizational culture that is the most significant, requiring years to adjust to the change. Practitioners as well as academicians aver that agility needs a suitable organizational culture, without which an agile approach is exceedingly improbable (Lindvall et al., 2002; Nerur et al., 2005). The major difference between the two approaches is that traditional approaches thrive in an ‘individualistic’ culture, where organization is hierarchical and control is authoritarian; while agile development requires a ‘collectivist’ culture, which involves a decentralized structure and a flexible environment to support adaptive teamwork (For a more detailed explanation of these organizational cultures, see O’Reilly, Chatman, & Caldwell, 1991). However, by analyzing the effects of these on the dimensions of IS development strategy, we posit that only two of these may be affected. Organizational culture should have a strong effect on the process-driven/people-driven factor, as processes need an optimization culture and self-organizing teams need a collectivist culture. It should also have an effect on the explicit/tacit knowledge factor, as individualistic cultures require adherence to measured repeatable processes while collectivist cultures require teams to adapt based on implicit knowledge. However, the effect of organizational culture on change responsiveness and user segregation/collaboration may not be significant. Hence we arrive at the first two propositions.

P1: IS development organizations that have a more individualistic culture will follow more process driven strategies of IS development than those that have a more collectivist culture.

P2: IS development organizations that have a more individualistic culture will follow strategies of IS development that rely more on explicit knowledge than those that have a more collectivist culture.

Of the environmental variables, the fundamental variable mentioned in the literature is that of project uncertainty (Larman, 2004; Lindstrom & Jeffries, 2004; The Standish Group 1994, 1995, 1999, 2001, 2003). Changing, incomplete, and incorrect requirements have forced practitioners to adopt strategies that are more responsive to change over the course of IS development, and that are more responsive to user needs by encouraging active user collaboration. Therefore, we posit that project uncertainty affects the two factors of IS development strategy, change responsiveness and user segregation/collaboration. Hence we arrive at propositions 3 and 4.

P3: IS development organizations will follow more change responsive strategies of IS development for projects that involve high levels of uncertainty than those that involve low levels of certainty.

P4: IS development organizations will follow more user collaborative strategies of IS development for projects that involve high levels of uncertainty than those that involve low levels of certainty.

Therefore we arrive at a contingency perspective of IS development strategy. There are four generic strategies that an IS development organization may follow, depending on the organization culture and project uncertainty, as outlined in figure 2.

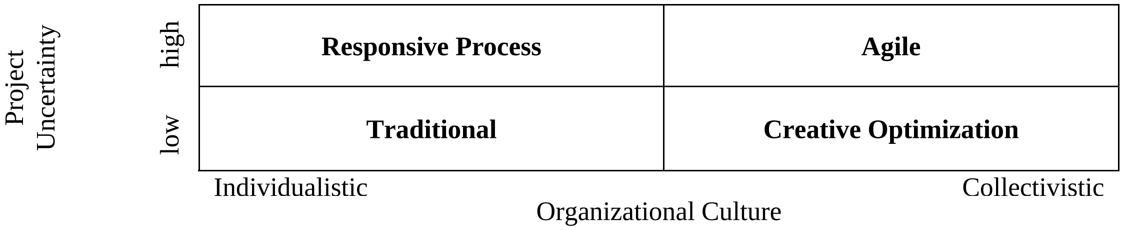

Figure 2. The relationship between project uncertainty and organizational culture

Two new strategies for IS development emerge in the matrix. The *responsive process* strategy is adopted by organizations with an individualistic culture which face high project uncertainty. These organizations face considerable challenges to use an agile strategy as this would entail a significant change in the organizational culture, requiring years to accomplish. However, they may be able to be more responsive to the uncertainty of the project through changes in skills and procedure, which are comparatively smaller organizational changes. This strategy involves high change responsiveness, indicating that the development will follow an iterative, incremental and emergent strategy, high user collaboration, process-driven and relying on explicit knowledge.

The *creative optimization* strategy is adopted by organizations with a collectivistic culture while facing low levels of project uncertainty. Again, these organizations may be unable to achieve the cultural shift required to take advantage of the benefits a traditional approach may have to offer in the context of relatively stable project functionality. Therefore, they seek to optimize through smaller changes at the skills and procedure levels. This strategy will have low change responsiveness, i.e., it will be more linear, constant, and predicted, low user collaboration, but will be people-driven and relying on tacit knowledge. The organizational strategic choice between the four generic strategies may be predicted by the contingency model of IS development as shown in figure 3. Using the framework of four generic strategies that an IS development organization may utilize, we arrive at our fifth proposition:

P5: The strategy that an organization chooses for IS development is dependant on the organization's culture. This relationship between organizational culture and IS development strategy is moderated by project uncertainty

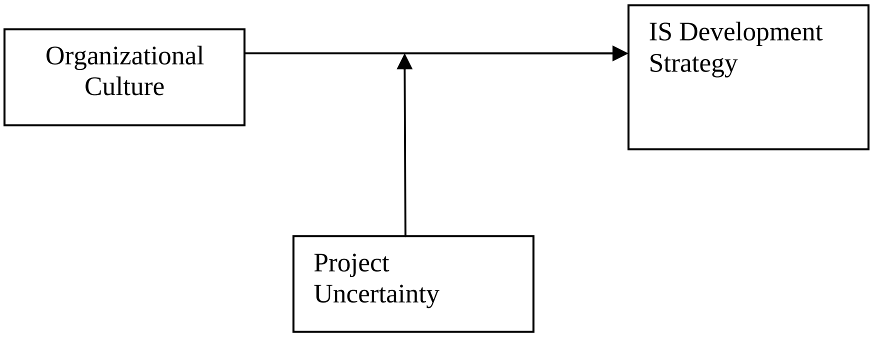

Figure 3. The Contingency Theory of IS Development Strategy.

Conclusions

This paper makes an initial effort at filling the gap between theory and practice in the area of IS development strategies. By establishing the dimensions of IS development strategies, we may provide researchers a valuable tool for subsequent empirical analysis of other organizational and environmental variables on IS development strategy, as well as effects of strategy on development success. It will also provide project mangers a useful guide for assessing their IS development strategy by measuring their levels on the demonstrated dimensions. Furthermore, by assessing the contingent effects of organizational culture and project uncertainty, we provide academicians an initial understanding of the complexity involved IS development strategy firmly grounded in strategic theory. The four generic IS development strategies also provide managers a useful guide for making choices based on their organizational culture and project uncertainty. This study therefore makes important contributions to both theory and practice of IS development.

References

Adler, P. S. (1989). CAD/CAM: Managerial challenges and research issues. *IEEE Transactions on Engineering Management*, 36(3), 202-215.

Adler, P. S., & Shenhar, A. (1990). Adapting your technological base: The organizational challenge. *Sloan Management Review*, 32(1), 25-37.

Ambler, S. W. (2003). The right tool for the job. *Software Development*, 11(12), 50-52.

Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64-69.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, 34(11), 131-133

Denne, M., & Cleland-Huang, J. (2004). The incremental funding method: Data-driven software development. *IEEE Software*, 21(3), 39-47.

Ginsberg, A, & Venkatraman, N. (1985). Contingency perspectives of organizational strategy: A critical review of the empirical research. *Academy of Management Review*, 10(3), 421-434.

Larman, C. (2004). *Agile & Iterative Development. A Manager's Guide.* Addison-Wesley Pearson Education, Massachusetts.

Lindstrom, L. & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information Systems Management*, 21(3), 41-52.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., et al. (2002). Empirical findings in agile methods. Presented at the Extreme Programming and Agile Methods-XP/Agile Universe, Chicago, IL, USA

Methods & Tools Newsletter, Retrieved February 13th 2006 from http://www.methodsandtools.com/dynpoll/archive.php

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 73-78.

O'Reilly III, C. A., Chatman, J., & Caldwell, D. F. People and organizational culture: Assessing person-organizational fit. *Academy of Management Journal*, 34(3), 487-516.

Shine Technologies (2003). *Agile methodologies survey results*. Retrieved January 2nd, 2006, from http://www.agilealliance.com/articles/shinetechnologiesagil/file

The Standish Group (1994). *The CHAOS report*. Retrieved January 2nd, 2006, from http://www.standishgroup.com/sample_research/chaos_1994_1.php

The Standish Group (1995). *Unfinished voyages*. Retrieved January 2nd, 2006, from http://www.standishgroup.com/sample_research/unfinished_voyages_1.php.

The Standish Group (1999). *CHAOS: A recipe for success*. Retrieved January 2nd, 2006, from http://www.standishgroup.com/sample_research/PDFpages/chaos1999.pdf.

The Standish Group (2001). *Extreme CHAOS*. Retrieved January 2nd, 2006, from http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf

The Standish Group (2003). *CHAOS chronicles version 3.0*. The Standish Group International Inc, Massachusetts.