View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by DSpace@MIT

Second International Symposium on Engineering Systems
MIT, Cambridge, Massachusetts, June 15-17, 2009

Exploring the Sources of Enterprise Agility in Software Organizations

Jayakanth Srinivasan[1]
Lean Advancement Initiative, Massachusetts Institute of Technology
School of Innovation Design and Engineering, Mälardalen University

Kristina Lundqvist[2]
School of Innovation Design and Engineering, Mälardalen University

Christer Norström[3]
School of Innovation Design and Engineering, Mälardalen University

Copyright © 2009 by [Jayakanth Srinivasan]. Published and used by MIT ESD and CESUN with permission.

Organizations involved in the design, development and sustainment of software systems have to manage the tension between creating new products and services, while at the same time maintaining their existing portfolio. This paper explores the sources of enterprise agility in software firms, wherein agility is defined as the ability of the organization to sense changes in its environment (both internal and external), and effectively respond to these changes. Using engaged scholarship as the overarching paradigm, we report on the findings of a process study that uses semi-structured interviews, observation, and archival firm & project information for data gathering, and grounded theory methods and comparative case analysis for data analysis and theory generation. The analysis highlights the importance of the four organizational enablers of: stakeholder alignment, employee empowerment, group & organizational learning, and governance mechanisms, as necessary but not sufficient precursors to obtaining enterprise agility. Furthermore, we provide illustrative case examples of the three mechanisms: Continuous Improvement, Creating Systems of Innovation, and Leveraging Globally available Capabilities, that software organizations use to gain enterprise agility.

Keywords: Software Organizations, Enterprise Agility, Engineering Systems, Engaged Scholarship, Mixed-Methods

1. Introduction

[1] Research Engineer, Lean Advancement Initiative, 77 Massachusetts Avenue, # 41-205, Cambridge, MA, 02139

[2] Professor of Dependable Software Engineering, School of Innovation Design and Engineering, P.O. Box 883, SE-721 23 Västerås, Sweden

[3] Vice President & Professor Computer Science and Computer Engineering, School of Innovation Design and Engineering, P.O. Box 883, SE-721 23 Västerås, Sweden

"*Software is a place where dreams are planted and nightmares harvested, where terrible demons compete with magical panaceas, a world of werewolves and silver bullets." -Cox* [1]

When Brooks pointed out that there was no silver bullet for solving the challenges of developing complex software systems [2] more than two decades ago, he noted that:

'The software product is embedded in a cultural matrix of applications, users, laws, and machine vehicles. These all change continually, and their changes inexorably force change upon the software product"- Brooks

Software systems have since become interconnected and ubiquitous, and the challenges of building these complex systems have only become more difficult. The recognition in the 1990's of needing to move from craft-based production of software to more industrial approaches [1], led to a shift in focus from the search of a technological silver bullet of the 1980's, to the emphasis on standardized processes in the 1990s. Yet, the 'software crisis' [3] still remains an open and more urgent problem.

When you frame the problem of software development as finding the unique blend of people, processes, and technology to solve some real-world problem, a potential answer emerges. Significant progress in the processes and technology strands of the puzzle have brought the people issues to the forefront. While people & organization issues have been at the forefront since the first NATO conference on software engineering [4], we still lack theories of software organizations that can help us better explain why some organizations are successful at developing software, and why others fail. This paper is a needed step towards to filling that void, and does so by developing a theory of enterprise agility in software organizations. The theory identifies the key mechanisms that software organizations use to sense changes in their environment, and reconfigure their resources to respond to those changes. In the ideal case, the organization would be able to adapt and influence those environmental changes.

Software organizations today operate in an environment that is characterized by volatility in customer preferences, rapid evolution of technology, increased workforce mobility, and greater fiscal responsibility on the part of capital providers. These organizations have to be agile enough to cope with an environment where *change is the only constant,* and yet remain disciplined enough to continue to do what made them successful. This notion that they have to possess both agility and discipline in how they create value serves as the starting point of the research presented in this dissertation. From a software engineering research standpoint, the use of the phrase *balancing agility and discipline* immediately brings to mind the work of [5], however, our intent is to go beyond the notion of managing the risks associated with selecting a software process model, to truly addressing firm-level value creation.

Within this paper, we present a simple classification of software organizations and justify our selection of a capabilities-based view as the foundation for our theory generation efforts. We discuss the three broad classes of mechanisms that software organizations use to build their capabilities, and identify the organizational enablers that support the use of these mechanisms.

2. Software Organizations

Unlike other organizations that can be categorized as typically belonging to a single industrial sector, software's pervasive nature makes that hard to do so. A simple exploration of NAICS (North America Industry Classification System) codes that explicitly mention software, and other sectors that we consider to be truly software-intensive (Table 1) highlights this challenge. For example, two industries that were traditionally thought to be manufacturing centric are now being driven largely by software innovations: the motor vehicle industry, and the aerospace industry. Broy et al. point out that while a current premium car has about a 100MB of binary code (270 user interaction functions distributed across 70 embedded platforms), the next generation vehicles (circa 2012) will contain about 1 GB of software [6]. More importantly, they highlight that even in the current state over 80% of the innovations come from computer systems, making it a major contributor to the value of contemporary cars, and project that 38% of the total value creation in automotive electrics/electronics by 2010 will be obtained through software (the worldwide value creation is expected to grow from 127 billion Euros in 2002 to an expected 315 billion Euros in 2015). Similarly, we have argued in earlier work [7] that software is the mechanism through which the true value of an aerospace platform is realized.

Table 1. Inadequacy of NAICS codes for Understanding Software Organizations

NAICS Code	**Explanation**
	Explicitly Includes Software
511210	Software Publishers
54151	Computer Systems Design and Related Services
5416	Management, Scientific, and other Technical Consulting Services
	Implicitly Includes Software (Illustrative examples)
3345	Navigational, Measuring, Electromedical, and Control Instruments Manufacturing
3361	Motor Vehicle Manufacturing

Despite this diversity at the industry sector level, software organizations can be understood with respect to the nature of their products, and the form of their value delivery. We extend Arora et al.’s classification [8] to include three subsectors:

- Shrink-Wrapped Software: This sub-sector focuses on software is often sold directly to the end consumer, either in the form factor suggested in the name (shrink wrapped boxes containing some physical media containing software), or in the form of an online download. There is a rich tradition of firm-level research in the area, for instance see [9]. More recently, there has been a call for greater research focus on product development within the sector [10].
- Software Services: This sub-sector, consisting of firms performing operations ranging from new software development, packaged software tailoring, and maintenance

operations makes up a significant portion of the software sector. While there has been significant research in the areas of outsourcing for application software development and packaged software tailoring, more work is needed in the area of software maintenance services.

- Software-Intensive Systems:This sub-sector consists of firms that span a spectrum of domains ranging from non-critical comfort functions in a vehicle to mission critical and safety critical functions such as nuclear reactor control and flight guidance. While there have been numerous studies that have focused on the challenges of building these systems in the production phase, little has been said about transitioning these systems from the concept to the production stage.

The two sectors that we specifically explore in our research are software-intensive systems development, and software services.

3. Models of Competition

To effectively discuss *firm-level value creation*, we have discuss the underlying models of market competition, and associated approaches that firms use to formulate competitive strategies. The common models of market competition include the Mason & Bain industrial organization model, the Chamberlinian monopolistic model, and the Schumpeterian creative-destruction model. While discussing each of the models in detail is outside the scope of this paper, we briefly highlight the key aspects of each of the models, and justify the capabilities-based model that we adopt as the foundation for understanding software organizations.

The Mason & Bain model [11,12] built on industrial organization economics, assumes that the rents a firm receives are a function of the industry structure that it is belongs to. The existence and value of barriers to entry (the number and relative size of firms, the existence and degree of product differentiation in the industry, and the overall elasticity of demand for the industry determine the firms overall performance. More simply, these three aspects form the Porterian levers of competition [13] within the structure-conduct-performance paradigm. Given the diverse nature of organizations involved in software development, and the relatively low barriers to market entry, the model does not provide sufficient insight into how software organizations create unique competitive advantage.

The Chamberlinian model [14] is built on the notion of firms having unique assets and capabilities, which then determine the strategies that a firm pursues and the associated performance outcomes it obtains. These unique assets include technical know-how, reputation, brand-awareness, and the willingness of managers to work together. One of the limitations of the model is the strict monopoly assumption, which is difficult to find in the case of the software industry.

The Schumpeterian model of competition is characterized by the assumption that revolutionary innovations in products, markets or technologies can only be imperfectly estimated by firms in the market. This model is representative of the environment which software organizations operate in, and the underlying idea of viewing the firm as a bundle of resources that provide strategic advantage [15] provides a foundation for studying these firms. An extension of this approach, the dynamic capabilities approach that was first articulated in [16], wherein dynamic capabilities are defined as:

“The firm's ability to integrate, build, and reconfigure internal and external competencies to address rapidly changing environments. Dynamic capabilities thus reflect an organizations ability to achieve new and innovative forms of competitive advantage given path dependencies and market positions.”

The construct was further refined in [17], who defined dynamic capabilities as:

“A *set of specific strategic and organizational processes that create value within dynamic markets by manipulating resources into new value creation strategies.”*

This has spawned extensive work in the areas of learning [18,19], in attempting to find micro foundations for dynamic capabilities [20] and in trying to understand strategic change [21].

Firms in software industry have commonality in that they operate in fast clockspeed environments [22]; with project teams as the primary means of creating value[23]; using a process which forms one the purest forms of knowledge work [24,25]. These characteristics coupled with the Schumpeterian competitive environment that these organizations operate it, make the ideas of dynamic capabilities the most applicable. However, there has been little research in using this construct, with the exception of the work in the Indian Software Services industry in [26,27]. In related work, the notion of enterprise agility has been explored in the software organizational context in [28] who define agility from the perspective of information technology, and [29], who brings together the notion of project agility and new product development agility in software organizations. We bring the two ideas together when we define enterprise agility as:

“The ability of the organization to sense changes in its environment (both internal and external), and reconfigure its resources and capabilities to meet those needs.”

This definition treats enterprise agility being an outcome of a firm developing and reconfiguring its capabilities (i.e. it's dynamic capabilities). At it’s core, this paper explores the mechanisms that software organizations use develop capabilities, and through the use of these mechanisms obtain enterprise agility.

4. Framing the Research

4.A. Mechanisms for Capability Development

Our definition of enterprise agility is predicated on the ability of an organization to evolve/adapt its capabilities. From the perspective of software organizations, there are three broad, overlapping approaches using which the organization can build capabilities:

- Software Process Improvement: When the organization has a clear sense of the set of capabilities that it will need in the long run, and has the time needed to developed them, then software process improvement is the easiest choice. However, there is no guarantee that the organization will actually achieve those capabilities, and even if it does, if those are the right capabilities in the long run. For example, Y2K organizations building up capabilities through training of personnel in other technologies such as Java and .Net to smoothly transition into related markets.

- Creating Systems of Innovation: When the organization has a sense of its potential future capabilities, it can develop them using systems of innovation, typically a mix of internal R&D, open innovation, and spin-offs from process improvement and outsourcing efforts. Again, this approach requires investment of time on the part of the organization.
- Leveraging Globally Available Capabilities: In the absence of time or the requisite talent base, the organization has to tap into the global talent base, either in the form of an outsourcing contract and/or an offshoring effort,and/or through global recruiting. This requires the organization to have a clear sense of what capabilities they are going to continue to retain as part of the core, and what portions they are potentially willing to commoditize.

4.B. Research Approach

Potts [30] points out the focus of research in software engineering has been driven by the 'research-then-transfer' model as opposed to the 'industry-as-laboratory' approach. The recent ICSE workshops [31,32] have highlighted that the need still exists for innovative research approaches to study broader scale socio-technical phenomena. The approach that we have developed and tested over the last four years to drive and support our own research is shown in Figure 1. It is built on using engaged scholarship [33] as the guiding philosophy, as the critical realist perspective [34,35] that it is built on, is one that deeply resonates with our own philosophy for research.

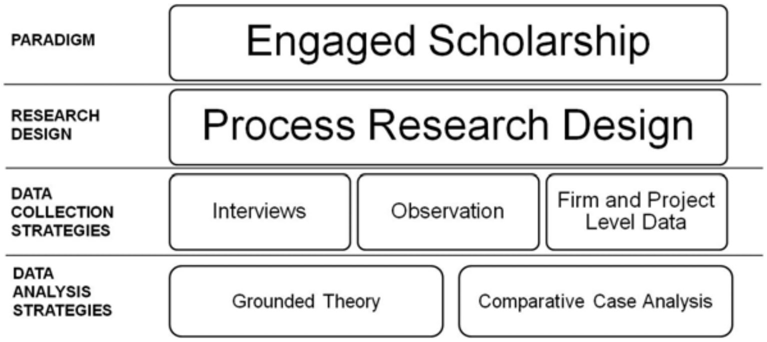

Figure 1. Research Approach

We use the word process along the same lines of [36], wherein a process can be understood as:

1. the logic used to explain a causal relationship in a variance theory
2. a category of concepts that refers to the actions of individual and organizations
3. a sequence of events that describes how some entity thing changes over time

There are two broad classes of research designs that can be used in studying the strategy process in organizations, a variance research design (aimed at explicating relationships between independent and dependent variables, and pertains to the 1st definition of process), and a process research design (aimed at understanding the evolution of a process, pertaining to the latter two definitions of process). The process research design that we adopt is built around developing that category of concepts, and in understanding how specific firms evolved in their quest for enterprise agility. Since the processes we are interested in studying at the firm level have not always been well understood or completely articulated, we adopted a multiple case sampling strategy [37] (wherein he defines a case study as an empirical inquiry within its real-life context, particularly when the boundaries between phenomena and context are not clearly evident), across two dimensions: the development process; and the primary value creation approach; as shown in Figure 2.

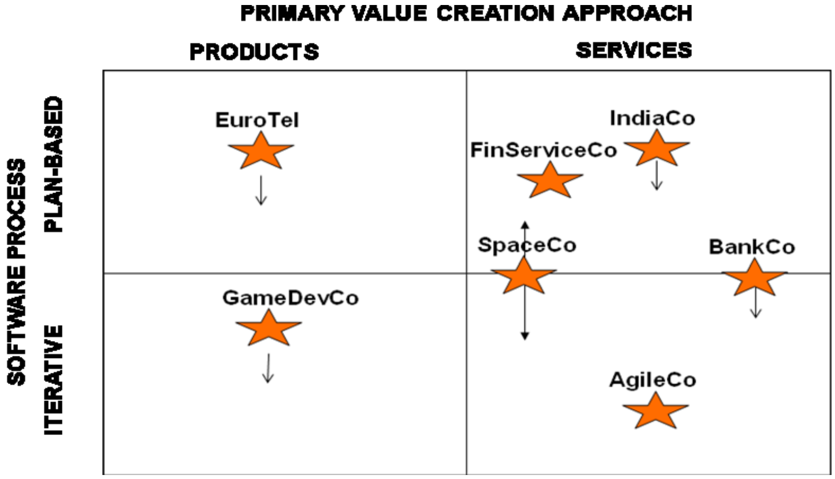

Figure 2. Understanding the Sample Space

Our first round of pilot case studies involved three organizations FinServicesCo, BankCo, and SpaceCo. The FinServicesCo and BankCo studies provided the insights needed to further explore agile process improvement using the in depth case studies of AgileCo and GameDevCo [38]. The insights from SpaceCo led to the creation of a theoretical framework for understanding aspects of innovation in software organizations, and ensure a first pass validation. The two in depth studies of EuroTel [39] and IndiaCo [40,41] were carried out to provide deeper insights into how organizations leverage globally available capabilities.

For data gathering, we use a mix of semi-structured interviews [42], observations [43] captured in field notes [44], and archival data. The intent behind using semi-structured interviews, was to go gather richer data from the participants by providing them with an opportunity to reflect more broadly from their individual perspective, and from their organization's context. This structure also provided us with the flexibility of exploring specific areas in greater depth, especially in the second and third rounds of fieldwork. While most interviewees consented to be taped during the interviews, in the case of the few that did not consent (either due to organizational policies or personal preferences) and during closed door meetings with the senior leadership teams, we captured field notes that later became invaluable sources of insights. We used observations to further enhance our understanding of the organization culture and social dynamics, i.e. as validation sources, rather than as the primary sources of insights. The two predominant approaches that we use to conduct the data analysis were grounded theory [45] in the vein of [46], and comparative case analysis [47]. We chose a grounded theoretic approach to data analysis as it enables us to capture the context within which the phenomena occurred, and create thicker descriptions of how each of the organizations evolved to their current state. When we consider the sample of the four in depth case studies, they are essentially polar cases along the software process dimension (iterative versus plan based), and the principle value creation dimension (products versus services), hence leading to the use of comparative case analysis in addition to grounded theory.

5. Executing the Research

5.A. AgileCo

AgileCo was started in 2001 as the Indian arm of a global software services provider. In 2005, AgileCo consisted of about 75 people and was expecting to double in size by 2006. We conducted 12 interviews at AgileCo in two sessions spanning a week in total. Our interviewees included three senior managers , four business analysts, and five developers. Our observations of members of AgileCo occurred when we were teaching with them at two workshops, as well as during breaks between interviews, and after-work social interactions. Given that their parent company had a long history of applying agile development, AgileCo also adopted agile methods, but faced significant challenges in terms of educating their personnel in the use of agile methods. Over the last seven years AgileCo has become one of the benchmarks of agile adoption and usage in India. In addition to mentoring other organizations in agile methods, their staff is encouraged to share their understanding of agile methods through conference papers, teaching tutorials, and participation in local knowledge networks. The four key aspects of AgileCo: personnel selection and training, building strong teams, managing customer expectations, and teaching/mentoring, are reflective of the learning from the pilot case studies, and additionally emerged from the analysis of the interview data as being key to the success of AgileCo.

5.B. GameDevCo

The study at GameDevCo was initiated in late 2005, athough it took almost two years to gain access to study the organization. We were aware that GameDevCo was in the process of adopting agile methods (since our primary project champion had just joined the organization), and we were hoping to understand how they had adopted agile methods, and how they planned to sustain it. The data was gathered using semi-structured interviews, after-hour conversations, analysis of project data, and observations on team meetings. The data gathering and analysis covered four steps: The first step involved conducting 22 semi-structured interviews, which included three of the four business unit heads, four scrum masters, the lead technical architect of their most successful product, three members from the verification and validation team, and three project teams, three sustainment personnel, and two operations team members; the second step in Aug 2008, was an interim review meeting with the project champion on ensure that the analysis was essentially correct, and to determine where the gaps existed that needed to be refined; The third step in Dec 2008, focused on the governance process between GameDevCo, and their corporate office; and finally in January 2009, we conducted a review with the top leadership team. This cycle of data gathering, inductive analysis, and stakeholder review, ensures that the findings in this chapter are trustworthy. The key themes that emerged from the analysis of GameDevCo focused on requirements management, requirements management & knowledge capture, product variation, verification & validation, and human resource management.

GameDevCo was born as a startup project that was designed and built on a university campus by developers who had a deep passion for the game of poker, and the technological expertise in the three critical areas of: Server-side software, gaming engine, and client-side software. The game was initially designed to develop the game playing skills using ‘play’ money, but the success of the product overall led to the creation of a product that would allow people to buy-into and play the actual games. This success led to the formation of a company to market the product on a larger scale.

5.C. EuroTel

TelCorp is a global leader in the design, development and sustainment of telecommunications equipment. From an organizational structure standpoint, TelCorp is best understood as having three business units: the Applications Business Unit (ABU), the Advanced Research Group (ARG), and EuroTel. The Applications Business Unit (ABU) acts as the primary face of TelCorp to its customers, and is responsible for short-term product innovation, product sales and client management for TelCorp. The Advanced Research Group (ARG) carries out both medium-term and long-term research in defining the next generation of product capabilities. EuroTel serves as the internal technological platform provider for TelCorp. Our first round of field work at EuroTel involved interviews with eight members of the senior leadership team. The selection of personnel to interview was driven by a desire to discuss technical and managerial challenges that were currently being faced, and how those challenges could/would be mitigated in the long run. These interviews provided the historical background necessary for understanding the evolution of the organization, and provided some preliminary insights into the drivers underlying their evolving relationship with their outsourcing supplier - IndiaCo. We followed the first round of fieldwork by conducting a workshop with members from both IndiaCo and EuroTel (including the senior leadership teams of both organizations), in order to understand the dynamics of the relationship, and gain a deeper understanding of their strategic direction of the relationship going forward. The findings of the engagement phase and the workshop led to a second round of fieldwork at EuroTel which focused on project team level issues. In this round of focused fieldwork, we interviewed twelve team members from two teams(including their project managers), in order to understand the project level drivers of success. The analysis of EuroTel focused on the strategic issues (disconnect from the rest of the enterprise, risk-average culture, and dearth of innovation) and the execution challenges (resource constraints and schedule pressure).

5.D. IndiaCo

IndiaCo’s roots can be found in an early 2000’s pilot project in the maintenance arena to demonstrate their capabilities in supporting the needs of their European client, EuroTel. While they have historically been a services provider, the offshore development center they established for EuroTel, represents one of their most successful efforts in the product development and sustainment arena. Although their relationship with their client began as a cost saving effort, the last eight years has seen IndiaCo evolve towards becoming a strategic partner. While we see their evolution as being consistent with the typical engagement model adopted by most software services firms, IndiaCo is unique in their approach to transforming a services based culture to the product development and support environment. Although the contractual relationship between IndiaCo and their customer remains time-and-materials based, their senior leadership has relied on continuous improvement to drive down costs, innovate processes for knowledge transfer and dedicated internal (basically non-billable) resources to demonstrate the increasing value proposition to support their growth. In the first round of fieldwork, we conducted sixteen interviews in total with twelve project managers, two senior leaders, and two developers. The focus on project managers was driven partly because we felt that the project managers were the closest to the actual challenges of executing projects with EuroTel. In addition to these individual interviews, we used coffee breaks and lunches to gain a sense of the IndiaCo environment. Each of these interviews were transcribed, and analyzed to find common themes across the various levels of analysis. In addition to the interviews, we also were given access to the training materials used internal to IndiaCo and to some project data. Once the preliminary data analysis was complete, we did a second round of fieldwork in August, in which we focused on team interviews with members of all seven projects that formed the core portfolio at IndiaCo. These interviews gave us deeper insight into the dynamics of the relationship between IndiaCo and EuroTel, and made more explicit the three major classes of projects in the portfolio: Product Sustainment, Product Ownership, and Knowledge-Work. The third round of meetings at IndiaCo in December, focused on presenting the findings of the case study, and getting feedback from members of IndiaCo. Recently, we visited IndiaCo for the third and final round of fieldwork, working primarily with the senior leadership team to discuss challenges imposed by the recession. Among the themes that emerged were: a focus on long-term performance, capabilitiy development (including knowledge management, and competency development), managing human capital (including leadership), and governance.

6. Conclusions and Future Research

Through the analysis of the data, and theoretical triangulation, we identified the four key organizational enablers of stakeholder alignment, employee empowerment, group & organizational learning, and governance systems. We decomposed each of the enablers into their constituent elements, and used those elements for better understanding the case studies. Our analysis of the organizational enablers showed that in the cases of organizations that were successful, the enablers were present to a greater degree than in organizations that were not so succesful. For example, success in achieving continuous improvement is predicated on alignment and engagement of key stakeholders (employees and senior leadership in particular), empowerment of employees to make the key changes needed, learning to ensure that best practices are easily disseminated (and mistakes are not repeated), and a governance mechanism to ensure that these efforts are in fact meeting the objectives. Similarly, innovation is carried out by people, and translated into organizational capabilities through a system that has to be governed effectively. In the case of leveraging globally available capabilities, stakeholder alignment is critical to prevent fear within the organization, and learning and governance are foundational to support long term sustainability.

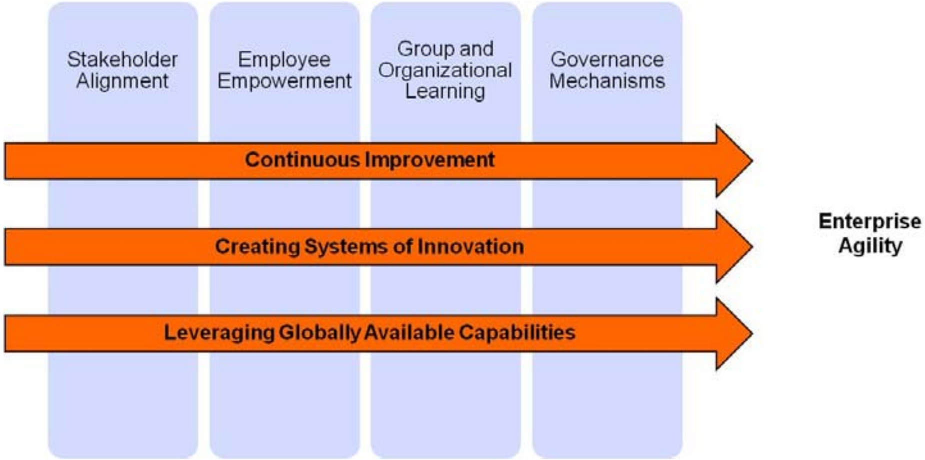
Stakeholder Alignment
Employee Empowerment
Group and Organizational Learning
Governance Mechanisms
Continuous Improvement
Creating Systems of Innovation
Leveraging Globally Available Capabilities
Enterprise Agility

Figure 3. Towards a Theory of Enterprise Agility in Software Organizations

Based on our work, we believe that enterprise agility is gained through the use of a combination of one or more of the mechanisms of continuous improvement, systems of innovation, and leveraging globally available capabilities. Furthermore, enterprise agility requires the presence of all four organizational enablers (in some degree). As with all research, it sets the stage for further exploration:

1. We have provided useful insights into the use of agile methods in the product development context, but more research is needed in the application of these methods to the development of mission critical and safety-critical systems
2. The framework for transitioning ideas to production was developed theoretically, and first pass evidence from the GameDevCo case shows that it has explanatory powers. More research is needed to improve its predictive capabilities.
3. The qualitative case study on customer-supplier relationships needs to be generalized across geographical and industrial contexts.
4. We have identified the four organizational enablers, and hypothesize that the relationship between them starts with stakeholder alignment as the foundation, and successively builds up through employee empowerment, governance, and learning. More research is needed to test this hypothesis.

7. Acknowledgement

The research presented here was supported in part by the Swedish Foundation for Strategic Research (SSF) through the PROGRESS center at Mälardalen University.

References

[1] B.J. Cox, “Planning the software industrial revolution,” *IEEE software,* vol. 7, 1990, pp. 25-33.

[2] F.P. Brooks, “No silver bullet: Essence and accidents of software engineering,” *IEEE computer,* vol. 20, 1987, pp. 10-19.

[3] W. Gibbs, “Wayt (1994)" Software's Chronic Crisis,” *Scientific American*, 1994, pp. 72-81.

[4] P. Naur, B. Randell, and F.L. Bauer, *Software Engineering: Report on a conference sponsored by the NATO SCIENCE COMMITTEE, Garmisch, Germany, 7th to 11th October 1968*, Scientific Affairs Division, NATO, 1969.

[5] B.W. Boehm and R. Turner, *Balancing agility and discipline: A guide for the perplexed*, Addison-Wesley Professional, 2003.

[6] M. Broy, I.H. Kruger, A. Pretschner, and C. Salzmann, “Engineering automotive software,” *Proceedings of the IEEE*, vol. 95, 2007, pp. 356-373.

[7] J. Srinivasan and K. Lundqvist, *Why is Aerospace Software Development and Sustainment Hard?*, Cambridge: Massachusetts Institute of Technology, 2005.

[8] A. Arora, V.S. Arunachalam, J. Asundi, and R. Fernandes, “The Indian software services industry,” *Research Policy*, vol. 30, Oct. 2001, pp. 1267-1287.

[9] M.A. Cusumano and R.W. Selby, *Microsoft secrets*, Free Press New York, 1995.

[10] L. Xu and S. Brinkkemper, “Concepts of product software: Paving the road for urgently needed research,” 2005.

[11] J.S. Bain, *Industrial organization*, John Wiley & Sons, 1968.

[12] E.S. Mason, “Price and production policies of large-scale enterprise,” *The American Economic Review*, 1939, pp. 61-74.

[13] M.E. Porter, “Competitive strategy,” *New York*, 1980.

[14] E.H. Chamberlin, “Monopolistic or Imperfect Competition?,” *The Quarterly Journal of Economics*, 1937, pp. 557-580.

[15] B. Wernerfelt, “A resource-based view of the firm,” *Strategic management journal*, 1984, pp. 171-180.

[16] D.J. Teece, G. Pisano, and A. Shuen, “Dynamic capabilities and strategic management,” *Strategic management journal*, 1997, pp. 509-533.

[17] K.M. Eisenhardt and J.A. Martin, “Dynamic capabilities: what are they?,” *Strategic management journal*, 2000, pp. 1105-1121.

[18] M. Zollo and S.G. Winter, “Deliberate learning and the evolution of dynamic capabilities,” *Organization Science*, 2002, pp. 339-351.

[19] S.G. Winter, “Understanding dynamic capabilities,” *Strategic Management Journal*, 2003, pp. 991-995.

[20] D.J. Teece, “Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance,” *Strategic Management Journal*, vol. 28, 2007.

[21] C.E. Helfat, *Dynamic capabilities: Understanding strategic change in organizations*, Wiley-Blackwell, 2007.

[22] C.H. Fine, *Clockspeed: Winning industry control in the age of temporary advantage*, Perseus Books, 1998.

[23] S. Faraj and L. Sproull, “Coordinating expertise in software development teams,” *Management Science*, 2000, pp. 1554-1568.

[24] R.W. Zmud, “An examination of'push-pull'theory applied to process innovation in knowledge work,” *Management Science*, 1984, pp. 727-738.

[25] T.H. Davenport and L. Prusak, “Working knowledge: How organizations manage what they know,” *Ubiquity*, vol. 1, 2000.

[26] S.S. Athreye, “The Indian software industry and its evolving service capability,” *Industrial and Corporate Change*, vol. 14, 2005, pp. 393-418.

[27] S.K. Ethiraj, P. Kale, M.S. Krishnan, J. Singh, S. Ethiraj, and D. Levinthal, "Where do capabilities come from and how do they matter? A study in the software services industry," *Strategic Management Journal*, vol. 26, 2005, pp. 25-45.

[28] E. Overby, A. Bharadwaj, and V. Sambamurthy, "Enterprise agility and the enabling role of information technology," *European Journal of Information Systems*, vol. 15, 2006, pp. 120-131.

[29] P. Kettunen, "Extending Software Project Agility with New Product Development Enterprise Agility," *Software Process: Improvement and Practice*, vol. 12, 2007.

[30] C. Potts, "Software-engineering research revisited," *IEEE Software*, vol. 10, 1993, pp. 19-28.

[31] S.E. Sim, J. Singer, and M.A. Storey, "Beg, borrow, or steal: Using multidisciplinary approaches in empirical software engineering research," *Empirical Software Engineering*, vol. 6, 2001, pp. 85-93.

[32] Y. Dittrich, O. Hazzan, F. Maurer, M. John, H. Sharp, J. Sillito, S. Sim, M.A. Storey, B. Tessem, and G. Venolia, "Cooperative and human aspects of software engineering (CHASE 2008)," 2008.

[33] A.H. Van de Ven, *Engaged scholarship: A guide for organizational and social research*, Oxford University Press, USA, 2007.

[34] R. Bhaskar, "A realist theory ofscience," *Brighton: Harvester*, 1978.

[35] R. Bhaskar, *The possibility of naturalism*, Harvester Wheatsheaf New York, 1979.

[36] A.H. Van de Ven, "Suggestions for studying strategy process: a research note," *Strategic Management Journal*, 1992, pp. 169-191.

[37] R.K. Yin, *Case study research: Design and methods*, Sage Publications, Inc, 2008.

[38] J. Srinivasan and K. Lundqvist, "Organizational Enablers for Agile Adoption: Learning from GameDevCo," *Agile Processes in Software Engineering and Extreme Programming: 10th International Conference, XP 2009, Sardinia, Italy, May 25-29, 2009, Proceedings*, Springer, 2009, p. 63.

[39] J. Srinivasan, "Studying Customer-Supplier Relationships in Global Software Development," 2008.

[40] J. Srinivasan, "Architecting a Lean Software Enterprise: The TODC Story," Hyderabad, India: Strategic Management Society, 2008.

[41] J. Srinivasan, A. lofgren, C. norstrom, and K. Lundqvist, "Lessons Learned from a Workshop on Relationship Building," Limerick, Ireland: IEEE Computer Society Press Los Alamitos, CA, USA, 2009.

[42] I. Seidman, *Interviewing as qualitative research: A guide for researchers in education and the social sciences*, Teachers College Press, 2006.

[43] J.P. Spradley and K. Baker, *Participant observation*, Holt, Rinehart and Winston New York, 1980.

[44] R. Sanjek, *Fieldnotes: The makings of anthropology*, Cornell University Press, 1990.

[45] B.G. Glaser and A.L. Strauss, *The discovery of grounded theory*, Aldine de Gruyter New York, 1967.

[46] K. Charmaz, *Constructing grounded theory: A practical guide through qualitative analysis*, Sage, 2006.

[47] C.C. Ragin and H.S. Becker, *What is a case?: exploring the foundations of social inquiry*, Cambridge Univ Pr, 1992.