Mälardalen University Press Dissertations
No. 77

EXPLORING THE SOURCES OF ENTERPRISE AGILITY IN SOFTWARE ORGANIZATIONS

Jayakanth Srinivasan

2009

School of Innovation, Design and Engineering

ISSN 1651-4238
ISBN 978-91-86135-38-6
Printed by Mälardalen University, Västerås, Sweden

Mälardalen University Press Dissertations
No. 77

EXPLORING THE SOURCES OF ENTERPRISE AGILITY IN SOFTWARE ORGANIZATIONS

Jayakanth Srinivasan

Akademisk avhandling

som för avläggande av teknologie doktorsexamen i datavetenskap vid Akademin för innovation, design och teknik kommer att offentligen försvaras fredagen 23 oktober, 2009, 13.15 i Kappa, Mälardalens högskola, Västerås.

Fakultetsopponent: Prof. Pekka Abrahamsson, University of Helsinki, Department of Computer Science

Akademin för innovation, design och teknik

Abstract

Software is one of the core elements that drive the modern economy, with visible use in areas such as personal computing, telecommunications and banking, and background use in areas such as aircraft traffic management, nuclear power generation, and automotive control systems. Organizations that build software are unique in that they span industrial domains, and at their core of what they do is codifying human knowledge. When we talk about software organizations, we think of organizations that work in the three broad areas of shrink wrapped application software, software-intensive systems, or software services. By shrink wrapped application software, we refer to the software that one can buy in a retail store for use on his or her computer. Software-intensive systems are part of a larger system such as air traffic management, and software services focus on making software work for other organizations. This thesis uses studies of eight software organizations to understand how these organizations are able to identify changes to their environment, and create the required capabilities to meet those changes – in other words, how these organizations gain enterprise agility.

To understand enterprise agility, we ask three simple questions, namely how does the organization improve what it currently does? What does the organization do? and Who does the work that the organization chooses to do? By answering each of these questions in the context of software organizations, we identify the three mechanisms of *Software Process Improvement (SPI), Creating Systems of Innovation (CSI), and Leveraging Globally Available Capabilities (LGAC)*. These three mechanisms are interconnected and interdependent. By creating rich descriptions of how these mechanisms are implemented in the organizations that we studied in the thesis, we are able to build confidence that these mechanisms are an accurate representation of the approaches that organizations use. In addition to identifying the mechanisms, by analyzing across the cases, we identify the four organizational enablers of *stakeholder alignment, employee empowerment, group & organizational learning, and governance*.

Organizations can create enterprise agility by ensuring the presence of the four organizational enablers and leveraging some combination of the three mechanisms. While it is possible for the organization to create enterprise agility in the absence of these mechanisms, we believe that the agility generated is not sustainable. To survive in the tough economic conditions of today, software organizations need to be aware of, and actively manage both the enablers and the mechanisms for sustained success. This thesis is a first step in finding more effective ways to manage software organizations as a whole, rather than as a collection of individual projects. It presents a philosophy of thinking about software organizations that addresses the uniqueness of these organizations while at the same time leveraging best practices and thought leadership from the disciplines of software engineering, quality, knowledge management, strategy, organizational theory, and stakeholder theory.

ISSN 1651-4238
ISBN 978-91-86135-38-6

To Amma and Appa

Acknowledgements

This thesis marks the culmination of an eleven year journey that began in 1998. Sitting at home with my parents, contemplating my future, there seemed two distinct paths that I could take: go to grad school, or enter the employment market. Mom has always had a gift for simplifying things, and in my case, she distilled the choices down to, 'go to school, and you can stay at home', or 'move out'! I ended up moving out to a grad school dorm for my first graduate degree in Avionics Engineering. Two years later, in 2000, we came to the same fork again – study more or enter the workforce. This time, it was an offer for research funding from MIT that got me to move to Cambridge, for a second graduate degree in Aeronautics and Astronautics, with the plan that in five years, I would graduate, and hopefully, get a respectable job. As with most projects, the final product was delayed, experienced cost overruns, … but …, now it is ready for release with an upgraded feature set. There are so many people who have had a visible or invisible impact in shaping both this thesis and my life. I am deeply grateful for your thoughtfulness, support, and kindness.

This journey has had its up's and down's, and I have been blessed to have had the faith, which has helped me sail through some stormy waters, to reach safe harbor. That faith, instilled in me by my parents, has taught me to be thankful, they gave me the humility to understand that every time you get knocked down, there is a purpose for it, and that you will be stronger for it. Thank You Murugha Swamy for helping me successfully draw to a close one important chapter in my life!

To my mentors and to the schools that trained me - Thank You. My mentors saw something in me that I myself could not see. The schools that I attended each helped me become a better scholar and a better human being. Mr. Krishnamurthy and Mr. Kannan, of Indian School Muscat, your patience in teaching me the fundamentals was one of the reasons this journey began. The faculty members of the Computer Engineering department at MIT-Manipal gave me the CS foundations to explore applied computer science.

MIT-Madras forced me to think independently. The faculty members of the Avionics program, each in their own way helped me expand my horizons on what was possible. Prof. K. Padmanaban, not only taught me the basics of aircraft design, but also set the bar for what it meant to be a great teacher. The first thing he said when he met me: 'tennis shoes are meant for sports' – was a lesson on being a good role model. To Mr. P.M. Soundarrajan, my first boss, who fanned my passion for aircraft, and let me contribute to the research in the laboratory, and through it, to my country. He showed me that when someone said it was impossible, they just wanted you to prove them wrong. To the entire team at HCL Technologies, in what was then the Networking Systems Laboratory, for showing me that what I knew about aircraft could be applied to networking software. My second boss, P.N. Janardanan, taught me that you don't have to be bossy to be the boss, you just have to be intelligent, inspirational, and when necessary, intimidating!

MIT is where I grew up as a scholar – MIT challenged me, inspired me, frustrated me, enraged me, and made me find my calling as a researcher and a teacher. Prof. Nancy Leveson funded my first three years at MIT, without her I would not be here. Prof. Lundqvist and I met as part of Nancy's lab, and it has been a hell of a ride for the last decade! Course 16, Aeronautics and Astronautics, a department that will forever be a part of me – I learned to become a better teacher thanks to 16.070 and Unified Engineering.

LAI helped me make the transition from student to researcher. To Dr. Kirkor Bozdogan who sat me down in 2004, told me get my act together, and then helped me do it! I have said it before, and I will say it again, Sir, if I become half the scholar that you are by the time I retire, I will consider my career a success. To Prof. John Carroll, thank you for the time you have spent with me patiently acting as a bouncing board for everything from research ideas to life dilemmas. Prof. Debbie Nightingale, who is a friend, colleague, mentor, and an all around amazing human being, thank you for allowing me to grow into my role, and for helping shape me into the person I am. I will never forget the day you told me to articulate what it was I cared about - I continue to strive to live those values even today. The other incident that I cherish is your reaction after a day-long brainstorming session, which concluded with 'it is in the interfaces, stupid!' That statement continues to drive my research in understanding organizations and their approaches to transformation. This thesis on enterprise agility has its roots in our discussions.

MDH, which completes my earned degree collecting phase, was where I matured as a researcher. To all the faculty members and researchers, thank you for taking time to spend talking to me about my research, and yours. Your ideas helped shape mine, and hopefully, those conversations were value added to you. To Dr. Radu Dobrin, a consummate theoretician, who showed me the ropes, and explained the Swedish educational system, thank you for everything! To the entire PROGRESS team, I will never forget the trip to Via Reggio where the legend of the one-eyed dog was born! To my thesis reviewers, and to the companies that I worked with as part of this thesis, Thank you!

To my advisors, Prof. Christer Norström and Prof. Kristina Lundqvist, you both have spent hours with me on this thesis, and I hope that the journey has been as much fun for you both, as it has been for me. Christer showed me that service to the school does not inhibit you from doing good research. Thank you for looking at the first draft of this thesis and sitting me down to storyboard it. Kristina gave me the intellectual freedom to do a qualitative dissertation, and backed me through the entire journey. We have worked together for almost a decade, and there are no words to adequately express how grateful I am!

This would not have been possible without the various funding agencies, including, the Department of Electronics (Government of India), NSF, Eurocontrol, Draper Laboratories, Ford Motor Company, MIT, LAI, USAF, US Army, SSF, ARTES, PROGRESS, Boeing. I promise I will come around looking for more funding soon.

I have been lucky that my academic career has provided me with an opportunity to teach and advise both undergraduate and graduate students. To the first 16.070 class that I helped

design and teach in 2003, thank you for making me a better teacher. To the graduate students that I helped supervise, Erisa Hines, Vikram Mahidhar, Allen Fong, Shweta Sharma, and Danny Gagne, thank you for letting me be a part of your academic journey. You have each enriched my work and made me a better advisor along the way.

To my friends, Fabio, Samantha, Deepti, Vijay, Barbara, Girish, Shankar, Tara, Vikram, Elwin, Ricardo, without you this journey would have been impossible. I am blessed that you are in my life!

Most importantly to my family, I know the toll that this thesis has taken on all of you, and I love you all. Thank you to Chechi and Anand, for always being there. Ney, for all the hours spent on reading thesis and drafts, for all the lost time when I was commuting across the Atlantic, and the list goes on and on. I have a lifetime to show you how thankful I am that you are in my life, know that you complete me!

Amma and Appa, this thesis is dedicated to you both. For the years of sweat, tears, and sacrifice, I hope that completing this thesis gives you as much joy and satisfaction as it gives me. It was a tiny idea that became a dream, and is now a reality. Even though you both have not always agreed with me, you have stood by me. For your blessings, your faith in me, and your undying support, I am grateful!

Jayakanth Srinivasan

Västerås/Cambridge
September, 2009

TABLE OF CONTENTS

x

List of Tables

List of Figures

1 ENTERPRISE AGILITY IN SOFTWARE ORGANIZATIONS

"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 (1990)

1.1 Introduction

Software is everywhere today! We use it in areas ranging from the control of complex systems such as cars, aircraft, nuclear power plants, and financial services, to managing simpler comfort functions such as air conditioning and calendaring. This explosive growth in the use of software to provide increased system capabilities has made understanding how organizations innovate, design, develop, and deploy these systems a critical area of research. The recognition in the 1990's of needing to move from craft-based production of software to more industrial approaches (Cox, 1990), 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, 'software's chronic crisis', a term coined by Gibbs (1994) in discussing the challenges of building large software systems, remains chronic. The factors that were mentioned in 1994 as being the root causes for projects failing continue to remain the root causes of project failure today. When Brooks (1987) pointed out that there was no silver bullet for solving the challenges of developing complex software systems more than two decades ago, he noted:

"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"

While the need for this socio-technical view of software development has been articulated, the mechanics of building software that addresses the human and organizational aspects are still being developed. Framing the problem of software development, as finding the unique blend of people, processes, and technology to solve some real-world problem, provides an alternative way of understanding software organizations. The people and organization issues have been at the forefront since the first NATO conference on software engineering (Naur & Randell, 1968), but 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 thesis is a needed step towards to filling that void. By identifying the mechanisms that software organizations use to gain competitive advantage and the organizational enablers that are necessary to make them successful, this thesis provides a deeper understanding of how software organizations gain enterprise agility. We use agility in the same spirit of how the word is defined in the English language:

"The power of moving quickly and easily; nimbleness, or the ability to think and draw conclusions quickly; intellectual acuity" – (Dictionary.com, 2009)

The use of enterprise as a qualifier, places the emphasis on how the organization as a whole possesses the attribute of agility, rather than just people and projects. 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 survive in an environment where *change is the only constant,* and yet remain disciplined enough to continue to pursue courses of action that made them successful in the first place. This notion that organizations have to possess both agility and discipline in how they create value, acts as the starting point of the research presented in this thesis.

From a software engineering research standpoint the use of the phrase 'balancing agility and discipline' was first introduced in the work of (Boehm & Turner, 2005), however, our intent is to go beyond the notion of managing the risks associated with selecting a software process model, to addressing firm-level value creation. The framework developed in this thesis, identifies the key mechanisms that software organizations use to sense changes in their environment and reconfigure their resources in response to those changes. The research is built on the premise that organizations that are able to develop unique capabilities that meet the needs of a rapidly changing operational environment are the ones that will be successful. Equally important is the notion that those organizations that can develop these unique capabilities faster than their competitors, will obtain a competitive advantage over them. Such organizations are said to possess enterprise agility. Given that the drivers of competition are many, and the associated capabilities equally large, we have bound the scope of the problem. We do so by asking three simple questions with respect to:

- How does the organization try to improve the way it does work?
- What does the organization choose to do to create future value?
- Who does the work?

Answering these questions lead us to the three mechanisms that we are interested in studying in greater details, as shown in Figure 1-1.

Software Process Improvement (SPI), refers to a set of techniques that software organizations use to understand and improve how they actually do work. These techniques fall into two broad classes: top-down approaches that focus on incorporating some predetermined best practices into the organization; and bottom-up approaches wherein the people within the organization determine the best set of improvements needed using an inductive process that builds on current organizational realities. There has been significant research into the top-down approach, especially in using models such as the CMM/CMMI (Paulk, Weber, Curtis & Chrissis, 1995), yet the outcomes of using this approach have not been conclusively proven to increase performance (Fayad & Laitnen, 1997; Staples, Niazi, Jeffery, Abrahams, Byatt & Murphy, 2007).

Agile methods grew out of the need to transition away from rigid plan-based development approaches to more flexible and adaptable software development strategies. Adopting these methods potentially leads to process improvement efforts that occur in a bottom-up fashion (Salo & Abrahamsson, 2005), and from a research standpoint have not been sufficiently explored. In Chapter 3, we use two in-depth case studies of organizations using agile methods both as their development approach of choice, as well as their primary continuous improvement strategy. We identify success factors, barriers, and provide recommendations for organizations considering the use of these methods.

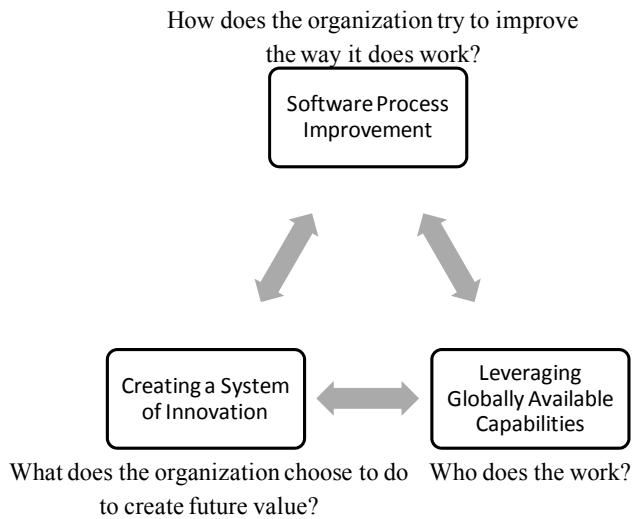


Figure 1-1 Identifying Mechanisms for Capability Generation

One of the challenges that software organizations face is in determining what they choose to work on to create future value, in other words, how they determine their future products and services. While a lot of research has talked about developing new products and services (albeit outside the software domain) (Clark & Wheelwright, 1992; Griffin, 1997), little has been done to better understand the connection between the strategic problem of creating a system of innovation, and the more tactical problem of transitioning ideas into products/services. From a strategic perspective, organizations typically use a mix of strategies ranging from developing their own ideas through research and development, to leveraging the work of others through approaches like technology scanning and open innovation. We build up the strategic discussion through an in-depth case study using publicly available data on Rockwell Collins (Section 4.3), and identify the factors that need to be considered to architect a system of innovation. For organizations to enable a smoother transition of the ideas to engineered products and services, this thesis identifies the key aspects of the problem that an

organization needs to focus on. Using a case example of a product development effort that was not successful (Section 4.5), we illustrate usefulness of the aspects. When an organization can strategically create a system of innovation (CSI), that successfully takes the ideas generated into finished products and services it will gain a competitive advantage.

When Tom Friedman (2005) wrote the book the 'World is Flat', he brought the challenges of globalization to the public consciousness. From a software organization perspective, globalization raises the issue of who actually does the work. Traditionally, the "make versus buy" decision was simply one of whether the organization could do the work cheaper internally, rather than having someone else do it. Now, the make-buy decision is much more complicated because it involves deciding not only the cost aspects, but also an understanding of the capabilities and long-term market access. A software organization today chooses to leverage globally available capabilities (LGAC) both for a cost advantage (it can get the work done cheaper by the supplier), as well as a capability advantage (the supplier has knowledge and skills that the organization itself does not have). Using an in-depth case study in Chapter 5 of an evolving customer-supplier relationship between a European client and their Indian supplier, we highlight the challenges of growing the relationship, and discuss why the Indian supplier has been successful. The lessons learned provide valuable insights into managing and evolving the relationship to be a partnership, as opposed to an arms-length contractually enforced relationship.

While the three mechanisms are important in and of themselves, the research carried out in this thesis allowed us to identify the four organizational enablers that are necessary but not sufficient precursors to enterprise agility. The four enablers focus on stakeholder alignment, employee empowerment, group and organizational learning, and governance. Analyzing across the case studies carried out in this thesis, we see that organizations that were successful leveraged one or more of the mechanisms in the presence of the organizational enablers. It is important to point out that while this thesis proposes a framework for creating enterprise agility; more research is needed to develop a generalizable theory of enterprise agility in software organizations.

In this chapter, we present a simple classification of software organizations and justify our selection of a capabilities-based view as the foundation for our research. We discuss the three broad classes of mechanisms (seen in Figure 1-1) that software organizations use to build their capabilities, and identify specific research questions that are answered in this thesis. The chapter concludes with a summary of the key contributions made through the research, and provides a high level overview of the rest of the document.

1.2 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. A simple exploration of the North America Industry Classification System (NAICS) codes that are explicitly targeted towards

software and examples of other industrial sectors that we consider to be software-intensive but are not classified so, highlights this challenge, as seen in Table 1-1,. 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. (2007) 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. 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, in the aerospace sector, the capability of a weapon systems platform is almost exclusively generated through software. It is, as we have pointed out in earlier work (Srinivasan & Lundqvist, 2005), a primary lever through which the true value of an aerospace platform is maximized.

Table 1-1 Example NAICS codes for Classifying Software Organizations (Bureau 2007)

NAICS Code	Description
511210	Software Publishers
54151	Computer Systems Design and Related Services
5416	Management, Scientific, and other Technical Consulting Services
3345	Navigational, Measuring, Electromedical, and Control Instruments Manufacturing
3361	Motor Vehicle Manufacturing

Despite this diversity at the industry sector level, software organizations can be broadly classified into three groups based on their primary value delivery approach:

- *Shrink-Wrapped Software*: Organizations belonging to this group focus on software that is often sold directly to the end consumer in the form of shrink wrapped boxes with some physical media containing software or in the form of an online download. Examples of this would include video games, productivity software such as Office, and operating systems software such as Windows or OSX. There is a rich tradition of firm-level research in the area, for instance see (Carmel & Sawyer, 1998; Cusumano & Selby, 1995). More recently, there has been a call for greater research focus on product development in this area (Xu & Brinkkemper, 2007).
- *Software Services*: Organizations belonging to this group focus on providing services that range from custom software development, packaged software tailoring, and maintenance operations. Unsurprisingly, this group makes up a significant portion of the software sector. Examples of this include organizations such as SAP and Oracle who provide enterprise solutions, to smaller and more specialized consulting firms, to outsourcing providers such as HCL and Wipro. While there has been significant

research in the area of outsourcing for application software development and packaged software tailoring, the area of software maintenance services outsourcing remains relatively unexplored.

- *Software-Intensive Systems*: Organizations belonging to this group focus on building products that have a dominant software component. These products 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. Examples of this sector include organizations such as Bosch and BAE Systems. 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.

From a sector perspective, we aim to contribute to, and extend the knowledge on, *software services* and *software-intensive systems development*.

1.3 Understanding Enterprise Agility

While the roots of the term enterprise agility are in the manufacturing context (1996), it has found wider use in the areas of supply chain management (Aitken, Christopher & Towill, 2002; Swafford, Ghosh & Murthy, 2006, 2008; White, Daniel & Mohdzain, 2005), technology management (Kivenko, 1995; Macvittie, 2006; Mafakheri, Nasiri & Mousavi, 2008), information technology management (Alter, 2007; Saran, 2005; Thompson, 2005; White et al., 2005), services industries such as insurance, healthcare (Fisher, 2007; Maciag, 2008; Wall, 2005), workforce management (Bellinger, 2006; Sherehiy, Karwowski & Layer, 2007), and acquisition management (Chatzkel & Saint-Onge, 2007).

The importance of agility in enabling enterprise competitiveness has been emphasized in multiple industrial sectors, as seen in Dove's survey of change proficiency issues (Dove, 1996). The top nine change proficiency issues across eight industrial sectors that he identified reflect the importance of organizational sensemaking (Weick, 1995), and responseability (Dove, 2001). Roth (1996) notes that organizations can achieve strategic agility by using 'economies of knowledge', i.e. the organization can leverage its business knowledge in combination with human and technical skills to consistently identify, assimilate, and exploit knowledge better than its competition. In a similar vein, Grantham and Williamson (2007) define an agile organization as one that strategically integrates the management of its real estate, human resources, and technology assets. As they point out, the organization can be understood at three levels that center on notions of completion, survival, and renewal that answer the questions of:

- What is being done?
- How it is being done?
- How the organization can create new capabilities?

While growth can emphasize the need for agility, as was the case with Motorola in the late 1990s (Stopper, 1998), a downturn as the one we are currently facing globally places a premium on generating enterprise agility.

In order to understand how software organizations create value as a whole, we have to understand the underlying models of how these organizations compete in the market. 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, and is representative of the environment which software organizations operate in (Nelson & Winter, 1982). The underlying idea of viewing the firm as a bundle of resources that provide strategic advantage (Wernerfelt, 1984) provides a foundation for studying these firms. An extension of this approach, the dynamic capabilities approach that was first articulated by Tecce, Pisano and Shuen (1997), who define:

> *"Dynamic Capabilities are 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 by Eisenhardt and Martin (2000), 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."*

Firms in the software industry have commonality in that they:

- operate in fast clock speed environments (Fine, 1998),
- use project teams as the primary means of creating value (Faraj & Sproull, 2000),
- leverage a process which forms one the purest forms of knowledge work (Blackler, 1995; Griffith, Sawyer & Neale, 2003; Zmud, 1984).

These characteristics coupled with the Schumpeterian competitive environment (Schumpeter, 1927) that these organizations operate in, make the dynamic capabilities construct 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 by (Arora, Arunachalam, Asundi & Fernandes, 2001; Athreye, 2005) and (Ethiraj, Kale, Krishnan & Singh, 2005). In related work, the notion of enterprise agility has been explored in the software organizational context by (Overby, Bharadwaj & Sambamurthy, 2006) who define agility from the perspective of information technology, and (Kettunen, 2007, 2009), who brings together the notion of project agility and new product development agility in software organizations. We define enterprise agility as an organizational attribute that reflects:

> *"The ability of the organization to sense changes in both its internal and external environments, and a result reconfigure existing resources to develop additionally needed capabilities to meet those needs."*

This definition makes explicit the connection between the two ideas of enterprise agility and dynamic capabilities, with enterprise agility being an outcome of a firm developing and reconfiguring its capabilities. This thesis explores the mechanisms that software organizations use to develop capabilities and identifies the necessary organizational enablers that support the development of enterprise agility.

1.4 Mechanisms of Capability Development

Our definition of enterprise agility is predicated on the ability of an organization to evolve/adapt its capabilities. As we noted in section 1.1, the mechanisms that organizations use can best be understood by answering the how, the what, and the who questions: how an organization chooses to improve the way it works (SPI), what it choose to work on to generate future value (CSI), and who does the work (LGAC), as shown in Figure 1-1. We can connect those mechanisms to the underlying requirements for capability development, as shown in Table 1-2. Each of the mechanisms contributes to the creation of capabilities to meet short-term or long-term needs, and have execution time horizons that are immediate or incremental.

Table 1-2 Connecting Mechanisms to Capability Development

Mechanism	Needs	Execution	Location
SPI	Clear understanding of long term needs	Incrementally developed over a long time horizon	Within firm boundaries
CSI	Defining long term needs	Mix of near-term, and long-term horizons	Within & across firm boundaries
LGAC	Meeting short-term needs, evolving to long term needs	Near-term time horizon	Across firm boundaries

When the organization has a clear understanding of how it does work, it uses process improvement as a means of improving and building additional capabilities. SPI alone does not guarantee the long term sustainability of the firm. For example, organizations involved in Y2K related work (essentially COBOL programming) either died out (post 2001) since there was no work, or built capabilities through training of personnel in other technologies such as Java and .Net to smoothly transition into related markets. The latter set of organizations treated it as an improvement effort that was built on the idea that programmers who were skilled in one language could be easily transitioned into a different programming language. A

more conventional software process improvement effort would be introducing a new requirements management process in the organization to meet client specific requirements.

The question of what work an organization does in the long run is a function of its core capabilities and its ability to generate new products and services that can drive the evolution of the organization – i.e. the organization's ability to innovate. Innovation is by necessity a mix of meeting short term and long term needs. While organizations have accidentally innovated to create a product or service, for the most part, they have to create a system of innovation that spans firm boundaries to create new products and services. This mix of open innovation, internal R&D and technology scanning enables the organization to develop capabilities that it needs to be sustainable in the long run. It is important to note here that innovation can occur from SPI and LGAC.

When an organization recognizes that it does not have the capabilities that it needs, or cannot afford to exploit the capabilities that it possesses, it has to look outside the firm boundaries to identify capabilities that it can exploit. This leveraging of globally available capabilities is done through the use of a mix of offshoring, nearshoring, and outsourcing. In most cases, the emphasis is on meeting short term needs such as cost reduction, but as the organization recognizes the advantages of exploiting an external capability base, the approach becomes part of their long term strategy. One of the challenges that emerged from trying to identify the mechanisms at the firm level is the need to find a research approach that can be applied across seemingly different domains that operate at different time horizons and in some cases across multiple geographic sites. The mechanisms themselves when individually examined from the perspective of earlier research and the opportunity to create new knowledge yield the three independent research questions that are answered in this thesis.

1.4.1 Software Process Improvement (SPI)

Organizations involved in the design, development and sustainment of software systems have to improve their processes in order to remain successful in a dynamically changing environment. In addition to managing the rapid changes in technology, they have to address the needs of customers whose product/service expectations improve constantly, and employees whose skills are transferable to their competitors. Given that software development is a non-routine complex undertaking requiring high levels of competence and a flexible organizing structure, the fundamental issue for software organizations is how to achieve a balance between control and goal orientation on one hand, and change and flexibility on the other. As Aaen (2003) points out, using blueprints for software process improvement emphasizes formal models at the expense of process user knowledge. By structuring software process improvement (SPI) efforts towards generating organization capabilities, we contend that it becomes easier for the organization to satisfy their key stakeholders, achieve a balance between the need for predictability achieved through standard work, and innovation from flexible and adaptable processes.

The dominant approach to carrying out SPI has been to use normative models such as ISO 9000 (Oskarsson & Glass, 1995), CMM (Paulk, Curtis, Chrissis & Weber, 1993), CMMI (Chrissis, Konrad & Shrum, 2003) or SPICE (El Emam, Melo & Drouin, 1997), which specify collections of best practices, that are sometimes stratified across maturity levels. SPI represents the most common approach that software organizations use to incrementally enhance their capabilities. Jones (1996) identifies three gaps in knowledge on software process improvement with respect to the cost of software improvement, the time taken to make tangible improvements, and value expected in terms of quality, productivity, and or user satisfaction. The six-stage process that he identified focuses on base lining, management, processes and methodologies, new tools and approaches, infrastructure and specialization, reusability, leading to industry leadership. Organizations using this approach carry out an assessment of the current state of software processes in the organization, make a determination of improvement opportunities against some selected best practices (such as those specified when transitioning between levels of maturity), and execute improvement projects to incrementally build the requisite software lifecycle capabilities. This approach of normatively guided, top-down, policy-directed improvement efforts demonstrated successes in the early self-reported case studies of Hughes Aircraft (Humphrey et al., 1997), Raytheon (Haley, Syst & Marlborough, 1996), and Motorola (Diaz, Sligo, Inc & Scottsdale, 1997). The first survey based studies on CMM adoption (Herbsleb, Carleton, Rozum, Siegel & Zubrow, 1994; Herbsleb & Goldenson, 1996), found that organizations that were successful in improving their maturity levels demonstrated significant gains in cost & schedule performance, product quality & productivity, and customer satisfaction & employee morale. It is important to note, that even in this survey of early adopters, 44% of the respondents had found little to no success in their improvement efforts. More recently, there has been increased scrutiny on using CMM based SPI as the dominant strategy for improvement (Fayad & Laitnen, 1997; Staples et al., 2007). Inductive process improvement approaches such AINSI (Briand, Eman & Melo, 1998) take a bottom up approach to software process improvement. These approaches focus on the task, process and resource (TPR) aspects of software development (Jakobsen, 1998), and as a result are able to create traction across the enterprise. Focusing on just the TPR aspects could potentially result in a series of localized efforts that do not translate into enterprise benefits. For a crisp discussion of top-down versus bottom-up process improvement strategies, see (Thomas & McGarry, 1994).

Agile software development approaches provide an alternative approach to thinking about the challenges of SPI. Bailetti and Liu (Bailetti & Liu, 2003) use information theory to develop criteria for comparing plan-based developmental approaches such as those implementing the CMM framework and eXtreme programming. By framing the design cycle as a function of solution knowledge, system requirements, implementation knowledge, design language and statements, they found that the CMM requires more solution knowledge than the XP team, and the rate at which the XP team converted solution knowledge to requirements was constant, while that for the CMM team increased its solution knowledge over time. In effect, they showed that the question that needed to be asked was not if the processes were better, rather that the question was *which process was better suited to which phase*. This finding is

consistent with our understanding of agile methods, i.e., agile methods are effective at specifying and supporting the actual work, while heavyweight frameworks support the managerial framework needed to understand, asses and drive policy-directed change. Despite this growing body of research on agile adoption, there is limited understanding on how the adoption of these methods can be sustained, leading to the first research question we aim to answer in this thesis:

Question 1: How do software organizations adopt and sustain agile methods?

1.4.2 Creating Systems of Innovation (CSI)

Innovation to a large extent defines what the organization chooses to do both in the immediate short-term, as well as in the long-run. Innovation has been recognized as a critical element for survival in the current operational environment; yet, the debate remains ongoing as to exactly what innovation is, and how it can be fostered within an organization. The discussions surrounding innovation have led to the creation of buzzwords like the 'creative economy' (Coy, 2000), as well as new organizational roles at the corporate level, such as chief innovation officer (Nussbaum, 2005). The importance of innovation for organizational success can be traced back to (Schumpeter, 1927), wherein he defines innovation as

> *"Changes of the combinations of the factors of production as cannot be effected by infinitesimal steps or variations in the margin. They consist primarily in changes in the methods of production and transportation, or in the production of a new article, or in the opening up of new markets or of new sources of materiel."*

This definition of innovation has stood the test of time, and while ideas like entrepreneurship have emerged in more recent time, the essence of innovation remains the same. As Drucker (1985) points out almost 50 years after Schumpeter when discussing innovation,

> *"It is the means by which the entrepreneur either creates new wealth-producing resources or endows existing resources with enhanced potential for creating wealth."*

In other words innovation can be in the product or the process (Tushman & Nadler, 1986). It can be classified based on the whether it is incremental or radical (Ettlie, Bridges & O'keefe, 1984), or modular or architectural (Henderson & Clark, 1990). At the heart of the innovation puzzle is the ability to connect the strategy & tactics associated with developing a system of innovation from a macro perspective, with the mechanics of effectively transitioning ideas into finished products and services at the micro-level. This leads to the second question addressed in this thesis:

Question 2: How can software organizations create a system of innovation that successfully addresses strategic challenges while meeting operational needs?

1.4.3 Leveraging Globally Available Capabilities

In the guest editors' introduction to the IEEE Software special issue on Global Software Development (Herbsleb & Moitra, 2001), note that

> *"Economic forces are relentlessly turning national markets into global markets, and spawning new forms of competition and cooperation that reach across national boundaries".*

In discussing the drivers towards increased geographical distribution of work, they highlight the five factors of:

- needing to capitalize on the global resource pool;
- the business advantage of proximity to market;
- exploiting market opportunities through virtual corporations and virtual teams;
- the severe pressure to improve time-to-market by leveraging 'round-the-clock' development;
- the need for flexibility to capitalize on mergers and acquisitions.

When firms attempt to address these issues by leveraging globally available capabilities, the resultant environment is multi-site and multi-cultural, leading to the organization having to address not only technical issues, but also social and cultural challenges.

The customer-supplier relationship in software development is as (Brereton, 2004) notes, 'a delicate, dynamic relationship that changes with time, demands, and different approaches to procurement'. The role of inter-organizational trust in outsourcing has been emphasized as a critical enabler to building sustainable relationships. Building on earlier work, Lee, Huynh and Hirscheim's study (2008) found that mutual trust was important for knowledge sharing and outsourcing success. In the case of product sustainment, this becomes even more critical, because the client provides the domain knowledge, while the supplier provides the technical expertise needed for effective problem solving. Building on transaction cost economics and the resource based view of the firm to determine the extra costs associated with outsourcing, (Dibbern, Winkler & Heinzl, 2008), further highlight the importance of knowledge management, team competence, and governance structure in identifying and mitigating risks, as well as mitigating unforeseen costs. Despite this growing body of research knowledge, little work has been done in the maintenance services arena that addresses both the client and the supplier perspectives. This leads to the third question we answer in this thesis:

Question 3: How does the customer supplier-relationship evolve in maintenance services outsourcing?

1.5 Case Studies

The case studies that we carried out as part of the research spanned a spectrum of organizations ranging from software services organizations in the financial sector to game development to aerospace software development, as shown in Table 1-3. Of the eight case studies presented in this thesis, seven involved fieldwork, while one (Rockwell Collins) was carried out using only publicly available data.

Table 1-3 Overview of Case Study Organizations

Case Study	Domain	Location	Methods
AgileCo	Software Services	India	Interviews, Observation
BankCo	In-house financial services	India	Interviews, Observation
EuroTel	Telecommunications	Europe	Interviews, Observation, Archival Data
FinServicesCo	Outsourced financial services	India	Interviews
GameDevCo	Game Development	Europe	Interviews, Observation, Archival Data
IndiaCo	Outsourced maintenance services	India	Interviews, Observation, Archival Data
Rockwell Collins	Avionics	Global	Publicly Available Archival Data
SpaceCo	Aerospace Software	USA	Interviews

1.6 Contributions

As was noted in Section 1.3, enterprise agility has been applied in multiple industry domains, and in multiple contexts. In this section, some of the key papers that relate to this thesis are presented, and the similarities and differences between them (with respect to the thesis) are highlighted.

Breu et al. (2002) used survey data from 515 organizations in the United Kingdom to determine the capabilities of an agile workforce. They found that an agile workforce acquire the five capabilities of intelligence (responsiveness to changing customer and market demands), competencies (speed of developing new skills and competencies, which include both technical skills, as well as the soft skills needed for business process change and

management) collaboration (ability to work across functional boundaries, and across projects), culture (empowerment for independent decision making) and information systems (infrastructure for introducing new information systems). Their finding that the first two capabilities are critical, is consistent with the needs of software organizations, and further supports the emphasis on people and organizational issues in this thesis. In reflecting on the challenges of speeding up learning, Mindrum (2008) notes that getting people thinking and acting in new ways is the primary 'drag' on an organizations perfect speed. In case of software organizations, the faster the organization learns, the easier it can evolve and adapt. This thesis emphasizes learning to be a key enabler for enterprise agility. Shafer et al. (2001) build on earlier work on the gap between human resource strategy and organization agility (Dyer & Shafer, 1998) through a case study of AEHN's approach to developing an agile workforce. The focus on developing employees who were business-driven, value driven, focused, generative, and resilient, was driven by the need to manage the turbulence in the heath care industry. This emphasis on people is emphasized in this thesis as well, in identifying stakeholder alignment as a key element for achieving enterprise agility.

In the context of software organizations, the term agility is often associated with the production agility associated with 'lightweight' or 'agile' methods. Hanssen and Fegri's (2008) case study on how an organization integrated software product line engineering (SPLE) and agile software development (ASD) highlights the importance creating a synergistic approach that bridged strategic, tactical, and operational, organizational objectives. As they note, by integrating SPLE and ASD, CompNN was able to create a holistic cycle of continuous improvement. Chin (2004) focuses on mechanics, enablers and barriers to successful agile project management, while implicitly assuming enterprise-level agility as an outcome. Their focus on process improvement is incorporated as one of the three mechanisms for enterprise agility presented in this thesis.

In their analysis of the evolution of Nokia, Doz and Kosonen (2008) formulate the notion of strategic agility along the axes of strategic sensitivity, resource fluidity, and leadership unity. More importantly, they note that these meta-capabilities are multiplicative, and that the lack of one or more of them severely degrades the organizations ability to exploit the advantages of strategic agility. Kettunen (2007) frames agility as a prospective strategy for organizations involved in new product development, and emphasizes the need for combining project level agility with organizational level agility. In that paper, he explores the connection between software process improvement and enterprise agility in the context of new product development. This thesis adds to that body of knowledge and further expands the knowledge base to include two other approaches for gaining enterprise agility, namely, innovation and global sourcing.

Hoogervorst (2004) proposes an alternative approach to gaining enterprise agility through the use of enterprise architecture to bridge the gap between strategy and execution. His approach emphasizes the need for an integrated, consistent, and coherent approach that bridges across the business, organization, technical, and information design. Similarly, this thesis posits that the three mechanisms that software organizations use to generate enterprise

agility are overlapping, and that as an organization deploys one mechanism, it will support/enhance other existing mechanisms towards holistically creating agility. Osborne (1998) discusses the 'paradox' of competitive agility, i.e., the need for an organization to be flexible (to respond to emergent threats), and yet stable (to learn and grow). In the case of our research, we believe that the paradox does not exist, as the mechanisms that we have identified provide both flexibility and stability, depending on how they are implemented and exploited by the organization.

In this thesis, we make five contributions to the existing body of knowledge in software engineering. They include:

Contribution 1:	*Identifying success factors to enable the adoption and sustainment of agile methods by software organizations*
Contribution 2:	*Developing a framework for creating a system of innovation that addresses strategic and tactical issues*
Contribution 3:	*Understanding the evolution of the customer-supplier relationship in maintenance services outsourcing*
Contribution 4:	*Identifying the organizational antecedents of enterprise agility in software organizations*
Contribution 5:	*Proposing a theory of enterprise agility in software organizations*

The approach that we have adopted for studying software organizations emphasizes the need for engaging key stakeholders in defining the research problem, and being active participants in the research process. This approach enables us to increase the validity of the findings, and more importantly, construct a value proposition that is beneficial to both the researcher and the organization under study. The intent is to determine the socio-technical challenges that software organizations face as they attempt to gain enterprise agility through the use of one or more of the mechanisms discussed in previous sections. The hypothesis generating research that we carry out in this thesis requires a mixing of multiple data gathering approaches to create a rich/thick description of the organizations studied, supported by rigorous data analysis and cross-context theorizing. The mixed methods approach that we developed and validated in the field, enables the researcher to gain access to the research site, obtain useful data, and generate usable theories of software organizations.

Using rich descriptions of agile adoption in two organizations GameDevCo and AgileCo, we identify the common pitfalls with respect to the adoption of agile methods, and further deepen our analysis through a set of best practices observed at the more plan-based development organizations EuroTel and IndiaCo. While a lot has been written about deploying agile methods in general, little research has been conducted on looking at how agile methods can be sustained to the point that they are part of the organizational DNA. The

first contribution of this thesis is providing a deeper understanding of how agile methods can be adopted and institutionalized by software organizations. The second contribution of this thesis is the development of a framework for creating a system of innovation that addresses strategic and tactical challenges. We develop the strategic perspective through our case study of Rockwell Collins, and gain a deeper understanding the tactical problem of transitioning ideas into engineered products or services. The framework was born out of conversations with senior aerospace engineers at SpaceCo, who had been phenomenally successful in generating ideas, but had struggled to bridge the gap between the individual innovators, and the collective engineering team. In applying the newly developed framework in understanding the dynamics of new product development at a game software organization , GameDevCo, we were able to obtain further validation that it covered the aspects that senior leadership should be cognizant of when managing high-risk new product development efforts. In looking at evolution of the relationship between EuroTel and IndiaCo, we were able to develop important lessons learned about path dependence, knowledge management, and the impact of the client capabilities in the evolution of the customer-supplier relationship in the maintenance services context. Our study of IndiaCo provides best practices on how to grow a software organization and identifies best practices for managing an Indian supplier.

The fourth contribution made in this thesis is the identification of the organizational antecedents of enterprise agility. By analyzing across the in-depth case studies, we identify the four key organizational antecedents of: stakeholder alignment, employee empowerment, group & organizational learning, and governance mechanisms that are necessary to enable enterprise agility. Our case studies provide illustrative examples of how the mechanisms of process improvement, systems of innovation, and leveraging globally available capabilities, provide enterprise agility. The fifth contribution made in this dissertation is the theory of enterprise agility in software organizations. The mechanisms in the presence of the organizational enablers provide software organizations with the ability to generate capabilities that they can then exploit in a changing market environment.

Table 1-4 Mapping Thesis Structure to Questions and Contributions

Contribution **Question**	**C1**	**C2**	**C3**	**C4 & C5**
Q1 SPI	Chapter 3			Chapter 6
Q2 CSI		Chapter 4		Chapter 6
Q3 LGAC			Chapter 5	Chapter 6

The research questions and contributions can be tied back to the thesis structure, as shown in Table 1-4. The left most column identifies the question number, and the mechanism that the research question focuses on, while the top row identifies the research contribution number. The chapter numbers within the table identify the primary chapters that connect the

contributions to the questions asked. The contributions made in this thesis emerged from a series of papers and presentations that were reviewed by the larger academic community. The specific aspects of the thesis that can be found in the papers is shown in Table 1-5. Additionally, there are four conference papers, and two journal papers that were derived from the thesis, which are currently under review.

1.7 Dissertation Outline

The remainder of the dissertation is organized into six additional chapters, three appendices, and the bibliography.

In chapter 2, we present the approach that we have adapted, tested, and validated over the last five years of carrying out the research. Our approach uses engaged scholarship as the overarching paradigm for carrying out mixed-methods process research. By using a blend of interviews, observations, and archival sources for data gathering; and ground theory and comparative case analysis for theory generation; our approach supports deeper insights into the firm-level socio-technical issues. In addition to providing illustrative examples of these insights, the chapter also presents lessons learned from actually applying the methodology in practice.

Table 1-5 Mapping Publications to Thesis Structure

Publication	Thesis Chapters
(Srinivasan, 2008c)	Chapter 2, Chapter 5
(Srinivasan, 2008a)	Chapter 5
(Srinivasan, 2008b)	Chapter 4
(Srinivasan, Dobrin & Lundqvist, 2009)	Chapter 3
(Srinivasan, Lofgren, Norstrom & Lundqvist, 2009)	Chapter 5
(Srinivasan & Lundqvist, 2009a)	Chapter 3, Chapter 6
(Srinivasan & Lundqvist, 2009b)	Chapter 3
(Srinivasan 2009)	Chapter 4
(Srinivasan, Norstrom & Lundqvist, 2009)	Chapter 2, Chapter 6
(Srinivasan & Lundqvist, 2010)	Chapter 3

In chapter 3, we answer the first research question with respect to agile adoption and sustainment using in-depth case studies of AgileCo and GameDevCo. We position the study

in the literature on agile adoption and present the findings of the three pilot case studies that were used to guide the in-depth case studies. Since AgileCo and GameDevCo represent a polar case with respect to success in adopting agile methods, the studies individually, and the cross-case analysis provides useful insights from both a theory and practitioner perspective.

In chapter 4, we answer the second research question with respect to understanding the problem of transitioning ideas into engineering products and services. Beginning with the challenge highlighted in the pilot case study at SpaceCo of transitioning 'sandbox development' into an engineered safety-critical product, the framework developed that decomposes the problem along organization, knowledge management, innovation and process dimensions is presented. Using the case of new product development at GameDevCo, we illustrate how the framework will provide insights to senior leadership on the key aspects that they need to pay attention to.

In chapter 5, we answer the third research question with respect to leveraging globally available capabilities in the context of software maintenance outsourcing. The embedded case study captures the perspective of the client, EuroTel, and the supplier IndiaCo, from the strategic lens as well as the project execution lens. Furthermore, we explore the explicitly stated dynamics of the relationship through a workshop that was conducted with stakeholders from both organizations. We highlight the factors that made IndiaCo successful, illustrate how history repeats itself at EuroTel. The lessons learned from the case provide insights on the areas of organizational learning, human capital management, and strategic relationship management.

In chapter 6, we develop a theory of enterprise agility in software organizations. Analyzing across the in-depth case studies presented in the earlier chapters, we identify the four key organizational enablers to obtaining enterprise agility to be: stakeholder alignment, employee empowerment, group & organizational learning, and governance mechanisms. Our case studies illustrate how the mechanisms of software process improvement, systems of innovation, and leveraging globally available capabilities, could provide enterprise agility to software organizations. The theory we develop posits that software organizations can effectively leverage these mechanisms in the presence of the four enablers to effectively & efficiently gain enterprise agility.

In chapter 7, we revisit the research questions, and illustrate how we answer them through the work presented in the dissertation. We make explicit the key contributions of the research, and identify areas of future research.

In the three appendices, we present the complete list of publications relating to this thesis, share the current state of the research protocol, and provide greater insight to the data analysis procedures adopted.

Finally the bibliography presents the references cited in this document.

2 RESEARCH DESIGN

2.1 Introduction

The focus of research in software engineering has been driven by the 'research-then-transfer' model as opposed to the 'industry-as-laboratory' approach (Potts 1993). This focus on the "research-then-transfer" model has resulted in the adoption of positivist approaches that focus on the phenomena at the task, tool or team level, instead of studying organization-wide challenges. The recent ICSE workshops (Dittrich et al., 2008; Sim, Singer & Storey, 2001) have highlighted that the need still exists for innovative research approaches to study broader scale socio-technical phenomena. In this chapter, we discuss the approach that we have developed and tested over the last five years to drive and support our own research. The approach (shown in Figure 2-1) is built around engaged scholarship (Van de Ven, 2007) as the guiding philosophy, and utilizes a process research design for understanding the evolution of a phenomenon. The data collection is executed using interviews, observations, and archival information, while grounded theoretic and comparative case analysis techniques are used for describing the phenomena in its context and to support theory development.

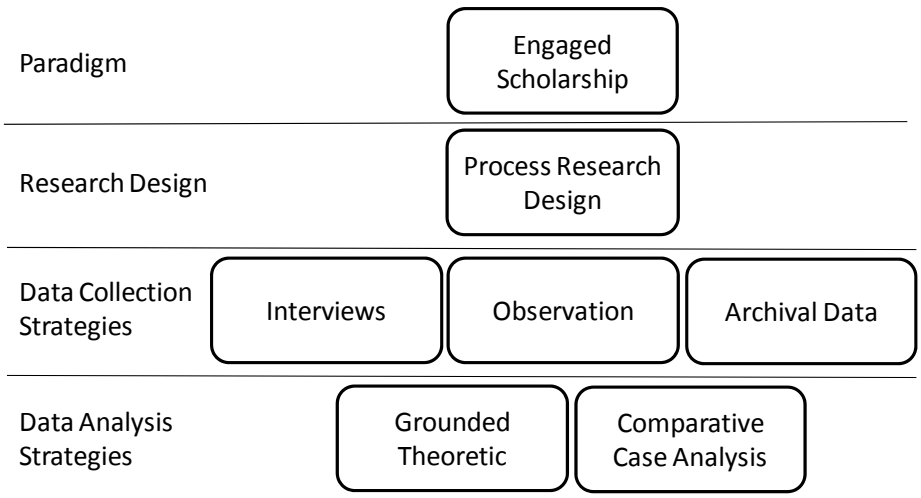


Figure 2-1 Approach to Developing Methodology

Van de Ven and Johnson (2006) define engaged scholarship as:

> *"A form of inquiry where the researchers involve other key stakeholders, and leverage their different perspectives to learn about a problem domain. Furthermore,*

the approach also refers to an identity with respect to how the scholar views his/her relationships with their communities and their subject matter."

The goal with exploring the sources of enterprise agility in software organizations is to work in a problem area that is relevant to industry, and at the same time, develop/extend the existing body of knowledge across multiple disciplines. The critical realist perspective (Archer, Bhaskar, Collier & Lawson, 1998) that engaged scholarship embraces, is one that deeply resonates with our constructivist philosophy of teaching software engineering (Srinivasan & Lundqvist, 2007).

We use the word process along the same lines of (Van de Ven, 1992), 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 individuals and organizations;
3. a sequence of events that describes how some entity or 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. It is important to note here that the three research questions that we are interested in exploring are in fact process oriented questions, providing a first-pass sanity check on having an appropriate design.

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 that covered the two dimensions (shown in Figure 2-2) of:

- Development Process adopted,
- Primary Value Creation Approach.

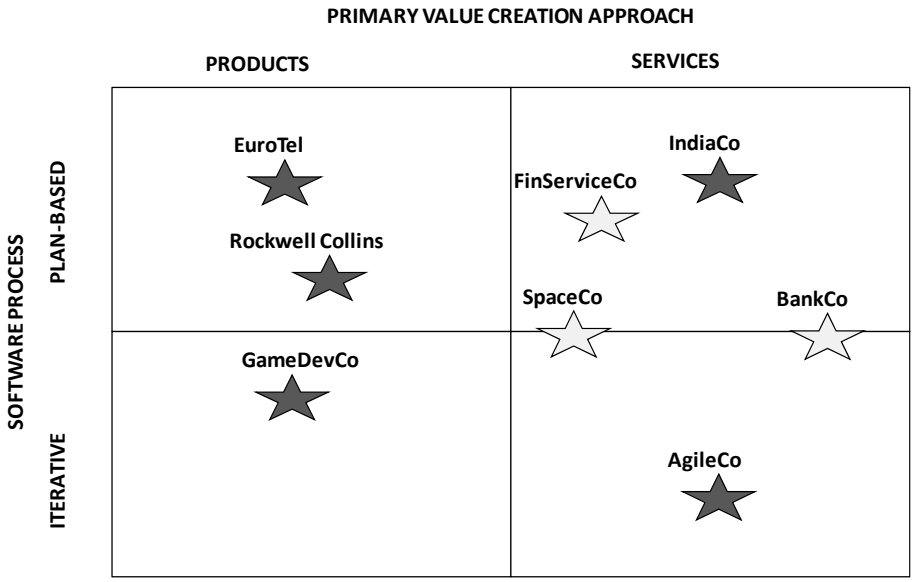


Figure 2-2 Multiple Case Sampling Strategy

Given the exploratory nature of our work, we selected pilot case studies that would provide insights and guidance in evolving the research protocol (the current version can be found in Appendix B). The selection of the cases was strategized ahead of time, and allowed to evolve as the case studies progressed. The mapping of the case sites to the research questions is shown in Table 2-1

Table 2-1 Connecting Cases to Research Questions

Research Question	Exploratory/Pilot Case	In-Depth Case
SPI	FinServicesCo, BankCo, SpaceCo	AgileCo, GameDevCo
CSI	SpaceCo	Rockwell Collins
LGAC	FinServicesCo, BankCo	IndiaCo, EuroTel

Our first round of exploratory case studies involved three organizations FinServicesCo, BankCo, and SpaceCo. The FinServicesCo and BankCo studies provided the insights needed to further explore agile adoption using the in depth case studies of AgileCo and GameDevCo, as seen in Chapter 3. The insights from Rockwell Collins and SpaceCo led to the creation of the theoretical framework, and first pass validation presented in Chapter 4. The two in depth studies of EuroTel and IndiaCo were carried out at the later stages of the research, at which

point the methodology had matured significantly, and provided the core data presented in Chapter 5. All five in-depth case studies supported the theory generation in Chapter 6.

For data collection, we used a mix of semi-structured interviews (Seidman, 2006), observations (Spradley, 1980) that were captured in the form of field notes (Sanjek, 1990), and archival data. The intent behind using semi-structured interviews (interviews wherein the interviewer has the flexibility to adapt/morph both the questions and the sequence of questions to gain deeper insights), was to go gather richer data from the participants by providing them with an opportunity to reflect more broadly from both their individual perspective, as well as from the organization context as a whole. This structure also provided us with the flexibility of exploring specific areas in greater depth, during the later rounds of fieldwork. Most of the interviewees consented to be taped during the interviews. In the few cases where we could not get consent due to organizational policies, personal preferences, or during closed door meetings with the senior leadership teams, we captured field notes that later became invaluable sources of insights. We used observation to further enhance our understanding of the organizational culture and social dynamics, i.e. as validation sources, rather than as the primary sources of insights.

The two predominant approaches that we used to conduct the data analysis were grounded theory (Glaser & Strauss, 1967) and comparative case analysis (Perrow, 1967; Ragin, 2007). 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 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), the use of comparative case analysis in addition to the grounded theoretic approach proved useful.

In the remainder of this chapter, we provide a synthesis of the data collected during the research, present an overview on the use of grounded theory in software engineering research, and discuss our approach to doing grounded theoretic analysis. We present the lessons learned over the last five years, and conclude by presenting the overarching approach that we have found to be effective in studying software organizations.

2.2 Data Collection

The data collection strategy used in this thesis relies on leveraging multiple sources of data to gain a trustable understanding of the phenomena under study. The data sources include the use of interviews, observation, and archival data, as shown in Table 2-2. The research protocol, given the exploratory nature of the thesis, evolved over the period during which the research was carried out, the current version of the research protocol can be found in Appendix B. It can be used by other researchers either to extend the work, or to find gaps in the theory developed. The evolution of the protocol is a reflection of the increases in scope of the research since its initial conception as purely about balancing agility and discipline in

software development. The protocol also reflects the need for adaptability and flexibility in later cycles of the research. Take for instance the evolution of the research at GameDevCo – the first round of interviews included three senior leadership interviews, and three team interviews among the total interviews. It was the learning from these first round interviews, the resulted in the addition of a few targeted questions in the third round of fieldwork (as seen in Table 2-3.

Table 2-2 Data Collection Strategies in Each of the Cases

Case Study	Data Collection Strategy
AgileCo	Round 1: 5 Interviews Round 2: 2 Interviews, 3 Observations from teaching Round 3: 5 Interviews, 2 Observations from office interactions
BankCo	7 Interviews, 1 Observation
EuroTel	Round 1: 7 Senior Leadership Interviews Round 2: Workshop Observations and Data Round 3: 10 project member interviews, 2 senior leader interviews
FinServicesCo	5 Interviews
GameDevCo	Round 1: 22 Interviews (including 3 team), 2 Observation Round 2: Archival Data on Policy Directives Round 3: 4 Interviews Round 4: Senior Leadership Briefing
IndiaCo	Round 1: Workshop Observations and Data Round 2: 16 interviews Round 3: 7 team interviews, Archival data on root cause analysis Round 4: Senior Leadership Briefing
Rockwell Collins	Publicly Available Data
SpaceCo	2 Interviews

Table 2-3 Specific Question in Round 3

1. What % of design changes are captured in some form (twiki, change logs etc)?
2. Is it done in a timely manner?
3. What % of design information do you think is lost?
4. How can this process be made better?

Observations, both direct and indirect were captured in the form of filed notes and we leveraged in the analysis. For instance the field notes made when teaching with AgileCo team members became one of the key discussions points used to triangulate and support the

argument for indoctrination and culture. In another field note, written when team members at AgileCo were arguing about RUP, is shown in Table 2-4. The final approach that we leveraged for data collection was archival data that was either obtained from public newspapers, magazines, and the organization's website, or from the organization itself in the form of training documents, published policies, and lessons learned documentation.

Table 2-4 Field Notes from RUP Argument

During dinner today, I was exposed for the first time to the level of 'indoctrination' that AgileCo members go through. Part of it, is the fact that the people arguing against RUP were 'born agile' - they had never been exposed to anything else, and the training at AgileCo does border on the dogmatic. Another possible explanation is that the person bringing up RUP is a RUP certified trainer, and is probably trying to drum up business for himself, and therefore highjackign a pleasant dinner. [need to investigate if they support non-agile practices at AgileCo] [[They do, in the form of spikes for problem solving, and with A's project management where he has an internal conversion factor between the story points used, and the cost in man hours]]

2.3 Data Analysis

The data analysis in this thesis is carried out using a grounded theoretic analysis approach for firm level analysis, and comparative case analysis for cross-case analysis. The use of the word grounded-theoretic is intentional in that we leverage the strengths of grounded theory in developing theories rooted in the data itself, while avoiding the weaknesses that come from a 'researcher as a completely blank slate perspective'. In this section, we present an overview of grounded theory, followed by our approach to grounded theoretic analysis.

2.3.1 Grounded Theory

Glaser and Strauss' landmark book (1967) entitled 'The Discovery of Grounded Theory', based on their research on dying hospital patients, introduced the term grounded theory into the vocabulary of researchers in the social sciences. A common understanding that one needed to get out in the field if he/she wanted to understand what was going on, and the importance of having a theory that is grounded in reality shaped their work. Their emphasis on the evolving nature of experience in the field, coupled with the active role of people in shaping the worlds they live in, led to an emphasis on the change and process that leads to the

variability and complexity of life. As Goulding (2002) points out, the guidelines that they specify for the developed theory are that should:

- enable prediction and explanation of behavior;
- be useful in theoretical advances in sociology;
- be applicable in practice;
- provide a perspective on behavior;
- guide and provide a style for research on particular areas of behavior;
- provide clear enough categories and hypotheses so that crucial ones can be verified in present and future research.

The grounded-theoretic approach that we adopted was the framework suggested by Charmaz (2006) wherein she defines grounded theory methods as systematic, yet flexible guidelines for collecting and analyzing qualitative data to construct theories that are grounded in the data themselves. These guidelines of using grounded theory methods are a set of general principles and heuristic devices rather than formulaic rules – to put it simply, the data forms the foundations, and the analysis generates the concepts that are constructed. Unlike the common misconception that grounded theory methods require the researcher to be a blank paper, we follow Glaser's guidance on leveraging extant theory for sensitising the researcher to the conceptual significance of the research findings (Glaser, 1978). Furthermore, the notion of conceptualizing processes as unfolding temporal sequences that have identifiable markers with clear beginnings and ends is intuitively appealing and supports practical analysis. By identifying these markers, and linking together the various temporal sequences, we are able to understand how the processes lead to greater enterprise agility.

While grounded theoretic approaches have a long history of use in the social sciences, they are only now beginning to appear for studying software organizations. The use of these approaches can be found in studies of information systems development, virtual teams, new product development, addressing specific stages in the software life cycle (requirements, architecture, testing), and software processes in the large. Each of these areas is of interest to our overarching research on enterprise agility.

In his review of the use of grounded theory methods in information systems development, Bryant (2002) notes that research into systems has to take into account knowledgeable social actors and their stocks of knowledge. More importantly, he notes that the process of research has to use a process of engagement and collaborative construction that involves both the participants and the researchers. Milis, Viaene, and Ribbers's analysis of 12 large IT projects in the bank and insurance industry found that the feasibility study impacted the nature of the justification and selection process for projects (Milis, Viaene & Ribbers, 2006). Their interviews with 45 stakeholders found that while the feasibility study is considered a critical

part of governance, it is often not done thoroughly depending on the focus of the project (internal or external) and the project trigger. We built upon the insights of both in how we have developed and implemented our research - the engagement stage of our research approach not only ensures that the problem is defined collaboratively, but also that it is feasible, and delivers value to both the researcher and the organizational stakeholders involved.

Pauleen and Harmer (2008) combine narrative inquiry and grounded theoretic approaches to understand the transformative effects of mobile technologies on the way users work, and how the technology changes user relationships with the work. Their approach of being theoretically agnostic in data gathering is similar to our approach in the first round of interviews. Being theoretically agnostic in the pilot/preliminary stage provided support and focus for our use of theory to guide the later rounds of fieldwork. In their study of how systems developers used development methodologies in practice, Hansen and Kautz (2005) focus on how the practice behaves (as opposed to studying if the practice behaves in a certain way). Their use of semi-structured interviews coupled with analysis of the literature and existing documentation is similar to the approach that we have adopted in our research. Krotov and Junglas (2006) combined grounded theory and multiple case studies in their analysis of 48 published cases to determine the role of mobile technology as an enabler for organization agility. While we believe that their use of the term organization agility is highly limited (since they scope it solely to using information technologies to provide business agility), their use of mixed methods is similar to our own approach to data gathering and analysis.

Gumm's analysis of the benefits of distributed software development, specifically at the distributed requirements engineering process highlights the interdependency between benefits and challenges in that environment. Her analysis across nine case studies found that the five benefits of process definition, documentation, requirements, autonomy and working situation, were mutually dependent, and often manifested as challenges. As she puts it best, these benefits are Janus-faced (Gumm, 2006). Sakson (1997) uses a grounded theoretic approach to examine the differences between bureaucratic and clan based (cultural) controls among technical specialists who have to coordinate across organizational and disciplinary boundaries. Her analysis of twelve interviews highlighted the importance of interpersonal techniques to support coordination.

Herrmann and Daneva use grounded theory to understand what methods and activities are needed to prioritize requirements when using cost and benefit predictions as criteria. Unlike conventional approaches, they only used document analysis to develop their framework, and acknowledge the limitations of their approach both in their discussions on validity as well as in future work (Herrmann & Daneva, 2008). Ashry and Taylor (2000) link innovation diffusion theory and requirements analysis using a grounded theoretic analysis of seventeen interviews. While their findings provide preliminary linkages to the factors affecting innovation diffusion, more research is needed for creating a generalizable framework to aid

researchers and practitioners in addressing the organizational consequences of integrating systems. Qureshi, Liu and Vogel's (2005) grounded theoretic study of the impact of collaboration tools on distributed project management enabled them to create a model that explains the interactions between communication, collaboration and adaptation. Their model provides a means for project managers to anticipate and mitigate potential challenges in distributed project management. Qureshi and Noteboom's study of two sets of globally distributed projects provides insight into the extent and type of adaption needed for use of collaboration technologies in distributed projects. Their analysis of collaboration transcripts between two groups of digital natives and digital immigrants found that the digital natives had less than a third of the episodes as digital immigrants (Qureshi & Noteboom, 2005).

While there is a growing body of knowledge on applying grounded theoretic analysis to software engineering problems as evidenced in this section, it is also clear that there is significant variation in how the analyses were carried out. Depending on the school of grounded theory followed, the incorporation of theory into the findings also varied. Part of the challenge as a researcher, was to find the appropriate data analysis strategy to effectively construct a story that was both true, and had explanatory power outside of its context.

2.3.2 Our Approach to Data Analysis

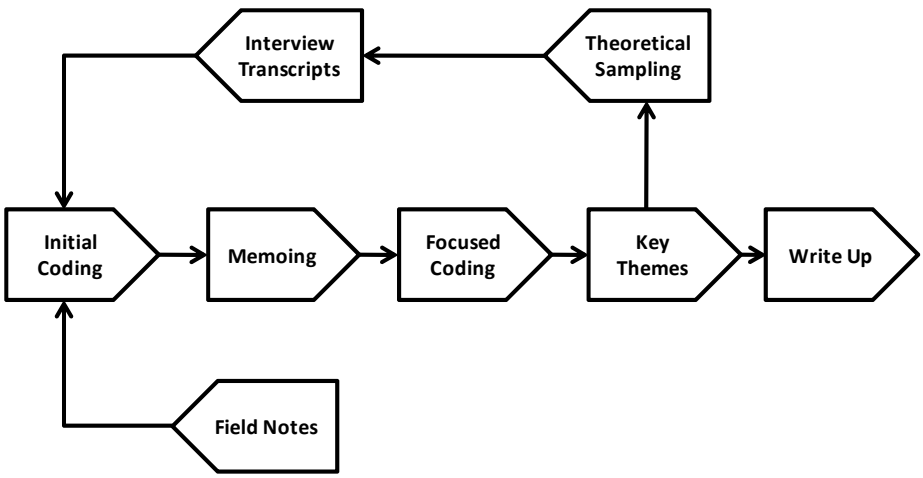


Figure 2-3 Grounded Theoretic Analysis Approach

We follow the approach specified by Charmaz (2006) for carrying out a grounded theoretic analysis as shown in Figure 2-3. The importance of using grounded theoretic, is that it emphasizes the adoption of techniques used in constructing a grounded theory, without sacrificing the need to be literature agnostic. The initial coding is carried out for each of the

interview transcripts and field notes as a means of reducing the data. The codes developed through initial coding are abstracted to tentative categories through the writing of memos. The memos allowed us to carry out focused coding and develop more advanced memos to determine the key conceptual categories/themes. The richness of the memos and the themes guided focused data collection, and the development of the final write-up.

Even prior to the carrying out of initial coding, elements of the dataset are visualized based on the counts of the occurrence of keywords. This rough pass provides some ideas on categories and themes, but cannot be assumed to be the perfect answer. Consider the example of the definition of project success by two teams at EuroTel shown in Figure 2-4.

Figure 2-4 Word Count Based Analysis of Project Success Criteria

The word-count based analysis provides a rough indicator of the challenges of quality, delivery and time in achieving project success. The next step in the analysis, is to code the transcripts to carry out a finer grained analysis. As Auerbach and Silverstein point out (2003), the central idea of coding is to move from raw text towards answering the research question under study. Initial coding provides a means of reducing the raw text to relevant text.

For example, the initial coding for the three paragraphs of text from an interview with a software designer at EuroTel resulted in 12 codes, ranging from good quality to knowledge availability. It is important to note that the same block of text can represent multiple codes. For instance, a block of text represents both knowledge availability as well as team culture. Considering that each of the 20 interviews consisted of uncoded/unrefined text ranging from 6 pages to 15 pages, it becomes important to reduce these initial codes to focused codes through writing memos. These focused codes lead to the creation of the themes that drive the write up. Memos allow the researcher to synthesize across the codes to gain a deeper understanding of the interactions between the codes. For instance, when analyzing the transcript of the interview with the software designer, we wrote multiple memos regarding

the perceptions of project success. An excerpt from one of the memos is shown in Table 2-5. These memos led to the creation of conceptual categories that included the notions of project success and project failure. This clustering of initial nodes into conceptual categories leads to the identification of themes that formed the foundations of the EuroTel case study. Analyzing the themes from the first round of fieldwork at EuroTel led to more focused interviews with project teams to provide a thicker description of how EuroTel has evolved over the last eight years, and how they used the three mechanisms of process improvement, innovation, and leveraging globally available capabilities identified in the previous chapter to gain enterprise agility.

Table 2-5 Excerpt from a Memo on Project Failure

Designer perceptions of project success and failure at EuroTel are strongly related. For instance, knowledge availability has been cited as a factor that determines both project success and project failure. When it comes to projects failing, the designer focused on the fact that the process limited decision authority. As a result, his team spent a lot of time waiting for decisions. Alternatively, when discussing the factors that made the project successful, he noted that having a well-planned project with accurate estimations enabled the team as a whole to achieve success….

When a key set of themes has been identified, or a potential pattern of behavior emerges, then more focused field work can be carried out using theoretical samples from within the organization. For instance when we analyzed the data from the first round of fieldwork at EuroTel, we found that project teams were stressed, people were risk averse, and the overall morale was low. This led to our carrying out a second round of fieldwork at EuroTel focusing on the dynamics of project teams.

2.4 Learning by Doing

One of the challenges of doing engaged scholarship is managing the expectations of the various stakeholders involved in the research process, especially in the sites at which field work was carried out. The primary stakeholder groups we engaged as part of the research process were the senior leaders, middle managers, and line employees. From an enterprise value stream perspective, the senior leaders determine the direction of the firm and the approaches that the firm adopts to gain agility; middle managers (project managers, system architects, product owners) translate this vision into action through policy directives, and day to day management activities; and line employees (developers, quality analysts, business analysts) create the value that the firm delivers.

Given that the research explored the sources of enterprise level agility, each of these stakeholder groups has a different expectation of the outcomes, as shown in Table 2-6. Senior leadership expected to get a validation of the strategic direction of the firm, and obtain

feedback with respect to where the organization currently was in its evolution. Having an independent voice that came from outside the firm boundaries provided a means of obtaining an unbiased pulse of the organization. Middle managers expected to get a deeper understanding of what was on the minds of their senior leaders, and feedback on the feelings of their employees. Line employees on the other hand, felt that the researcher provided a means of carrying their concerns to the senior leadership team, and wanted to have a safe environment in which they could voice their opinions about the firm's leadership and operational challenges without repercussions. In some cases, they felt that the opportunity to interact with the researchers allowed them to both vent, and reflect on the organization. As one interviewee put it,

> *"This feels like a session with my therapist, I actually have time to stop and think about how we are working now"*

Table 2-6 Expectations of Stakeholder Groups

Stakeholder Group	**Expectation**
Senior Leadership	• Validate the strategic direction of the firm • Provide a pulse of the organization
Middle Managers	• Provide insights into the minds of both senior leadership and line employees • Provide strategies on improving their day to day work
Line Employees	• Carry their voice to senior leadership • Provide a 'safe environment' to share thoughts, ideas and frustrations

As our research has evolved over the last five years, the practices that enabled us to be successful were:

1. Engaging leadership
2. Adding value before, during and after the research
3. Balancing the research/consulting divide
4. Asking questions
5. Validating in the field
6. Revisiting the data
7. Having fun.

The first three are about establishing the context for the research, while the latter four are about managing the execution of the research.

2.4.1 Engaging Leadership

Engaging leadership is critical to the success of the research project. In the case studies that involved fieldwork reported in the thesis, the first contact was through senior leadership in five organizations (EuroTel, BankCo, FinServicesCo, IndiaCo, and SpaceCo). In the other two organizations, the initial contact was through a middle manager (GameDevCo) and line employees (AgileCo). In these seven cases, we briefed their senior leadership team on the goals of the project, and obtained their buy-in by making clear our expectations with respect to access, and organizational resources needed. One of the keys to success in engaging leadership is in articulating the value proposition of the research. In our research, we were able to highlight the connection of the research to enabling better strategy formulation and execution. In addition to providing them with insights into how their firm operated, the research would provide them with an opportunity to understand how other firms addressed similar challenges. One of the advantages that we leveraged in our discussions with senior leadership was that the research was externally funded. We emphasized that their commitment of organizational resources was limited to in-kind contributions of their own expertise, and that of their employees. Obtaining this buy-in from the senior leadership team enabled middle managers to provide access to their teams, and provided the teams with an opportunity to step away from day to day activities and reflect on both the past and the present.

2.4.2 Adding Value Before, During, and After the Research

In addition to sharing the findings of the research, part of the value proposition was access to the researcher's existing body of knowledge. In our case, it was earlier work that we had done on both the technical side, as well as the management side. We were able to share lessons learned about successful enterprise transformation efforts at other organizations, as well as examples of best-in-class organizations from other domains such as automotive, and aerospace & defense.

Being able to discuss strategy with senior leadership, project management with middle managers, and technical issues with the line employees enabled us to add value to the conversations. In each of our interviews, we emphasized the importance of sharing knowledge, and used references from the literature and existing case studies as a means of seeding questions. For instance, when discussing the challenges of inculcating continuous process improvement with senior leadership, we used examples from Toyota and Rockwell Collins as exemplars. When discussing project management strategies with middle managers, we shared lessons learned from other organizations. With line employees, the value we added

was providing a safe environment, and actively listening to their challenges. Examples of our adding value at the various phases of the research are shown in Table 2-7.

Table 2-7 Examples of Researcher Value Added Activities

Prior to Starting	During the Research	After Completion
• Co-Teaching with team members of AgileCo • Facilitating a workshop with members of IndiaCo and EuroTel • Teaching an executive seminar at IndiaCo	• Draft Case Write Up • Summary of Lessons Learned • Discussions during and post interviews	• Final Case Write Up • Briefing the senior leadership teams at GameDevCo, AgileCo, IndiaCo, and EuroTel

2.4.3 Balancing the Research/Consulting Divide

While approaches such as action research blur the divide between research and consulting, the approach developed here blends process consultation and clinical inquiry. Our approach depends on understanding, defining, and explicating extant problems, not on solving them. Since the research is a part of a doctoral dissertation, we were able to mitigate the problem solving expectations by emphasizing that the intent was on articulating the challenges, and that solving them would be another potential research project. An effective strategy at managing expectations was involving both the thesis advisors to address questions that were out of the scope of the research.

2.4.4 Asking Questions

While 'asking questions' may sound like an oxymoron when discussing a research approach that uses interviews as one of the approaches for data gathering, the intent is to enable the researcher to go outside the box to obtain alternative perspectives. A case in point at GameDevCo was a conversation with an employee we met at the coffee bar whom we did not formally interview. The conversation was about her experience at GameDevCo, wherein she pointed out that she loved working at the GameDevCo location rather than at the corporate office. As she said,

> *"This place is so much more relaxed than the corporate office ... I am fresh out of school, so I get to learn more here"*

This provided further insights into the tensions between GameDevCo and their corporate headquarters. Asking questions can further help clarify terminology that the researcher may not be familiar with within the organizational context. For instance, AgileCo uses the term velocity as a means of assessing work accomplished in any given iteration. Since the case study was carried out at a very early stage of the research, we were not familiar with how that measure affected project success. When queried about the term, it was explained to us that the velocity metric was the basis on which they planned future iterations – it was not just a measure of work completed, but a measure that gave project mangers confidence about what could be accomplished in the future. This led to the finding of how project mangers acted as 'translators' between their development teams, and their outside clients.

2.4.5 Validating in the Field

The most effective means of enhance learning is to validate in the field itself. We established an end-of-the-day meeting with the project champion to share what we had learned, both to verify if it was, in fact, correct and to gain clarifications on issues that were still unclear. This meeting with the champion at the end of a day of interviewing also allowed us to refine both the avenues of further inquiry as well as our mental model of what had occurred in the organization. Two instances come to mind at IndiaCo:

- When we were visiting IndiaCo for the second time, one of our project champions was unable to meet us in the morning as he had been working to resolve a project crisis the previous night, and had only left the office early that morning. During the interviews, the issue of the compression of knowledge transfer cycle times from the client to the IndiaCo team had come up and, in one case, an interviewee pointed out that part of the challenge had to do with the fact that IndiaCo was transitioning from a *Services Culture* to a *Product Culture*. There was no evidence that we had seen or heard that we could use to triangulate that finding. The meeting with the champion at the end of the day, allowed us to determine that the crisis that he had been handling rose from exactly that challenge. The knowledge transfer cycle time had been compressed from 24 weeks to 12 weeks, and since the project he was managing involved a stable product, the client had not transferred the hardware to the offshore team, and the transfer team had been too busy to address the contingencies.
- Another issue raised by the IndiaCo team members was the fact that unlike their experiences with another client, EuroTel did not pay bonuses to experienced team members as a way of rewarding their tenure with the relationship. The team members felt that such a bonus would provide them further incentives to stay with the project, and would have addressed some of the human capital challenges that they had faced early on in the relationship. The senior leader of IndiaCo provided a counter point during one of the evening meetings in which he pointed out that it was his responsibility to ensure that team members wanted to stay in the relationship, and that EuroTel should not have to worry about it.

These insights would not have been possible without the clarification meetings. Even though these issues would have been raised as part of the data analysis, their importance and impact might have been lost by being outside the research context. An equally important source of validation, as discussed earlier, is the final case brief out, in which, the members of the organization provided feedback about the findings of the case study.

2.4.6 Revisiting the Data

At the heart of the success of the approach is the cycle of induction and abduction that comes from revisiting the data. In addition to the interview transcripts, the field notes and memos provide a rich source for further reflection and analysis. A case in point at AgileCo was one of the comments made by an employee who had left the organization, wherein he pointed out that the primary reason that he left AgileCo was that the culture had changed. We had captured this as part of the field notes, and on further reflection decided to ask current members of AgileCo about whether or not the culture had in fact changed/evolved. As one of the interviewees said,

> *"Right now, we have a lot more 'management' – before the iteration manager role was a rotational one, and we did it in addition to our regular roles. Now we have people dedicated to doing that – while I can understand why some teams need it, in our team it adds no value, as the iteration manager has very limited technical skills"*

Similarly, at EuroTel, our initial interviews and data analysis led us to the conclusion that there was in fact a risk-averse culture, but had no means of triangulating that finding. We structured a second round of interviews, focusing at the project team level to determine how teams perceived project success. It was due to this second round, that we found evidence to support that viewpoint, as EuroTel teams ranked meeting schedule requirements as being more important than either quality or meeting the feature set requirements.

2.4.7 Having Fun!

One of the most challenging aspects of doing field research is the stress and anxiety in setting up the visits, gathering useful data, and analyzing the data away from the field. This requires a level of dedication and focus that can sometimes rob the joy of doing the research. We learned through experience that by having fun in each of steps of doing the research from the data gathering to the hours of analysis, the research process provides greater insights (and a better quality of life).

2.5 Ensuring Trustworthiness of the Findings

The traditional notion of believability of the findings, especially in positivist research, is found in the notions of construct validity, internal validity, external validity and reliability. From a qualitative validity perspective, we use extensive quotations, triangulation with other sources, anchoring in the literature and member checks. We first address the conventional

notions of validity and reliability, and then focus on the four aspects of trustworthiness: credibility, transferability, dependability and confirmability.

2.5.1 Conventional Validity and Reliability

Construct validity focuses on whether the theoretical constructs are interpreted and measured correctly. In the research presented here, the questions were designed to identify phenomena that either have a theoretical basis in the published literature or had been observed in practice. There are three threats to construct validity that we explicitly addressed: mono-operation bias, mono-method bias, and confounding levels of the constructs. The sampling strategy ensures that there are a minimum of two cases of each of the three mechanisms. Furthermore, the data collection approaches provided a means of eliminating any bias introduced during data collection. From a participant's perspective, we avoided both hypothesis guessing and evaluation apprehension by openly discussing the problem we were interested in, and by ensuring the anonymity of the participants. The research protocol served to support the researcher's role as being one of collaborative inquiry, rather than as an experiment.

Internal validity focuses on the study design, specifically on whether the results really do follow from the data. In the thesis, we established a clear protocol for gathering, storing, and managing data. The use of qualitative data analysis software supported our data management strategy. The use of intermediate reports and multiple cycles of fieldwork ensured that there was traceability between the analysis findings, and the raw data gathered.

External validity focuses on whether the claims for the generality of the results are in fact justified. Given the exploratory nature of the thesis, and the constructivist-bias of the researcher, the claims made in the individual cases are local theories that are drawn from the data. Similarly, the generalizations made through the cross-case analysis were supported using theoretical triangulation.

Reliability focuses on whether the study yields the same results if other researchers replicated it. Given the intensive nature of the field-work, and the relationships built as part of the fieldwork, the finer details may not be the same, but the findings would be broadly consistent, if another researcher were studying the phenomena at the same time.

2.5.2 Trustworthiness/Qualitative Validity

From the perspective of ensuring qualitative validity, Creswell (2002) suggests the use of

a. Triangulation: the use of different sources of data to confirm results and build a consistent story

b. Member checking: going back to research participants to ensure that the interpretations of the data make sense from their perspective.
c. Rich, thick descriptions: using detailed descriptions to convey the setting and findings of the research.
d. Articulating bias: being honest with respect to the biases brought by the researchers to the study and use this self-reflection when reporting findings.
e. Report discrepant information: when reporting findings, report not only those results which confirm the emerging theory, but also those which appear to present different perspectives on the findings.
f. Prolonged contact with participants: Make sure that exposure to the subject population is long enough to ensure a reasonable understanding of the issues and phenomenon under study.
g. Peer debriefing: Using a peer debriefer who can ask questions about the study and the assumptions present in the reporting of it, so that the final account is as valid as possible.
h. External auditor: Finding an external auditor to review the research procedure and findings.

These eight techniques ensure the credibility, transferability, dependability, and confirmability of the findings.

Credibility focuses on establishing that the results of the study are credible to the participants in the study. Since the expectation of the research is that it has captured the collective experience of the people interviewed, then the people interviewed should agree with the findings. In this thesis, we repeatedly synchronized with key stakeholders in the organizations studied to ensure that the findings were an accurate reflection of what was seen by the researcher, and understood to have been the experience of the participants. In all of the in-depth case studies, we were able to carry out multiple rounds of fieldwork, providing us the opportunity to ensure that we had accurately understood and described the phenomena.

Triangulation is a common strategy used to increase the believability of the findings. Of the three commonly used triangulation approaches: data triangulation, method triangulation, and analyst triangulation, we have made extensive use of the first two, and limited use of the last. Our data was gathered from multiple sources, using multiple methods. For instance, in the case of EuroTel, we used group workshops, individual interviews, observations, archival data, and. Given that the research was carried out by a single investigator, it was difficult to obtain analyst triangulation though in the case of EuroTel and GameDevCo, we were able to draw upon peer experts to review the findings. A sterner test was through the external reviewers of the papers that we have published.

Transferability refers to the degree to which the results can be generalized/ transferred to other contexts and settings. The case studies were selected based on the fact that they represented polar cases with respect to the value creation strategy, and the development process. Furthermore, the comparative cases turned out to be polar cases with respect to

success. The findings from the cases support transferability, but further research is needed to develop a truly general theory of enterprise agility in software organizations.

Dependability refers to the ability of the researcher to capture the ever changing context within which the research is carried out. In writing the case studies we have made focused efforts to bound the analysis and findings to the specific context within which they occur. The only instance where we do not account for the context is in the selection of the four organizational enablers. While there may be other organizational enablers for obtaining enterprise agility, the hypothesis derived from the research is that the four enablers of stakeholder alignment, employee empowerment, governance, and learning, together cover all the requisite elements from an organizational perspective.

Every researcher brings their own biases to the analysis of qualitative data. We have tried to make most of the biases and assumptions explicit in the thesis, and have further enhanced the confirmability using multiple cycles of member checking and peer review. Consider the case of IndiaCo, where we carried out a significant amount of fieldwork. When the first version of the case was written up, one of the peer reviewers pointed out that it was projected as a 'perfect' organization, with words such as world-class, best-in-class being used frequently. Identifying this bias enabled us to be more reflective both with respect to asking questions, as well as carrying out the analysis. The complete mapping of strategies to the case studies is shown in Table 2-8.

Table 2-8 Techniques Used to Increase Trustworthiness

	AgileCo	**GameDevCo**	**IndiaCo**	**EuroTel**	**Rockwell Collins**
Triangulation	X	X	X	X	X
Member Checking	X	X	X	X	
Rich Descriptions	X	X	X	X	X
Articulating Bias		X	X	X	
Discrepant Information	X	X	X	X	
Prolonged Contact	X	X	X	X	
Peer Debrief	X	X	X	X	X
External Auditor	X	X	X	X	X

2.6 Research Strategy

The approach that we believe works effectively for carrying out firm level analysis is shown in Figure 2-5. The stages of engagement, preliminary study/exploration, subsequent field visits that are focused on specific areas, and knowledge sharing form the skeleton around which the researcher can leverage specific data gathering and analysis tools. Our experience has shown that grounded-theoretic approaches combined with cross-case analysis provided the most useful insights.

Figure 2-5 Research Approach Developed

The engagement phase focuses on articulating the researcher's assumptions about the problem, and having those assumptions be assessed by the stakeholders involved in the project. In this phase, it is also incumbent upon the researcher to surface the assumptions of the participants to establish a shared understanding of the problem being studied. This engagement phase is often the most difficult phase, as the possibilities of success are often limited. The exploration phase requires the researcher to cast a wide net in gathering information, and translating that information either into knowledge about the organization under study or into areas that require further exploration through focused field work. Given that our work focuses on people and organizations, our data collection mechanisms were through interviews and analysis archival documentation. In the case where we studied large groups, mechanisms such as blogs, twitter, and source code repositories provided useful insights (Gagne and Srinivasan, 2009). The focused field work mechanisms are deep dives into specific phenomena based on findings of the engagement and exploration phases. It is often in this phase that puzzling, often contradictory phenomena are found. The knowledge sharing phase focuses on disseminating the findings both with the organization under study and with the community at large.

Areas this approach does not work, are when a researcher is attempting to assess a very narrow, or a very broad phenomenon. In the former case, a well designed experiment would provide more effective insights will less significant workloads, and in the latter, techniques such as surveys would be more effective, as it would be impossible for the researcher to engage in the field.

3 CASES IN ADOPTING AGILE METHODS

3.1 Introduction

The software development approaches that constitute 'agile methods' have forced a paradigm shift in the mechanics and management of software development. We use the word paradigm shift in the same vein as (Kuhn, 1970) because agile methods have resulted in what is essentially an epochal change. The four values and twelve principles espoused in the agile manifesto that was published in 2001 (Beck et al., 2001), challenge the conventional notions of how software should be developed and managed. The last eight years have seen increased adoption by practitioners on the use of these methods, and reported benefits in terms of increased employee morale (Cockburn & Highsmith, 2001), productivity (Cohn & Ford, 2003) and customer satisfaction (Williams & Cockburn, 2003). The academic literature in the area is burgeoning, and most published findings have been written by practitioners (Abrahamsson, Salo, Ronkainen & Warsta, 2002). Our goal in this chapter is to contribute to the literature on agile methods by answering the question of

How do software organizations adopt and sustain agile methods?

We chose to tackle the question by focusing on two pilot case studies in India, followed by two focused case studies on agile adoption and sustainment. The two pilot case studies of BankCo and FinServicesCo were representative of organizations embarking on their agile journey. AgileCo on the other hand, had been using agile methods since its inception in 2001, making it the ideal case for answering the second half of the question on sustaining the use of agile methods. The three cases were all in the software services sector, leading us to revisit the question of using agile methods in a product oriented context as our second in-depth case study of GameDevCo in 2007. The complete chronology of the case studies is shown in Figure 3-1.

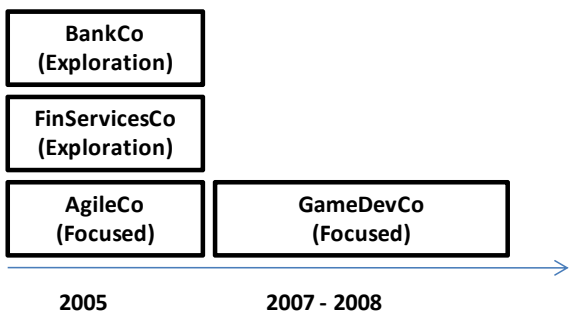


Figure 3-1Case Chronology in Agile Adoption

In the remainder of this chapter, we cover the case studies individually, followed by a comparison of the focused case studies to identify similarities and differences.

3.2 Understanding Scrum

While there are a number of methods that fall into the category of agile methods, we focus on Scrum. The notion of using cross functional teams in software development can be traced back to (Takeuchi & Nonaka, 1986), on applying ideas from the game of rugby to product development. The concept was formalized for the software engineering community by Ken Schwaber in 1995 (Schwaber, 1996). Since its formalization, Scrum has become more widely adopted, and as Marchenko and Abrahamsson (2008) note is set to become the de-facto standard in industry. Scrum is described by the Scrum Alliance as an agile software development framework with three roles, three ceremonies and three artifacts. This can be visualized as three layers showing the ceremonies, the process and the roles, as shown in Figure 3.7.

The three roles of Product Owner (PO), Scrum Master (SM), and the self-organized Scrum Team (ST), have distinct yet coupled responsibilities as summarized in Table 1. The product owner is responsible for the business value of the project. He or she creates a list of requirements (often called user stories) that are prioritized based on business value (deliver value to one or more stakeholders, often the customer). This list is further augmented with preliminary cost estimates developed by the scrum team, to create the product backlog. In preparation for the actual execution of the project, the PO presents the overarching roadmap and the release plan in the Sprint Planning Meeting. The ST in conjunction with the SM pulls features from product backlog, and refines them into the actionable tasks that populate the scrum backlog. The actual execution of the project is carried out using time blocks called sprints that last between two and four weeks. The duration of the time block is dependent on the nature of the project, the people in the project, and the organization policies with respect to project management.

During the sprint, the SM leads the team in the daily scrum meeting to create a shared understanding of the state of project. Every team member shares what they did in the previous day, what they plan to do today, and what challenges they faced. This shared situational awareness enables the team to surface and manage new dependencies, and equally important, enables the scrum master to identify and correct any real-time impediments to the team. One of the features of most agile teams is their use of a ‘project wall’ to enhance situational awareness – during the scrum meeting, the team members often physically move tasks (written on cue cards or post-it notes) between the sprint backlog, the ‘in-work’ section, and ‘completed’ sections of their project wall. The scrum master also has the responsibility of maintaining the burn-down chart – the amount of work that has actually been accomplished to date in the sprint. This chart is refreshed daily, and acts as an additional information radiator for the team to see progress. At the end of the sprint, the scrum team demonstrates the developed software to the product owner, who assesses the effectiveness of the sprint and

determines the necessary reprioritization of the product backlog to create the goal of the next sprint. The first half of the sprint review meeting also serves as an opportunity to review the big-picture effects of the product, including the market, technology, and business impacts. The second half of the meeting is led by the scrum master and is devoted to the retrospective. The retrospective provides the scrum team with the time needed to reflect on the process, identify improvement opportunities, and create strategies for closing the gap. The process starts over with the planning meeting for the next sprint. The simplicity of the framework masks the difficulty of making it actionable, as seen in the case of GameDevCo.

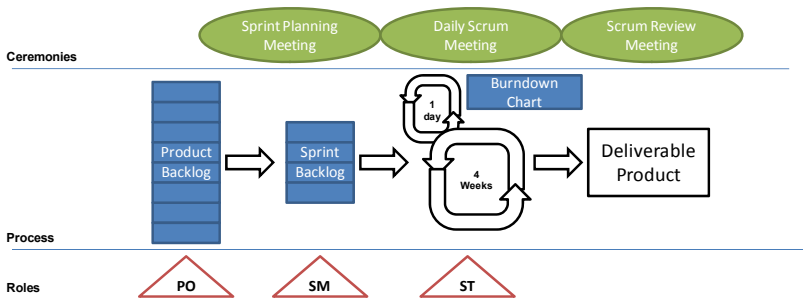


Figure 3-2 Scrum Overview

Table 3-1 Scrum Roles and Responsibilities

Scrum Role	Responsibilities
Product Owner	Define product features and roadmap Prioritize user stories based on expected business value Manage the product backlog by adjusting and reprioritizing Accept the product created by the self-organizing team
Scrum Master	Ensure that the team is functional and productive Shield the team from external interferences Coordinate meeting activities and carry out project management activities
Self Organizing Team	Does the actual work Supports the effort estimation of the product backlog Creates a shared sprint goal and selects an executable sprint backlog Pulls work to create the sprint backlog

3.3 Agile In India

In 2005, we conducted two pilot case studies and one in-depth case study to better understand how agile methods were being adopted and sustained in the Indian context. At the time when the studies were conducted, there were very few firms in India that were using agile methods, making it the ideal environment for studying how software organizations adopted agile methods. At the same time, our in-depth case study of AgileCo represents an Indian organization that was 'born agile', and had been successful at sustaining the use of agile methods for over four years. As seen in Table 3-2 these firms were all in the broad area of software services, and varied with respect to organization size and history of agile adoption.

Table 3-2 Overview of Agile Case Studies in India

	Length of Agile Adoption	Domain	Size	# Interviews
FinServicesCo	Initiated a pilot study in 2005	Financial Services	> 1000	5
BankCo	Long global history, initiated local adoption in 2005	In-house financial software services	< 100	7
AgileCo	Since inception in 2001	Software Services	< 100	12

In 2005, the agile software movement was nascent in India, providing the perfect research setting for studying how firms adopt agile methods. Martin Fowler (Fowler, 2007) wrote about the experience of running offshore projects in India and Australia, wherein he noted the importance of retaining their high standards in selecting employees (offering jobs to 1 in 200 applicants), maintaining a mix of new hires and seasoned employees, and mentoring developers in the new office on the use of agile methods. Of the 11 lessons learned, two are worth mentioning here: *don't underestimate the culture change* and *expect to need more documentation*. One of the fundamental challenges of transitioning to an agile environment is the increased autonomy that it provides to the software development team, and the resultant flattening of the decision hierarchy. In addition to the software process improvement literature, we explored the literature in the areas of distributed software development, software contracting, and outsourcing, but found little that provided detailed descriptions of the challenges of adopting agile methods in the Indian context. The exceptions included papers that present different views of the problem using agile methods in a globally distributed development context such as those by Summers (2008), Sureshchandra and Shrinivasavadhani (2008a, 2008b), and Shrinivasavadhani and Panicker (2008).

As Summers points out (2008), the main reasons for seeking a services partner in India were flexible resources, skills available, cost savings, and professionalism. Their 'over the wall' approach in which requirements were developed at the customer site and then handed over to the business analysts in India to flesh out and implement, failed to demonstrate success, however, transitioning to an agile approach provided greater success. The four key barriers

that he identified were: culture, communication, working practices, and single vision. Sureshchandra and Shrinivasavadhani (2008a) note that Wipro's three year journey in adopting agile methods began to proactively understand, focus and deliver to customer needs. Their experiences in 90 projects, mentored through a central team of certified scrum masters has enabled them to create a model to transition from collocated agile teams to distributed agile teams. The eight best practices they have identified include: partitioning stories based on functionality and story-wise collocation of teams; using design structure matrices to understand functional dependencies; rotation of personnel between on-shore and offshore to distribute business knowledge; creating a shared understanding between customers and offshore teams through site visits; dedicated video conference rooms; technical scrums to prevent speculative coding; visual controls on all sites to create shared understanding and promote collective ownership; photo chart of the entire team.

While the literature provided some insights into successful practices, it was important to understand the context within which agile methods were being adopted and sustained in India.

3.3.1 FinServicesCo

FinServicesCo is large financial software services provider, with over 2000 employees. It was originally started as an accounting services firm that expanded to incorporate financial services. It was not till the late 1990's that FinServicesCo started developing software, and grew explosively by competing on cost differentials and deep domain knowledge. In 2005, they were considered to have highly mature processes (having been assessed as a CMMI level 5 organization) that complied to the rigid process standards that were imposed on them by the financial services industry. Their heavily plan-driven approach had enabled them to meet the rigorous service level agreements with their clients. We met with the CEO of FinServicesCo, their head of business development, and the three project managers who were trying to pilot agile methods, to understand their motivation in trying to adopt agile methods.

In our interview with the CEO, he touched upon organizational values, organizational culture, and the challenges of rapid growth. He shared with us, the outcome of a leadership off-site that they had conducted just prior to our interview:

> *"As an organization, we value, entrepreneurship, people-centered, and customer-centered. The last two are obvious...but being entrepreneurial means being able to think of the problem and the solution simultaneously, it's about taking ownership of a solution, and being restless and scrappy to get better."*

When discussing the role of organization culture, he pointed out:

> *"An organization over a period of time builds a personality and attracts people with a similar personality - at first instance, the name that you have as an organization has*

an impact, and the people who have worked with you for 5-10-15 year's help built it. Generally, the system repels people that don't get it"

When discussing the issue of growth, he pointed out:

"Managing growth in the current environment is a challenge ... we have added 60% new people to this organization, and most come with baggage ... how do you get them to think differently.. I spend 2 hours with a new class ... I get each of them to ask a question ... and use that to talk about the theme ... things that will take you far in the organization and things that don't get appreciated"

From the perspective of the project managers, agile methods were a means of achieving alignment to the values and objectives defined by their senior leadership. As one of the project managers put it:

"We wanted to align people to the overarching organization goals of growth that our CEO had set for the whole organization."

Their approach to achieving that alignment was to define, what they called 'push-pull' aspects for each of the five roles within a software development team: developer, senior developer, team lead, project leader, and project manager. The intent of defining of these aspects was articulated through an example as:

"For example, what a developer could push for in terms of escalations, and what he or she could pull in terms of what they needed to get started on something."

Since they are at the starting point of their agile adoption journey, the notions of team building and mentoring were just that – notions. As an organization, FinServicesCo had shared values that they expected from each team member, and their emphasis on agile methods had placed an increased focus on organizational control and transparency. The emphasis on flattening the hierarchy, increasing decision transparency, information sharing, putting aside personal egos, and building team spirit, is directed towards building a culture that would support the adoption and use of agile methods. In addition to senior (permanent staff) members mentoring junior and contract employees, peer-to-peer mentoring was expected to increase agile awareness and adoption.

The primary motivation for piloting agile methods was to increase project team alignment to organizational values, and to increase developer productivity. The importance of the organizational aspects of clear role definitions, shared values, and senior leadership vision, were investigated further in our focused case studies.

3.3.2 BankCo

BankCo is a new organization that was formed in early 2005 in India to support the internal software needs of their global parent organization which specializes in the banking sector. While some maintenance operations had been previously off-shored to a team in India prior to the formation of BankCo, agile methods were not used by the legacy teams, and they had not developed any new products. Our interviews included a vice president of the parent organization, the CEO of BankCo, their agile coach, a senior project manager, and three developers. Of the three developers, the first was a contractor with little agile experience, the second had recently joined after two years at AgileCo, and the third developer had joined after a short stay at FinServicesCo. As with FinServicesCo, we present the senior leadership perspective first, and then dive deeper into the perspectives of other members of the organization.

BankCo's CEO had a clear vision for where he wanted the organization to be. His vision for growing the business was articulated as:

> *"We will do the exact opposite of building a software factory – we will hire talented people who are equal to their counterparts in London and Paris – we will build quality products using talented people, and grow the business by moving projects onshore and taking larger chunks of existing projects – there will be a natural convergence of projects ending up here."*

This view was echoed by the VP of their parent organization, who talked about the cost differentials of operating in India, and the larger talent base that was available in India. The current approach of revenue generation by BankCo is to charge the parent organization on a unit-cost-per-person basis, such that they can invest in local projects and still make the books balance. When asked about the expected size of the organization, BankCo CEO said:

> *"I expect to have about 100 people here, anything more will be a bonus and any less and the attrition rates would be too high."*

Their intent in adopting agile methods was to rapidly demonstrate value to their parent organization, while gaining maximum utility from the capabilities of their talent base. Reflecting the vision set forth by their CEO, BankCo has been extremely selective in who they recruited into the organization. While BankCo is looking to hire across a broad spectrum of experience levels, their recruitment strategy is seen to:

- focus on hiring people that have not peaked in their careers,
- and on hiring people who have the ability to think independently.

Given their need to grow to meet their size requirements, they have used a strategy of hiring contractors to take on junior developmental roles while their permanent staff takes on the

senior roles of program managers, business analysts and senior developmental roles. As one of the contract personnel noted:

> *"This is so different from my own organization which is CMMI level 5, I have more autonomy here than with my own organization, and we (BankCo) actually get better quality products delivered faster. In my organization, we have a system that is based on performance and tenure – people get promoted just because they have been in the organization long enough – here it is purely a meritocracy."*

In addition to looking for strong talent, BankCo also focused on 'fit' with BankCo's culture. BankCo recognizes that their team has to operate in close conjunction to their European counterparts – being able to communicate their expectations is extremely important. Given that their parent company's culture is to manage by exception, BankCo's CEO has placed a significant emphasis on employee empowerment and open bi-directional communication across the entire organization. The emphasis on recruiting people that can spot and deliver value has led to them rejecting a large number of applications.

BankCo currently has Greenfield teams that use agile methods as their core development methodology as well as teams involved in sustainment that operate using a traditional plan driven approach. Although one would expect a culture clash between the agile and non-agile teams, they have successfully avoided the problem by having the same project manager for both sets of teams. This project manager served as the bridge between the teams, infusing the open communications and flattening the hierarchical structure in the plan-based development teams, and bringing in greater documentation discipline to the agile teams. As the project manager noted:

> *"When I first got here, it was as if these teams existed in separate worlds – it didn't help that the team doing agile work was physically located in a different building than the rest of the teams, but the issue was more of product support versus product ownership. Now we have greater interactions across these teams, and are migrating best practices between them."*

One of the greatest strengths of the agile team was the presence of an agile coach to train the team in the use of agile methods. As one developer noted:

> *"Working with the agile coach was brilliant – brilliant – brilliant – when I came to BankCo earlier this year, he was very different from anyone I had ever met before – he was calm professional and very passionate about work. We were unaware of agile, refactoring, test driven development. He sat with each of us, showed us how to do things. When we didn't have a wall for the story board, he came up with the idea of using the mobile white board that you see here."*

From a mentoring standpoint, the CEO noted that with the exception of the coach, he had really not put anything else in place. His expectation was that the teams themselves would

self organize and that the senior members of the team would serve as mentors to the less experienced members, and that the open bi-directional communication, coupled with a flat organization structure would enable peer-to-peer interactions as well.

When talking about the impact of the coach on mentoring, the contract developer noted:

> *"I am actually getting feedback about myself for the first time in my career (this comes from a person with 8 years of experience) – the right feedback that was required, and more importantly, he made me think about what other directions I needed to think about. He encouraged the process of thinking as opposed to telling me what to do. The feedback was not always positive, but it was put in a constructive manner. He also shared a lot of stories about his experiences when his projects were cancelled, how he felt, the possible risks of the project we were currently doing – little things on how to make myself better."*

One of BankCo's challenges is in building the high performance teams needed to grow their business. The legacy teams had been working on sustainment operations, and their primary role was to carry out bug fixes on someone else's code base. They were not doing any new development on their own. The establishment of BankCo's agile teams to carry out new product development did result in a change in culture within BankCo's development unit as a whole. As one developer noted:

> *"The biggest difference in working in an agile environment is the collective ownership and team ownership –you feel that your project is your own as opposed to your team or project leader's. We fundamentally changed how we worked, all of sudden, we were creating documentation, feeling enthusiastic about coding – doing the stuff we were actually supposed to be doing, as opposed to doing things to make the quality department happy."*

The agile teams have a clear vision for what they want to be, as articulated by their agile coach:

> *"Be a totally amazing software team that revolutionizes the way software is built. More importantly this team has to deliver high quality software, as they are the ones that are going to establish the trust between BankCo and the parent company. The team has to have technically gifted people that can get the job done!"*

BankCo's teams have both technical and business competencies. Their CEO emphasized the need for every team member to a clear understanding of where they fit in the organization and how their role would evolve. This was fundamental both to growth as well to prevent long-term attrition. As one of the developers pointed out:

> *"I am more in the vein of my CEO, I am more focused on the business side of things – I can execute technically, but in the long run my focus is on value identification. XYZ*

on the other hand, takes after our agile coach, he is very technically focused and excels in the value delivery."

In 2005, BankCo was still in the early stages of its agile adoption. The clear vision articulated by their CEO, coupled with strong project management and coaching has considerably eased their transition to using agile methods. Their emphasis on mentoring and adherence to the agile principles within their development approaches has enabled them to incorporate the philosophy more easily into their organization. As their CEO noted, they are currently in the middle of a change, and only time will tell if the change was successful.

3.3.3 Discussion

The FinServicesCo pilot study brought to the forefront the importance of senior leadership vision in supporting the adoption of agile methods, and the practical impact that it has with respect to fundamentally changing the way the organization works. The approach of clearly defining the roles and responsibilities of five key positions in the organization enabled them to enhance the processes used to do work, for example, in clearly specifying when to escalate a problem, and what resources he/she could draw upon if needed. In the case of BankCo, we learnt some of the key best practices that are transferrable to other organizations. Examples include the recruitment strategy of hiring personnel who had not peaked in their careers and were independent thinkers, importance of the agile coach, clarity of senior leadership vision, and managing the culture clash between agile and non-agile teams.

Both the case studies provided valuable insights that fed into the design and analysis of our in-depth case studies. The five aspects that we explored in the in-depth case studies are:

1. Importance of senior leadership vision;
2. Emphasis on culture change;
3. Criticality of the human resource strategy;
4. Role of mentors and coaches;
5. Challenges of managing growth (especially with contract personnel).

3.4 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 expected 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 2001-2009 timeframe 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. We focus the discussion of AgileCo along the four key aspects of *personnel selection and training*, *building strong teams*, *managing customer expectations*, and *teaching/mentoring*. These four aspects emerged from the analysis of the interview data as key elements that made AgileCo successful.

3.4.1 Personnel Selection and Training

AgileCo was highly selective in picking people for their organization. In addition to technical excellence, they focused on ability to adapt to the organizational culture as well as growth potential within the organization. As one senior manager noted:

> *"The selection process we have put into place makes sure that we don't get any duds – we look at coding and aptitude, and at least two interviews. As far as growth within the organization goes, intelligence, communication and technical skills are the foundation, how fast you grow however, is a function of attitude and willingness to find benefits for the customer."*

Personnel selection at AgileCo can be broadly divided into three categories, new hires (straight out of school), experienced technical personnel, and experienced managers. The details of the hiring process for each of these classes are different as the role expectations are different, however, the overarching structure is similar to the process shown in Figure 3-3.

Figure 3-3 AgileCo Selection Process

New hires are selected straight from school through both a college recruitment program, as well as through open calls. In addition to assessing their analytical capabilities, the foundational technical skills such as data structures, algorithms, and basic programming are judged. To account for a significant variation in process understanding, every one of their hires straight out of college is put through a rigorous training program that teaches them the standard processes that are used within the organization. In addition to class room lectures, the new recruits work in teams to create software solutions to problems that have already

been solved for actual customers by the organization. This exposes them to both the mechanics of the process, as well as gives them increased technical skills. Given that India forms their largest recruitment center for people straight out of school, AgileCo's parent company has centered their training program for all of their new recruits in India. As their director for training noted:

> *"We expect to do a lot more work in India, so it's good for the young people recruited outside India to come experience Indian culture, learn the process from expert teachers, and build their own learning in a 'safe environment'."*

On a lighter note, it was mentioned by multiple people that the training program initially was referred to as a 'boot camp', but that created issues with respect to getting visa's for the global new hires, leading to a renaming of the training program to 'AgileCo University'. When talking to some of the global new hires that were present at the time of the study, one of them pointed:

> *"The great part about coming to India, besides all the great training is that we can get handmade suits that are tailored – we cannot get that at this price point back home."*

Every person interviewing for an experienced technical position at AgileCo takes both an aptitude test, as well as a coding test. The aptitude test is a multiple-choice questionnaire designed to assess the individual's analytical and general problem solving capabilities, similar to the analytical section of a general aptitude test like the GRE (Graduate Record Examination). Passing this test is a prerequisite to taking the coding test. This filter was established as the coding test takes a significantly greater investment of the organizations technical resources.

The coding test is designed to test the person's fundamental knowledge in the software development lifecycle, covering aspects of requirements understanding, design specifications, code development and bug fixing. Successfully passing this test is a prerequisite, especially for hires in the developer or business analyst categories, as they work closest to the delivered product. As part of the test, the job candidate is expected to develop a solution to a client-specified problem, usually drawn from actual projects that were executed by AgileCo, and which involve ambiguous requirements with incomplete information with respect to both the problem context as well as client expectations. The effectiveness of the solution, as well as the approach adopted by the candidate is critical to passing the test. Observations of how the job applicant addressed the gaps in requirements, and elicited needed information are captured by members of AgileCo and are shared with other team members who perform the technical interviews. Once a candidate passes the coding test, they attend a technical interview that is conducted by three members of AgileCo (either individually or collectively, depending on project constraints and availability). In this stage, the candidate is expected to explain the solution that they developed as part of the coding test, and further explicate their rationale for the key decisions that they made. The discussion of the data structures used and

algorithms developed allows team members to determine whether or not the candidate has the technical foundation needed to be successful in the organization. It is also here, that in-depth knowledge of the individual in their chosen technologies is probed. For instance, a candidate coming in with Java on their resume is probed deeply on the underlying class structures, as well as in how the code actually runs on a given target platform - aspects of how the complier functions, and the optimizations it introduces are default questions. When discussing the importance of the coding test and technical interview, one of the members of AgileCo noted:

> *"A lot of people work with the resume-driven programming mindset – they work on a project for three to six months just so that they can list a technology on their resume – the coding test, and the technical interview really brings out the depth of knowledge that the person has."*

A candidate that has been successful at the technical interview then goes through a HR interview. At this interview, AgileCo team members partner with members from their human resources department to assess the cultural fit of the team member in the organization. AgileCo's flat organization structure and focus on open and honest dialogue requires the willingness of team members to be active listeners, and to offer constructive critique to ideas that are offered. Additionally, their use of first names in the office, irrespective of organizational rank, can be difficult to people coming from a very hierarchical organization.

This was further illustrated when we were teaching the XP Planning Game at a conference with members of AgileCo. In one of the tables, during the retrospective after a round of the planning game, the table members were unsatisfied with the performance of one of their team members, and were vocal about their opinion. It turned out that the person that they were not happy with was in fact a project leader at an organization with a very hierarchical structure. His statement at the coffee break was:

> *"We would have won the game if they had only listened to me. I am a project leader with seven years of experience, and I have never had a team like this"*

To which the table team members responded:

> *"He may have the experience but he doesn't understand this process, and we were better off after we completely tuned him out."*

As the facilitators reflected on the session at the end of the day, an AgileCo team member noted:

> *"That guy is a classic example of someone we would never hire, he just doesn't listen to anything but his own voice – we expect team members, especially leaders to listen carefully… this is something we try to catch at the interview phase because you cannot survive in our organization expecting to use command-and-control behavior."*

Another strategy that AgileCo effectively employs to understand the fit of senior personnel to the organization, is to use junior members as part of the interview team. This provides the job candidates with an exposure to AgileCo's culture, and provides the team with an opportunity to assess whether or not he/she would be able to work effectively in their environment. When it comes to hiring managers, AgileCo's emphasis is on assessing the individual's ability to mentor, motivate, and manage their teams. Given that a significant portion of their technical team is under the age of 25, it becomes critical that managers have a deep understanding of the processes used by AgileCo, and can effectively coach their team members.

3.4.2 Building Strong Teams

AgileCo's primary units of organization are the project and the team. Each project consists of multiple teams that work together to solve a specific customer problem. Each of these projects is headed by a project manager, who acts as the primary interface between customer-side management and the teams, as shown in Figure 3-4. Each team consists of developers (Dev), business analysts (BA), quality analysts (QA), team leaders, and the on-site customer. In the cases where an on-site customer is not available, one of the team members acts as a proxy, to ensure that the voice-of-the-customer is heard by the team. Developers are responsible for the core technical content of the developed software. In addition to developing the code for their respective modules, they also have to develop the unit test cases to show that their code is operating as expected. Quality analysts are responsible for the integration test cases, and acceptance test cases. They work closely with both the developers, and testers to ensure that the product delivered to the customer works as expected, and more importantly satisfies customer requirements.

Business analysts are primarily responsible for eliciting customer requirements, and analyzing them to determine whether or not those requirements need to be implemented in a given release cycle. The business analysts work closely with their customers, as well as developers and quality analysts in ensuring consistent value delivery to the customer. Of the five team roles, the one that is least clear is that of the team leader – in some teams, the team leader was a rotational position which included team management responsibilities, in addition to one of the other roles (BA/QA/Dev), while in others, it was a dedicated role. This nature and importance of this role was also a polarizing issue within AgileCo since it fostered a difference in understanding with respect to how teams functioned, as well as a significant variation in the technical knowledge expected in the team leaders.

These strong and integrated teams are one of the strengths of AgileCo. The overall process of project execution (shown in Figure 3-5), leverages the strengths of various agile practices to create a hybrid approach that is best suited for their organization. Once a project is received from a customer, an iteration planning meeting is held in which the project manager and business analysts present the goals for the project, and the expected features of the end product in the form of stories. The team then determines the iteration goals, and estimates the stories that can be executed within the allotted 4 week time span. This set of stories is then

transcribed in the form of post-it notes and posted on the team's story wall, which captures the current state of the project.

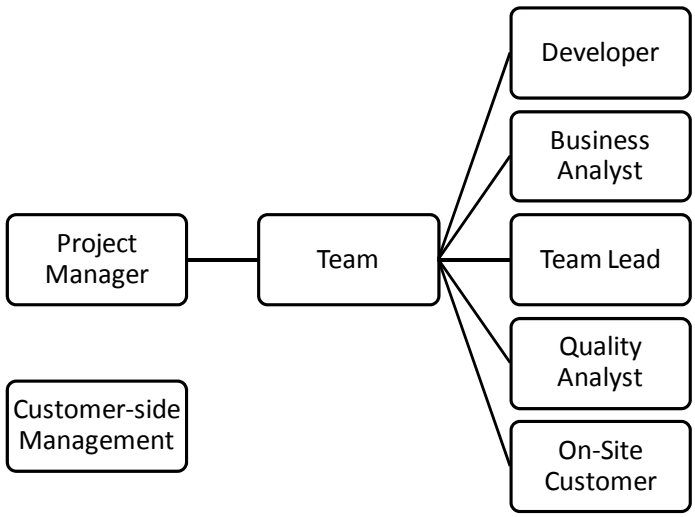


Figure 3-4 Team Structure at AgileCo

Each story is either in the queue, in development, or in integration & testing. In the daily standup meetings, the entire team gathers around the wall to assess the current state, and pick assignments for the day. This ensures that everyone on the team has the same shared mental model with respect to status at the individual, team and project levels. As the team picks their stories to work on, they also select the pairs within which they will work in for that given day. Once a development pair checks in the code they are essentially taking responsibility for the quality of the delivered product. The teams at AgileCo have automated the build and test process which is triggered nightly – teams are not allowed to proceed onto other stories until the system has successfully passed its integration test. At the end of each iteration, the teams hold an iteration retrospective to assess the strengths and weakness of each iteration, and capture the lessons learned both for future improvements, as well as for new technical insights.

AgileCo's adoption of the practices of iteration planning meetings, standup meetings, pair-programming, collective code ownership, and information radiators ensures that all the members of their team have a clear understanding of the current state of the project, and can identify roadblocks to progress. Two practices in particular are worth exploring in greater detail - the use of 'pair-programming' and 'spikes', as it addresses the balance between team collaboration and individual exploration.

Figure 3-5 Project Execution Process at AgileCo

Pair-programming is the principle mechanism that the project teams use to self-organize. Pairs are formed at each stand-up meeting, and self selected by the team members based on the story-cards that they are working on. While project managers track pairings from the standpoint of ensuring the growth of junior team members, they do not directly intervene in the actual pair formation itself, rather choosing to have a discrete word with senior members about mentoring. As one developer noted:

> *"If you underperform, then nobody wants to pair with you. We had the case of one developer who always seemed to have an excuse for why they were late in delivering their story cards – and invariably, it had to do with their partner. Over time, none of us wanted to pair with that person, and after the first iteration, that person left the team to take on a more managerial role – good riddance is all we had to say – it was a collective effort on the team's part because it was affecting overall project delivery."*

As one senior manager noted:

> *"It is a naked and blunt yardstick – people that deliver value to others are looked up to!"*

Spikes are the principle mechanism through which AgileCo teams handle unexpected roadblocks – both customer induced, and technical. When a development pair has significant difficulties in making progress in implementing a given story, they detach themselves from the team for a couple of days (typically less than a week) to further investigate the problem. This is akin to spiking the ball to stopping the clock in a football game. In the spike, they cycle back to the business analysts and in some cases the end customer to further investigate the meaning of the story, and to determine whether the customer's intent was captured effectively. Once this has been validated, they address the technical challenges through an exploratory strategy that includes accessing personal networks of experts, reading technical documentation, and in extreme cases, posting on technical forums. Once the problem is resolved, they merge back into the iteration. The use of spikes has to be approved by the team

leaders and the project managers. In the case when a spike cannot be executed, the team leads or project managers make the necessary modification to the iteration goals to ensure overall progress towards project goals. One of the ancillary benefits of spikes is the deepening of either technical knowledge or customer knowledge in the team members, and in some cases, both.

The two values that AgileCo emphasizes are the importance of collective code ownership, and value delivery to the customer. Since they are an offshore operation, often working with globally distributed teams, the importance of not breaking a build is strongly emphasized. We observed a build light, that went off when a build was broken, and the team was not allowed to work on additional story cards, until the build had been fixed. In addition to the overt mechanism of the build tower, the teams use peer pressure and subtle mechanisms to reinforce build integrity. One of the stories told to us by a senior developer reflected this:

> *"We emphasize integrity within the teams. One of the advantages of writing acceptance tests is that you know when a given feature is going to break the build. I had a young developer working on a feature for a system that was supposed to go live on Monday. He ran the unit tests for the module and checked in the code before he left on Friday so that the customer knew that everything was on-track. Both of us knew that the acceptance test would fail – I left early on Friday evening, and told him to make sure that the acceptance tests passed before checking in the code – when I came back late Friday night, I saw that he had checked it in – when I asked him about it, he told me that he was going to fix it by coming in Saturday – instead of blowing up – I used it as a teaching tool for the entire team. He was not supposed to work over the weekend – we follow the concept of a 40 hour week religiously, and if the customer had built and gone live earlier, the software would have failed – he never did that again, and the team learned the importance of customer value."*

AgileCo reinforces the importance of the team concept by issuing rewards based on team performance rather than individual performance. The team itself decides on rewards for team members to recognize excellence. The overall performance rating for an individual at the end of year incorporates peer feedback as well as the performance of the teams in which they were working. When someone misses a meeting without informing their team, it becomes his or her responsibility to buy coffee or ice cream for all the remaining team members. Furthermore, that person is used openly as an example of how not to be when they teach both their own students, as well as others. We saw an example of this when we were co-facilitating the XP planning game at a conference. One of the team members, who was responsible for bringing over the teaching materials could not make his flight connection and did not let this team members know about it. As a result we had to improvise and create some of the stories on the fly, and when he finally did arrive, our fellow facilitators from AgileCo made it a point to highlight that he had failed in his first responsibility – which was to his team, and that was unacceptable.

The weekly social functions at AgileCo, coupled with team events ensures that there is a lot of open knowledge sharing within and across teams. One of the incidents that we noted at one of the social events was a developer talking to a peer:

> *"I have been struggling with this for so long! Why didn't you tell me you knew how to solve it – the response came … you didn't ask."*

The managers also use these events to get a pulse of their teams, as they are so focused during work hours on the actual management of projects that they often miss subtle issues. As one of the project managers noted:

> *"I love to get these guys out of the office, because they are so open – they get an avenue for sharing their technical thoughts as well – most of the time it's all garbage to me, but it gives them two hours to just talk and learn from each other."*

3.4.3 Managing Customer Expectations

In 2005, AgileCo was faced with having a customer base that was not well versed in agile terminology. A case in point was pair-programming, wherein the customer questioned the idea of having two people programming together on a single system. They felt that they should not be paying for two people when only one was doing the work. AgileCo came up with the notion of ideal hours – the amount of time it would take to solve a given problem and chose to bill customers that way. One of the tensions of doing that was that the development team perceived that it was a comparison across the number of hours someone worked on a feature as opposed to the quality of the work. As the manager involved noted:

> *"One of things that I learned to do was to protect the team from the customer and executive management. We told the client that we were going to ideal hours, and instead of being adversarial with my team, I told them to give me ammunition to sell their ideas. Over a couple of iterations, I developed a conversion factor that effectively translated a story point to a standard number of hours. By keeping the story boards updated prior to a standup, I had all the information that I needed for the leadership calls, without having to gather 'overhead' data from my team."*

3.4.4 Teaching and Mentoring

Although AgileCo is dominated by young professionals with less than three years of experience (in 2005), they blended these young professionals with experienced team members, some of whom had not used agile methods prior to joining AgileCo. The advantage of having these experienced personnel is their deep belief in the process that AgileCo follows, with an ability to articulate from previous experience as to why the process worked. As a senior developer noted:

> *"People often miss the rigor of agile methods. One of the things that I have found over my career is that implementing CMM is commercially unviable – A lot of people*

that went through the CMM experience with me, became the best XP programmers. One of the things with agile methods, is that if you are in the organization for more than two or three years, is that you have to be really good at what you do or you are extraordinarily lucky!"

The experienced personnel also act as mentors to younger personnel. When discussing mentoring, the expression that was used often was that AgileCo was a *village without doors*, and you are really not considered to be a senior member until people ask for your help. When discussing the role of experienced personnel in the organization, a senior manager noted:

"You have a responsibility to make sure that junior members of the team get face time with the customer and have the ability to actually deliver a solution. You have to set up the iterations such that they can see how the customer uses the system, and make them gain a perspective that even when the solution seems sub-optimal from a feature standpoint, it probably delivers greater value from a solution standpoint to the end customer. We started an on-site project with six experienced people and three freshers (junior members) – midway through the project, we lost four of our experienced people, but the customers still valued the contributions of the team."

While the mentor-mentee relationship is both formally and informally enforced within AgileCo, another role that experienced personnel play is in peer-peer mentoring. All of the experienced members operate in a relatively ego-free environment, and recognize the critical role that it plays in AgileCo's culture. As one of the managers noted:

"When another senior colleague came on board, he wanted to put his stamp on the way things are done – we as an organization understand that, but he also has to understand the importance of what the organization currently has. It would not be proper to call our relationship a mentor-mentee relationship. Rather it is a peer-to-peer influence that I get to exert that has allowed him to adapt to the organization culture."

What their experienced team members have been able to do, is foster a culture of open communication and highlight the importance of taking responsibility for one's actions. They emphasize the importance of individuality within the organization's value framework. A case in point was the story of a young mentee who had incorrectly escalated the issue to senior leadership, and received a highly cryptic response. His mentor was wondering how is mentee would respond to the situation, and whether he should intervene to support him. As we were discussing the situation, the mentor received an email from the mentee, stating that he had made a mistake (the choice of words was more profane, but the meaning is still captured here), and he had called the people involved, and had listed a set of actions he was taking to fix the mess.

3.4.5 Discussion

AgileCo has had great success in adopting agile methods in the Indian context. Their open-kimono policy in sharing knowledge and in mentoring both their own employees and other organizations has positioned them as a leader in the use of agile methods in software development. As one of their senior personnel noted:

> *"We have achieved success through the nature of the projects we work on, and in indoctrinating our junior members."*

AgileCo's success in using agile methods as an organization-wide standard has enabled them to indoctrinate less experienced people in the process, and yet, balance out that indoctrination through the use of more experienced personnel, most of whom came from plan-based development environments. The intuitive notion that such senior personnel would find it hard to function in an agile environment, not embrace it and act as the drivers of change, was proved wrong. These mentors maintained an oral history of the limitations of plan-based approaches, but at the same time, emphasized the discipline it takes to execute agile processes.

Without the presence of experienced personnel, organizations run the risk of 'by-the-book' agile implementations that at a surface-level reflect the adoption of agile practices, but do not result either in the organization level transformation or the increased delivery quality that is expected. One of the limitations of indoctrination is the literal and rigid interpretation of agile methods that are espoused by junior members. This was one of the issues that came up over dinner conversations when someone talking about a more structured process like the Rational Unified Process was 'pooh pooped' by the entire table, with none of the junior members at the table being able to make a coherent argument for why RUP would not work in their organizational context.

As AgileCo continues to grow, it has to ensure that is does not become too inflexible to tailor the process to the problem. Their team based culture, ability to mix personnel with varied experience levels, and ability to maximize their geographical location makes them viable in the long run. As one of the business analysts visiting AgileCo from the parent office noted:

> *"This is the purest form of agile in the entire company. Anywhere else, I have to operate under the pressures of my client, but here, I am shielded from that, and can execute the process while, at the same time, delivering great customer value."*

A flip side to the geographical location is the limited domain nature of the projects that AgileCo gets to work on. At the fundamental level, the adoption of agile approaches has to change the nature of the work associated with software development, while at the same time fostering deeper understanding of the system being developed. As one senior manager pointed out:

"Agile+CMMI is possible, you just need to have the right mindset."

From our perspective, agile approaches provide a means of doing the work of software development and form mechanisms for grass-roots level transformation, while CMMI and other top-down process improvement efforts, provide the means of creating policy-driven change. That is not to say that organizations can just morph into following agile methods. There needs to be significant discipline on the part of senior leadership to support the process, and shield the teams utilizing these processes from pressures from the external environment. For instance, the project manager has the responsibility of preventing the team from thrashing either due to customer variation or due to resource challenges. Since using agile methods drives the organization towards the use of high performance team structures, effective change management strategies are needed.

Another finding that we initially found counter-intuitive, but on reflection found to be true, was the fact that India provides the best location for implementing agile processes and practices. It is important, however, to note that using agile methods requires the associated cultural change of enabling team member to work collaboratively with the customers, and adapt to change in near real-time.

3.5 GameDevCo

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. The development work still happened in the university town, while business development and operations were carried out in the nearby city center. This birth phase of GameDevCo spanned two years between 2000 and 2002.

GameDevCo's senior leadership team noticed that software was not being developed on-time, and on-cost. To add insult to injury, the constant bug-fixes and patches were being implemented in the 'live environment'. As a result, they imposed a very structured development process on the team based on the heavily plan-based development approach adopted at a major telecommunications company. In addition to imposing the structured development process, the operations team was made responsible for the deployment of the product. In this growth phase (between 2002 and 2004), they developed a business-to-business (B2B) model, in addition to the business-to-consumer (B2C) business model to create complementary revenue streams. The increased popularity of poker in the world (partly fueled by the World Poker Tour), highlighted the need for continuously improving the product to meet increasing competitive pressures.

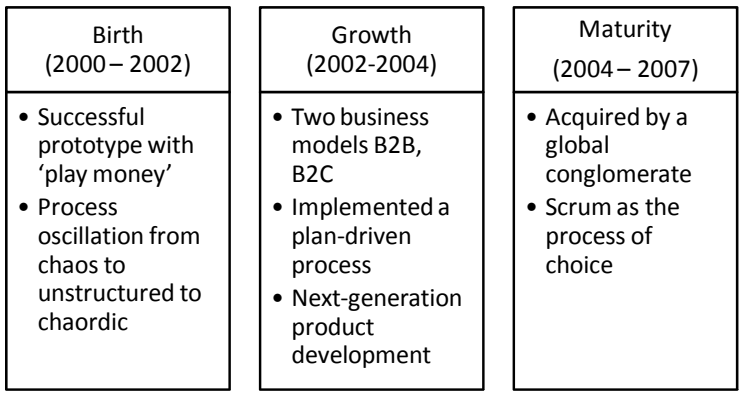


Figure 3-6 GameDevCo Evolution

The structured development process really did not improve organization performance, and given that the development team was improving the product incrementally, GameDevCo's senior leaders felt that the developers should be provided with the autonomy needed to make the decisions needed, while ensuring that some formal product data management was being carried out. In this maturation phase (between 2004 and 2007), they decided to adopt scrum as the development methodology of choice. To gain competence in the process, they hired an outside consultant to train their entire workforce on using this new process. As they learnt about the strengths and weaknesses of their product, they started adding new features, and capabilities, resulting in even greater complexity. They realized however that they were having difficulty in evolving their product hence they created a 'skunk works team' to architect a new product that would be as good as, if not better than the existing product. The term Skunk Works comes from the Lockheed Martin's Advanced Development Projects group that has designed some of the most advanced aircraft in the world. Nowadays the expression is used for a group within an organization that has great autonomy to work on advanced projects. As they were institutionalizing the process, two critical events took place:

- The consultant that was mentoring their adoption of scrum left the organization
- GameDevCo was acquired by a global conglomerate, whose core competencies were not in online poker.

When the data for this study was gathered in 2007/2008, GameDevCo was learning to deal with an additional layer management from a parent company that did not really understand their product/service portfolio. More importantly, since the parent company did not use scrum as the development process of choice, they were imposing new requirements on the development team by adding on a 'systems verification' step. In the hope of bridging the

growing gap between product management, operations, and development, all of GameDevCo's operations were bought together to a single city location.

We were aware that GameDevCo was in the process of adopting agile methods through our project champion, and we were hoping to understand how they had adopted agile methods, and how they planned to sustain it. It took almost two years before we got access to the organization for carrying out the study. 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, as shown in Figure 3.6. 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. These interviews were transcribed and themes extracted using a grounded theoretic approach. Memos were written to develop and refine the themes until no radically different insights were obtained.

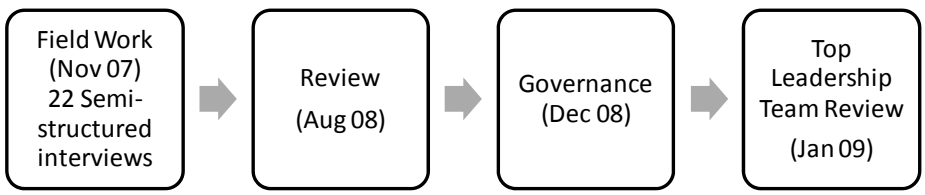


Figure 3-7 Data Collection and Analysis at GameDevCo

In Aug 2008, we had an interim review meeting with the project champion to ensure that the analysis was essentially correct, and to determine where the gaps existed that needed to be refined. In Dec 2008, we took a more in-depth look at the interactions between GameDevCo, and their corporate office, focusing more specifically on their governance process. While there are definite synergies and complementarities that can be leveraged between GameDevCo and their corporate headquarters, there is limited understanding of where exactly these synergies exist, and how best they can be leveraged. In January 2009, we conducted a review with the top leadership team, in which two key issues with respect to the requirements tool, and verification and validation were raised. This cycle of data gathering, inductive analysis, and stakeholder review, ensures that the findings from the case study are trustworthy.

3.5.1 GameDevCo Scrum Implementation

When discussing the implementation of scrum within GameDevCo with the teams, one of the contract personnel noted:

> *"I am a contractor who has been here for a year, and I still don't know what scrum is"*

This reflects the challenge associated with the growing size of the organization. In order to meet their strategic objectives, GameDevCo has had to hire contractors to join the scrum teams. The expectation with these personnel has been that they hit the job running, and hence, do not receive any training in the process. It is left to the teams to educate the contractors on the process. Given the intensity at which these teams work, the peer training also gets lost. Another issue that comes up arises from the graphic design work that is carried out as part of the user interface development – where the role of scrum is still unclear.

From an organization standpoint, there is still significant variation in how scrum is implemented, and what the expectations are from the process. As one of the scrum masters noted:

> *"We spend more time arguing about what the scrum books say about the process, rather than trying to figure out what the process is trying to do for us"*

The role of the scrum master is another area where there is significant variation within the organization. Team members described the job of the scrum master as taking care of the 'external stuff', and that it sometimes came down to filling out reports and getting coffee for the team. The scrum masters themselves see their role as being a key facilitator for the teams, and ensuring that they can, to some degree, shield their teams from external influences – the challenge being that most of the scrum masters were hired from outside GameDevCo, and did not understand the culture of the organization or the teams themselves. Furthermore, these scrum masters were often experienced project managers with limited exposure to agile methods. This lack of experience impacts the organizational learning significantly. When asked about the effectiveness of scrum in the organization, one of the teams noted:

> *"We are not doing scrum, we are sprinting"*

At the GameDevCo review meeting, it was pointed out that the original mentor that they had used for supporting Scrum adoption was rehired to come in and coach key team members on the use of the process. Since he had no clear mandate on what the organization was trying to achieve by hiring him, nothing came out of that effort.

One of the key impediments to successfully using Scrum at GameDevCo is the lack of coherence between the role definitions that GameDevCo uses, and that mandated by the corporate office to manage software development. The corporate process follows a conventional stage gate process, with five stages separated by gates, as shown in Figure 3.8. These projects, which are defined as temporary organizations of people and resources grouped together to achieve a given objective, are only seen to exist in the intermediate stages of idea validation through to full-scale production. Within each stage, the governance approach is a function of the cost, complexity, and coupling.

Figure 3-8 Corporate Process for Systems Development

The governance approach coupled with the stage of development defines the project management roles. The corporate process specifies the roles of Product Owners, Project Managers, and Project Leaders as key to successful governance. However, as it was noted earlier, GameDevCo only recently implemented the role of product owners, and is currently using Scrum Masters and System Architects to meet their governance requirements. When discussing the issues of role mapping to the corporate process, one of the scrum masters noted:

> *"The mapping is a bother which has been up for discussion numerous times. The current status is that we do not adhere to the guide to the letter. Depending on the size of the project, a project manager may or may not be present. The project manager also may or may not have a second layer of managers (in some instances scrum masters) beneath the own layer. There is also a large difference between us and corporate on this. Also, the concept of scrum master is quite different between the two sites"*

3.5.2 Requirements Management

As noted earlier, GameDevCo built the first version of their product through a strong foundation in the underlying technology (server-side, game engine, and client-side), and a deep passion for the game of poker. Since it started as a 'proof-of-concept', there is limited historical documentation on how the product was actually developed. The market success has resulted in continuous improvement of the product, carried out at a pace that has not always been conducive to documentation. As one senior architect noted:

> *"It's not that people do not want to generate documentation – there is a question of how much documentation is sufficient"*

Given that the product owners were only recently appointed (at the time of the study), the product backlog consisted of user stories at various levels of abstraction, which were often not consistent with either the current version of the running software or the product roadmap. When queried about this variance in the product backlog, one senior developer noted:

> *"When we find a bug or get a new requirement set from the product manager, we get into a room, and hash it out together, agree upon the design. The challenge we have*

is that the decision stays on the whiteboard, and often we forget what we agreed upon"

The impact of the variation in requirements is also seen in terms of the amount of time that the development team spends on actually refining the requirement to the point where it is implementable. When queried, one senior team member said that they spent anywhere between 50 to 75 percent of their time in getting the requirement to a point where they could work on it. Since the scrum teams are often not involved in the initial schedule estimations recorded in the product backlog, the downstream effect of ambiguous requirements is schedule overruns and poor system quality. The poor quality is primarily due to the multiple scrum teams being unaware of the design dependencies and decisions made by other teams. The expectation is that the various teams would update the requirements/user stories that have been refined/changed in a tool called FP_RM. The universal consensus in both GameDevCo's management and the product team is that the tool is inadequate. As more than one person put it:

"The tool sucks!"

This raises an important issue of having the right tools to support the process. When talking to one of the senior leadership team members about the tool, they said that it was a corporate mandate, and given the vast investment made in the tool already, they would just have to live with it. On probing further at a recent case review meeting, the history of the tool selection emerged, as shown in Figure 3.9.

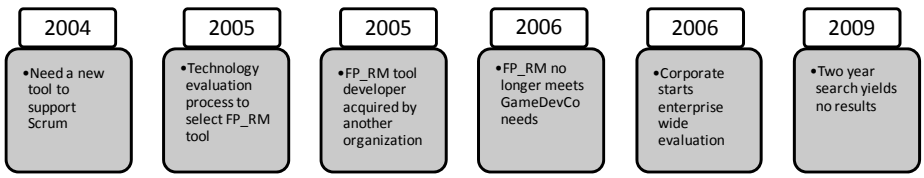


Figure 3-9 History of FP_RM Tool Adoption at GameDevCo

In 2004, as the organization was transitioning to adopting Scrum as the development process of choice, there was growing recognition among the organization members that the tool that they were using to manage requirements would not be adequate to support the process. In early 2005, they began a technology evaluation process to determine the best tool, and converged on FP_RM, which at the time, was being developed and sold by a small start-up company. When they piloted the tool, GameDevCo's focus groups loved working with it and believed that the tool was the right fit for the organization. A couple of months after that, the tool became the organizational standard. In the same timeframe, the company that developed the FP_RM tool was acquired by another organization whose emphasis was on plan-based development. As a result, newer releases of the tool contained greater process constraints on how requirements could be refined and adopted. By mid 2006, there was growing recognition

in the technical staff of GameDevCo that FP_RM did not work as expected, and that they would have to find an alternative tool. Since this represented a significant investment on the part of the organization, they had to clear the acquisition with the corporate office. The corporate office initiated a search for a tool that took two years and effectively found nothing. In the meantime, GameDevCo decided to find a workaround to the problem, by using an internal Twiki site to capture necessary documentation and knowledge – which has not been successful either. When the question was asked of the senior leadership team, one of them responded:

> *"Many requirements are poor but not all of them are bad – it has now become an urban myth that the requirements are all bad"*

Table 3-3 Understanding and Assessing Knowledge Loss at GameDevCo

	Programmer	***Lead Programmer + Design***	***Architect***	***Programmer + Design***
% of Design Changes Captured	10 – 15%	What's changed can be read in the release notes, but this is not updated regularly	There is a log for large changes - we catch about 50% - for newer requirements, it's about 80%	Larger changes are usually caught, but the smaller stuff is only talked about in design meetings
Timely Capture	No	No	Yes, if it is in a project	Design meetings every week
% lost	New designs captured using uml, but often neglected when refactoring	The rest	We lose the changes where the state before the changes was not documented	Smaller changes get lost, also, changes where the original state was not documented just get the new design and why (the previous state is not documented afterwards)

The result of the ineffective requirements management tool is that critical design information and design decisions are lost. While the expectation is that these decisions are either logged on the team Twiki or updated in the product backlog, we found that neither the decision log in the Twiki, nor the user stories and their dependencies in FP_RM were updated regularly. To explore this loss of information further, we interviewed four team members drawn from various roles. We discussed the percentage of design changes captured, whether it was timely, and what was effectively lost, and the results were telling as seen in Table 3.3.

From a programmer's perspective, a very small percentage of the design changes are actually captured, and even those that are captured, are not done so in a timely manner. When you are working in a short cycle time environment, this loss of design information has long-reaching impacts. This loss is further accelerated when teams refactor their code, and impact on the design is not considered. When discussing the importance of capturing this knowledge, the programmer noted:

> *"It would be nice to have guidelines on where and how documentation should be updated, and it would be even better if everyone did it the same way, for instance on the Twiki site"*

The expectation at GameDevCo is that in the worst case, the release notes would have sufficient documentation to support the team in understanding the current state of the product, but as one of the lead programmers pointed out:

> *"Historically, there has been more documentation, but we stopped updating it when no one was reading it (sometimes even release notes were not read) – we need to find a reason!"*

A case in point to illustrate this loss of design knowledge is the presence of a bug in the Legacy_Game product, which has been unresolved for almost two years. At the last review, it was pointed out that multiple attempts had been made to fix the problem. There had been no resolution yet, as no one knew where the source of the problem was. The conversation between the project manger (PM) of the Legacy_Game and the business unit head in charge of game development (GD_BU_Head) highlights the point.

> *Legacy_Game_PM: We know that this problem has existed for almost two years and it is really costing us in terms of customer loss on a daily basis. One of my guys told me that we know the root cause of the problem, and that he needs two days to fix it.*
>
> *GD_BU_Head: We tried to fix this problem last year as well, and we dedicated a week of development time to finding and fixing the problem – and nothing happened*
>
> *Legacy_Game_PM: Oh, but my guy said that it would only take a day to fix, but we need system verification time*

GD_BU_Head: We dedicated that resource last time as well, but we didn't find anything – right now, we are hoping to get the Next_Generation_Game out with the problem solved.

After the meeting, the Legacy_Game project manager was reflecting on the challenges of fixing the bug, and he noted:

"Initially, players had multiple mechanisms that they could use to achieve the same outcome – right now, we have standardized to this single mechanism and it has a bug – Apparently, we don't have the design knowledge that we need to effectively fix the problem"

3.5.3 Managing the Product Portfolio

The decision to concurrently carry out new product development, while sustaining the existing product, has caused significant tensions between the Legacy_Game and Next_Generation_Game teams. The decision to develop the Next_Generation_Game was partly motivated by the challenges in sustaining the current product. However, since Legacy_Game is the primary source of revenue for the organization, it has to be maintained while it is in the live environment. GameDevCo had two teams of almost equal size working on the two projects, but imposed a barrier between the two teams that prevented them from working together. One of the challenges that the Legacy_Game team faces is the perception that their product is flawed. One of the business unit heads that we spoke to strengthened that perception, when he said:

"Legacy_Game code is a mess"

Almost in contrast to that, one of the developers pointed out:

"The guys that created the mess are in Next_Generation_Game now! We have completely refactored the mess, and it is now more modular and elegant than Next_Generation_Game. We are delivering and improving Legacy_Game faster than they can catch up"

When reflecting on the importance of aligning the two products, one of the Legacy_Game team members explained:

"You have to understand that we are the ones that are making money for the company right now – We are delivering in the real world, while Next_Generation_Game is working towards a vision – you can't compete against a vision"

This tension between the products is further highlighted by the existence of the 'Ghost-in-the-room' phenomenon. The Legacy_Game team is constantly second-guessing what they should improve/implement as part of their daily activities of patching and releasing. This is fueled by

their lack of knowledge of what Next_Generation_Game is doing, as well as uncertainty with respect to what the team members are going to be doing once Legacy_Game is phased out.

At the review meeting, it was pointed out that they had recognized the problem, and that all development work on Legacy_Game had been suspended. All the team members from Legacy_Game migrated to working on the Next_Generation_Game with the hope of releasing the next product early next year. As the Legacy_Game project manager pointed out:

> *"Right now, the project is on life support – we have four developers working on it and are really not trying to fix any of the major issues that come up."*

3.5.4 Organizational Learning

This tension between the two teams, coupled with the organizational barriers imposed on the teams, further translates to increasing knowledge gaps – as one of the lead programmers commented:

> *"We don't talk to each other – the idea was that we would form self-organizing communities that would share knowledge, but that hasn't worked out. We have tried every form of organization from integrated product teams to communities of practice to get to where we are today"*

One of the issues that was brought up by almost all of the participants, and further validated at both the reviews, was that GameDevCo was too busy to learn. This impacts GameDevCo from both a strategic and operational perspective. At the time of the study, they were continuously improving Legacy_Game and concurrently developing the Next_Generation_Game. This imposed significant cost and schedule pressures on both teams. The Legacy_Game team members were in *patch-and-release* mode most of the time, and the improvements they made were incremental and often not captured along that way. The Next_Generation_Game team, on the other hand, had already suffered two release slippages and was yet to demonstrate the ability of their product to meet requirements with respect to having the requisite features, and meeting the load conditions of live operations.

When reflecting on the fact that GameDevCo was too busy to learn, the head of the GD business unit pointed out that when he was managing the platforms group, they had some of the same challenges as the Next_Generation_Game team was having.

> *"We had the same issues with respect to releasing buggy software that did not always have all of the features that were needed – I had to make an executive decision to stop releasing for three months till we base lined where we were – Today, almost two years after we did that, the platforms group has not missed a significant delivery"*

The story illustrates the challenges that GameDevCo currently has with respect to maintaining their rapid pace of operations. This need for speed is further emphasized by the

corporate office's emphasis on meeting schedule requirements. From an execution perspective, the sprint review meeting is designed to support organizational learning about both the product and the process. Since the scrum masters each have individual philosophies of what scrum is, and how it should be implemented, the learning from the sprint review meetings has become almost non-existent. As one of the scrum teams noted:

> *"Our retrospect starts with the scrum master saying: 'What didn't work in this sprint? What can we improve in the process?'... We look around and in 5 minutes we are done"*

The focus on process improvement supports teams that are just beginning their scrum journey in both gaining a deeper understanding of the process, as well as in taking ownership of the process. As the team matures, process focused retrospectives produce diminishing returns. The emphasis on sprints, takes away from the overall learning at GameDevCo as well. As one of the senior developers explained:

> *"We could have short cycles in the beginning as there was not that many dependencies between the teams – now we need time to integrate and time to get the systems to work together – but we are compressing the sprint planning and sprint reviews instead"*

3.5.5 Verification and Validation

One of the challenges that GameDevCo faces is getting discipline into system verification and validation (V&V). As the head of V&V noted:

> *"We only started thinking about system verification as a capability a year and a half ago – I was brought in to bring some discipline into V&V, but it has been a tough road. We have young developers here who have always been successful; they don't know when a project fails"*

This track record of success has to some extent blunted the urgency of building up system V&V capability. The analysis of the interview data highlights two causes for this apathy: the lack of a system owner; and a lack of understanding of what V&V means in the product development context. The documentation-gap, noted in the requirements section, has been further exacerbated by the lack of a single 'belly button', namely, a system owner, who defines/controls the system level requirements. As one of the quality analysts noted:

> *"We don't have a systems owner – so you don't know if the system-level is actually correct. Since we don't have good requirements, we had to hire an external consultant to reverse engineer the system level requirements. The consultant was very good for the first three months, but other people found out, and we lost him."*

This problem is even more apparent in the context of new product development, wherein the high-level architecture is understood, but the requirements are being refined through the creation of rapid prototypes. The transition to scrum, as the development methodology across the entire organization, shifted the unit testing burden on to the developers. The scrum team built the test cases based on the scrum backlog, and aggregated the test cases for each of the scrum tasks to form the overall test suite, instead of deriving the system-level tests and acceptance tests based on the product backlog. One of the developers working on the product version that is running live said:

> *"The system is working fine with the test suites we built, and we have automated the testing process to the point where we are able to run all of the test suites overnight"*

In almost direct contrast to that view, is the perspective of members of the QA team:

> *"Testing is not just automation and running tests every night"*

What gets lost in the almost adversarial relationship between quality analysts and the developers, is the fact that they are working towards a common purpose. Currently members of the QA team are not part of the actual scrum team; instead, they are a common resource pool that all the scrum teams working on the product use collectively. This organization structure is effective if the system requirements are clear, and the QA team can effectively plan its own sprints to support the overarching product roadmap and expected release cycle. But as discussed earlier in this section, the system level requirements are not known for either Legacy_Game or the Next_Generation_Game.

From an infrastructure standpoint, the V&V team does not have a test environment that replicates the operational environment. As a result, they cannot determine if the system under test will actually perform as expected under live conditions. This is further reflected in the long system integration and verification cycle times, making V&V activities even more unpopular with the scrum teams.

A case in point on the resource crunch and poor usage of available resources was made in the top leadership team review meeting. The Legacy_Game project manager pointed out that his team needed to release a critical patch for the product and that they had not been able to schedule time with the system verification team. The system verification lead pointed out that, as the test environment had been scheduled for the Next_Generation_Team, she could not allocate the resources at the time. She further pointed out that she had advised the Next_Generation_Game team to postpone system verification as they had not yet generated the required acceptance tests, and that the system would fail the test. As it turned out, the system verification demonstrated that the new product did in fact fail, and that the limited resources could have been used to support Legacy_Game.

3.5.6 Managing Human Resources

One of the critical challenges facing GameDevCo is the management of its human resources. As one of the business unit heads observed:

> *"Design knowledge is in the head of a few people"*

GameDevCo cannot afford to lose this knowledge that is resident in the heads of a few key people, typically those that have been with the project since its inception. GameDevCo's transition from a technology development company to a product company has created an identity crisis within the organization, which has now spread to its people. As a lead programmer lamented:

> *"We are losing talent all the time – if the right thing came up, I would move. I love what I do but it doesn't feel like they have the same focus on technology anymore"*

GameDevCo's transition from the university town to the big city has exposed their talent base to a larger, more competitive market, where they can satisfy both their need to work on technology development, as well as obtain greater financial benefits. During the top leadership team review meeting, it was pointed out that most of the developers who had moved from the university town to the city, no longer worked for GameDevCo anymore. Another area of concern at GameDevCo is the mismatch between how developers expect to be rewarded and how the organization executes. As one of the developers pointed out:

> *"We got a bonus for delivering something on time –it was a monetary bonus which was something in the order of 10% of my monthly pay – it is what I would spend on a Wii – It would have had so much more impact if they had gotten the Wii's for us and had them engraved to commemorate the moment – I have raised this issue with our HR people multiple times, but they don't listen"*

3.5.7 Discussion

GameDevCo's agile journey raises interesting issues when viewed through a socio-technical lens. The roles-artifacts-ceremonies elements of scrum discussed in section 3.4.1 forms an effective foundation for discussing the current state of the process within GameDevCo. In keeping with the generally accepted principle that an effective team size for a scrum is seven plus or minus two people, GameDevCo uses multiple scrum teams within the product development team. To support the coordination across the multiple scrum teams, they created the 'scrum-of-scrums' ceremony. Our observations of both the daily scrum meetings, and the scrum-of-scrums indicated adherence to widely adopted practices, and highlighted the atmosphere of trust within the teams. In the daily scrums, team members openly shared their challenges, discussed accomplishments, and plans for the current day. At the scrum-of-scrums meeting, we observed the team's use of bug types to focus on critical problems. While we were not able to observe either the planning meeting or the sprint review meetings,

the interviews revealed limitations in the implementations of both the ceremonies. In sprint planning meetings, the root cause of the problems are the maturity of requirements. The limitations of the sprint review meetings are associated with organizational learning.

The requirements issue also highlights the challenges with respect to maintaining the product backlog artifact. The sprint backlog is visually maintained on the project wall, which shows the current state of the sprint backlog, the work in progress, and the work completed. Since the scrum team updates and maintains this artifact daily, it has been extremely successful. One of the innovations in GameDevCo is the automatic generation of the burndown charts that are used both for team situational awareness, as well as upwards reporting by the scrum masters. Additionally, GameDevCo makes extensive use of information radiators (in the form of screens showing build data, bug rates, overall project progress) within the organization – with mixed results. In the case of the new product development team, a large percentage of the software had not yet been developed. As a result, most of project appeared red, either due to build issues or due to system verification and validation challenges, resulting in most of them ignoring the screens. From a role perspective, the challenges were centered mostly on the scrum masters and the product owners.

Product development and sustainment is more analogous to a marathon, rather than a sprint. While it is possible to sprint a marathon, it requires tremendous discipline and stamina. From an organization perspective, GameDevCo recognized the need for increased team autonomy to support better product quality and enhance organizational innovation. As we have seen in the case study, there are six areas that GameDevCo has to focus on, namely: Scrum Implementation, Requirements Management, Managing Product Evolution, Organizational Learning, Verification & Validation, and Human Resource Management.

The scrum framework can be effectively used in product development, but as with any process, the impact of the process in adding value to the organization, is more important than staying true to the ideology of the process. That is not to say that ideology is not important – it is critical that the ideology be consistent across GameDevCo. The fragmentation of the implementation can be traced both to the loss of their agile coach, as well as their hiring of relatively inexperienced scrum masters. An agile process requires an agile tool – while the tool that is currently in use in GameDevCo works for managing archive quality requirements, it has not been able to meet the rapid development cycle time, as design refinements occur in parallel. With a rapidly evolving product, and a mobile talent base, organizational learning becomes critical. The process does support learning, however, the supporting infrastructure and the implementation of the process itself does not support learning. Last, but not least, GameDevCo has to build verification and validation capabilities to support product evolution.

The six areas that GameDevCo has to focus on are challenges that most product development organizations face – the use of agile methods has resulted in the organization being able to more clearly see some of the challenges. While the short cycle times bring the limitations to the surface, the use of scrum itself has contributed to some of the challenges. The lack of

consistency in the adoption and implementation of scrum results both in visible heterogeneity across the organization, but more importantly, reduces the ability to shift personnel between teams seamlessly.

3.6 Analysis

When AgileCo and GameDevCo are compared to the common baseline of the values and principles of agile software development movement, their similarities and differences emerge. When it comes to the importance of *individual and interactions over processes and tools*, AgileCo embodies the value, partially because they already have the tools and processes in place that supports their work activities. Since AgileCo has a strong culture in place that supports information sharing and open communication, it is easier for them to foster interactions. GameDevCo's challenges arise both from the unhealthy competition between the two principle teams, as well as the lack of tools and processes. Their emphasis on individuals and interactions currently occurs in some ways to the detriment of the organization.

Table 3-4 Adherence to Agile Software Development Values

Values of Agile Software Development	AgileCo	GameDevCo
Individuals and interactions over processes and tools	strong	partial
Working software over comprehensive documentation	strong	partial
Customer collaboration over contract negotiation	strong	partial
Responding to change over following a plan	partial	strong

AgileCo's development approach minimizes the need for comprehensive documentation through the creation of software that has 'sufficient' information embedded in the code and the test cases/suites. While the traditional notions of documentation may not be met by AgileCo, they leverage tools that will automatically generate the documentation needed from other sources (such as code, story cards etc), which can further be refined by their writers. GameDevCo suffers severely from a lack of legacy documentation, since they are too busy fixing and developing software to generate 'any' documentation.

From a collaboration standpoint, AgileCo's reputation has enabled them to build trust-based relationships with their clients. Furthermore, they use an innovative measurement approach that enables them to communicate with their clients in conventional terms such as man-hours while internally using metrics such as story points. GameDevCo's recent acquisition by another firm has made collaboration a little difficult. Since their funding is governed by headquarters, strategically they are operating under a 'contract' mode, which is reflected in the behavior of their developers.

Since AgileCo is in the software services business, responding to change is not always possible. Project managers are bound by a contract, but do provide the teams with the degrees of freedom that they have available in that context. GameDevCo on the other hand, has

almost been in a purely reactive mode, constantly evolving to meet the needs of current users, while not adhering to a plan.

In many ways, AgileCo represents the ideal case, wherein a firm chooses to use agile methods because the process fits the organizational culture. This culture is constantly reinforced by example by the senior leadership team and it has the learning mechanisms needed to evolve the process to meet the organization's needs. On the other hand, GameDevCo is still trying to find the right process to match its needs, as seen by the mappings in both Table 3.4 and Table 3.5.

Table 3-5 Adherence to Agile Software Development Principles

Principles of Agile Software Development	AgileCo	GameDevCo
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	strong	partial
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	strong	strong
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	strong	mixed
Business people and developers must work together daily throughout the project.	strong	mixed
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	strong	strong
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	strong	mixed
Working software is the primary measure of progress.	strong	mixed
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	strong	strong
Continuous attention to technical excellence and good design enhances agility.	strong	mixed
Simplicity--the art of maximizing the amount of work not done--is essential.	strong	mixed
The best architectures, requirements, and designs emerge from self-organizing teams.	strong	strong
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	strong	mixed

The values and principles form an effective 'roadmap' against which agile adoption and sustainment can be managed. The importance of a vision and a coach should not be minimized. Through the cases we have provided rich descriptions of the elements of successful and non-successful agile adoption that organizations can use both as a signpost within their own transformation, as well as to determine pit falls that can be avoided.

4 CREATING SYSTEMS OF INNOVATION

4.1 Introduction

Innovation has been recognized as a critical element for organizational survival in the current operational environment; yet, the debate remains ongoing as to exactly what innovation is, and how it can be fostered within an organization. The discussions surrounding innovation have led to the creation of buzzwords like the 'creative economy' (Coy, 2000), as well as new organizational roles at the corporate level, such as chief innovation officer (Nussbaum, 2005). The importance of innovation for organizational success can be traced back to Schumpeter (1927), wherein he defines innovation as:

> *"Changes of the combinations of the factors of production as cannot be effected by infinitesimal steps or variations in the margin. They consist primarily in changes in the methods of production and transportation, or in the production of a new article, or in the opening up of new markets or of new sources of materiel."*

This definition of innovation has stood the test of time, and while ideas like entrepreneurship have emerged in more recent time, the essence of innovation remains the same. As Drucker (1985) points out almost 50 years after Schumpeter when discussing innovation:

> *"It is the means by which the entrepreneur either creates new wealth-producing resources or endows existing resources with enhanced potential for creating wealth."*

In other words, innovation can be in the product or the process (Tushman & Nadler, 1986). It can be classified based on the whether it is incremental or radical (Ettlie et al., 1984), or modular or architectural (Henderson & Clark, 1990). At the heart of the innovation puzzle is the ability to connect the strategy and tactics associated with developing a system of innovation from a macro perspective, with the mechanics of effectively transitioning ideas into finished products and services at the micro-level. This chapter focuses on the second research question addressed in this thesis:

How do software organizations create a system of innovation that successfully addresses strategic challenges while meeting operational needs?

To answer this question, we have to address both the macro and the micro perspectives, as shown in Figure 4.1.

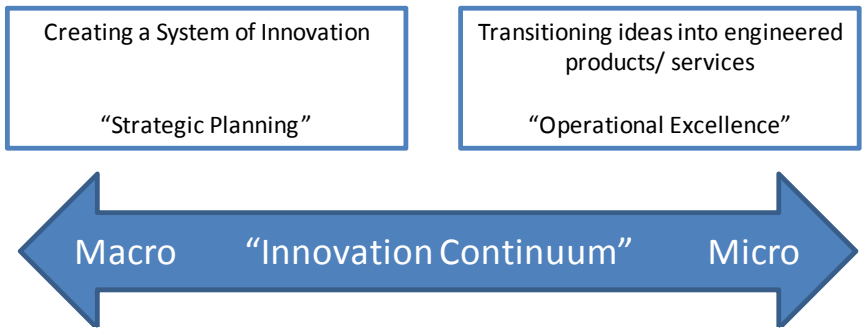


Figure 4-1 Understanding the Innovation Continuum

The remainder of this chapter is organized as follows: in Section 4.2, we discuss the three common approaches that organizations use to innovate, namely, internal R&D, technology scanning, and last but not least, open innovation. We develop a descriptive understanding of how Rockwell Collins has been successful at integrating the approaches to create a system of innovation in Section 4.3, and use that understanding to develop an ideal system at the strategic level in section 4.4. We then develop a framework that identifies the key dimensions of the problem of transitioning ideas into engineered products or services, and illustrate how these dimensions were addressed (or not) in the GameDevCo case study in section 4.5. The final section discusses the implications from both the case studies for further research as well as industrial practice.

4.2 Macro-Perspective on Innovation

One of the challenges of implementing a system of innovation is in understanding the nature of innovation and how it occurs. Reflecting on the field in the 1980s, Van de Ven (1986) noted that, historically, research in the area was both narrowly and dominantly technically focused, and furthermore, the research had focused mostly on the internal model of organizational innovation through internal research and development. Focusing on the management challenges of innovation, he identified the four factors that needed to be understood that facilitated/inhibited innovation from occurring:

- the human problem of managing attention
- the process problem of managing good ideas into good currency
- the structural problem of managing part-whole relationships
- the strategic problem of institutional leadership

These factors have to be understood in the current day context and strategies for ameliorating these limitations are shown in Table 4.1. The issue of managing attention brings to the forefront the issue of inertia and path dependence for enabling innovation. By inertia we refer

to the organization having built a certain resistance to change because their expertise is focused on their current products and services. Since organizations build their competencies around work that they have done earlier, their future innovations necessarily build upon the path that got the organization to where it is today.

Table 4-1 Strategies for Managing Innovation

Factors *(Van de Ven, 1986)*	Strategies for Improving Innovation
Human problem of managing attention	Managing inertia and path dependence
Managing good ideas into good currency	Creating infrastructure to support innovation
Managing part-whole relationships	Knowledge Management; Managing requisite variety
Institutional leadership	Recognizing and planning for innovation

This idea of inertia and path dependence is also captured by Leonard-Barton (1992) as 'core rigidities' of the organization. The challenge is not just in generating ideas, but in actually moving the idea beyond the concept stage into production phase while there is still a market for the product or service. Consider the introduction of the T-Mobile myTouch 3G in the United States, Dan Ionesco of PC World notes (2009), "it offers too little too late". His review compared it to comparable phones from the competitors such as the iPhone, leading to a recommendation to wait for another product. If the organization does not have the necessary infrastructure to enable and drive innovation, then ideas will not move into finished products or services within the time frame in which they provide competitive advantage to the organization. This infrastructure includes the creation of the right metrics and incentive systems, the right human capital strategy, and creating an integrated organization to formulate and shape the firm's innovation strategy. Part of that challenge of transitioning ideas into finished products and services is managing the 'part-whole' relationships. Ideas are typically generated by a single person or a small team, but the finished product/service involves a larger team, and often may not include the innovator/innovation team. Managing part-whole relationships is about ensuring that knowledge is shared across the organization. This is similar to Dougherty's conception of product innovation as the creation and exploitation of knowledge to link market and technological possibilities (Dougherty, 1992). It is important to note here that our notions include both products and services. Another perspective of the part-whole relationships is managing the diversity in perspectives that exist and evolves over the time it takes an organization to recognize an idea as being important till it actually becomes a finished product or service.

Organizational innovation is not sustainable if there is no institutional leadership, in other words, the organization has a strategy for how to simulate, drive and sustain innovation. In the classical sense, it was about the organization having a clear idea of the markets it wanted to serve, and developing technology strategies and product roadmaps to ensure that it could grow sufficient capabilities to meet those market needs. In the current environment however,

the organization cannot afford to rely on static strategies for innovation – it has to recognize and build an innovation strategy that is both fluid to adapt to evolving conditions, and yet stable enough to address its current market base. Further, we delve into the three commonly used mechanisms for structuring organizational innovation, and highlight the challenges associated with each of the three approaches.

4.2.1 Internal Research & Development

Internal research and development is the traditional innovation pathway to most firms. In fact, the first research and development laboratory for GE was established in 1895 (Suits, 1953) to provide an exclusive focus for R&D. The key driver for carrying out internal research and development is that it was, and still remains, a key element of competitive advantage both for firms, as well as nations. The National Science Foundation (NSF) in the United States has been conducting a national survey since 1957 as a means of understanding the amount of R&D spending in industry. The changes made to the survey methodology in 2008 (Wolfe, 2008a) were motivated by the need to understand the current day realities of R&D spending and investment. In the 1950's, the largest sponsor for research and development was the government itself and firms could tap into that source of funding in addition to investing their own funds. This trend has completely reversed now, with businesses serving as the dominant source of research funds, as illustrated by the R&D spending in the United States for the year 2006 (Wolfe, 2008b). Of the total 247,669 million dollars, 24,304 million (9.8%) was sponsored by the government, while 223,365 million (the remaining 90.2%), was sponsored by industry.

Table 4-2 Evolution in the Dynamics of R&D Spending (Wolfe 2008)

1950s	2000s
• Government largest source of R&D funding • Business largest basic research performer • Dominated by manufacturing companies • Large companies dominate R&D • Domestic competitive focus • Focus on in-firm science and technology resources and central research labs	• Business largest source of R&D funding • Academia largest basic research performer • Increasingly performed in service industries • Increasing R&D activity in small companies • Global competitive focus • Increased leveraging of science and technology resources outside the firm

Equally important is the nature of the research and development work being performed. Understanding the extent of R&D expenditure across the three categories of basic research (no final objective has been identified, but the outcome may be beneficial to the firm),

applied research (some product/service identified that leverages the knowledge generated), and development (commercializing the product/service), as well as who carries out the bulk of the activities highlights the changes in the last fifty years. While the independent research and development groups within large organizations carried out a majority of the basic research, today, it is universities that conduct a majority of the basic research and development.

There has also been an important shift in *where* R&D is being carried out. In the 50's, the barriers from both a human capital and from a financial investment standpoint, were too high for small and medium sized firms to carry out R&D activities as a 'core competence'. Grabowski and Baxter's study of the chemical industry (Grabowski & Baxter, 1973) found a phenomenon akin to price leadership with an 'action-reaction' pattern between the industry leader and their competitors. In other words, the competitors would observe the actions of the leader and map their actions accordingly. The advances in computing technology and the opening up of the global market for talent has lowered the barriers to entry for both small firms, as well as individuals to carry out research and development. This virtual elimination of barriers has resulted in firms sourcing innovation outside of their own boundaries, either through technology scanning or through open innovation. The dominant issues when focusing on internal R&D as the sole innovation approach are:

- Managing the 'not-invented-here' (NIH) syndrome (Katz & Allen, 1982), wherein firms reinvent the wheel instead of building on what is already accomplished just because it was not developed within the boundaries of the firm.
- Not thinking 'outside the box', since research is built around the competencies that the firm already possesses.
- Finding the right measures to assess the effectiveness of R&D projects. As (Hauser & Zettelmeyer, 1996) emphasize, there is no 'one size fits all' set of metrics to measure R&D projects. This is because the nature of the project, to a large extent, defines the appropriate metric that needs to be used.

On the other hand, internal R&D ensures that the tacit knowledge developed during the idea generation and concept development phases is accessible to the team during the engineering of the finished product or service. In addition, a strong internal R&D team provides a more effective means for leveraging approaches, such as technology scanning and open innovation.

4.2.2 Technology Scanning

Technology scanning was formally proposed in 1968 as a means of creating a research and development program for the state of Connecticut (Simons 1971). The approach structures the five-step process of:

1. Developing future system requirements
2. Obtaining buy-in from senior decision makers to establish congruent goals

3. Developing programs to achieve these goals
4. Measuring the effectiveness of the programs in meeting the goals
5. Revising/creating new programs in a manner that ensures that effectiveness criteria are usefully and accurately defined.

This cycle when conducted in a synergistic manner, provides the strategy for driving research and development efforts, and to a large extent, become the front end for internal R&D efforts. When internal R&D projects do not meet the organization's needs, it leverage technology flows (Scherer, 1982), and exploit spillovers (Jaffe, 1986). By technology flows, we refer to the distinct products or services that are provided by a supplier, and can be incorporated immediately by the buying firm to create new value. Spillovers on the other hand occur when members of the organizations R&D team visit other, similar organizations, participate in conferences etc to identify ideas that are transferrable to the organization. In the case of spillovers, the transferred idea cannot immediately be leveraged to create new value.

In more recent literature, the emphasis on technology scanning has expanded to include the connection between organization strategy and research and development. Van Wyk (1997) points out that strategic technology scanning should be part of the larger corporate environmental scanning approach that provides insights to senior leadership. These insights would then be able to support more refined and informed decision making with respect to organizational innovation. Granstrand, Bohlin et al. (1992) found that technology acquisition was the dominant management issue in Swedish and US multi-technology corporations. In addition to technology flows and spillovers, firms can choose to acquire other firms as a means of acquiring their capabilities. As Wernerfelt points out, mergers and acquisitions provide an opportunity to trade otherwise non-marketable resources and to buy/sell resources in bundles (Wernerfelt, 1984). Recognizing the importance of the globalization, organizations have distributed their research and development efforts across multiple geographical locations. This distribution both supports the generation of multiple perspectives on the same problem, and also provides closeness to the local markets that may not have been otherwise achievable. This internationalization of R&D has, to some extent, been a function of national culture as well. For instance, Grandstrand (1999) found that Swedish firms were early adopters of internationalization efforts, while Japanese firms lagged in the initial stages, and were beginning to increase their R&D internationalization efforts. Similarly (Von Zedtwitz & Gassmann, 2002) in their analysis of 1021 R&D units found that research itself was conducted only in five regions, while development was more geographically distributed. They further identified the location rationale to be either due to access to science or access to markets.

Technology scanning serves both as a means of creating the firms innovation strategy, as well as a means of executing on a developed strategy, as shown in Table 4.3. When used as a strategy formulation tool, it requires the organization to create the necessary infrastructure and incentives to stimulate spillovers, and ensure that senior leadership both understands the implications of the findings presented (in order to obtain support for the strategy), and can

provide the support needed to drive the innovation strategy. From a strategy execution perspective, technology scanning requires the organization to manage its technology flows effectively through contracting and licensing, and more importantly through the synergistic joint ventures and acquisitions.

Table 4-3 Technology Scanning for Strategy Formulation and Execution

Strategy Formulation	Strategy Execution
Leadership buy-in to drive decision making; Creating incentives and systems to support spillovers	Managing technology flows through contracting and licensing; Acquiring capabilities through synergistic acquisitions and joint ventures

4.2.3 Open Innovation

The term open innovation came to the general consciousness through Henry Chesbrough's article in Sloan Management Review (2003), wherein he highlighted the fact that in the past, internal R&D was not just a strategic asset to the company, but it also served as a formidable barrier to entry for their competitors. Using the metaphor of the firm boundaries acting as the edges of a funnel that successfully refines ideas into finished projects and the market, Chesbrough introduces an alternative model of innovation that firms can adopt, namely the open innovation model. In this model, organizations create wealth by deploying and exploiting pathways to the market that may be outside its current market. It is important to note here that open innovation allows for ideas to both leave the firm boundaries on development, but also to enter the firm boundaries at a later, more mature stage of research or development. This notion of permeable firm boundaries is key to open innovation.

One of the challenges of making open innovation accessible is in finding the right metrics to manage the innovation process. The traditional approach to managing research projects, in which roadmaps could be developed because resources (both own and competitors) were well defined, and the rate of change of new information arrival was relatively small, the dominant driver is the minimization of false positives, as opposed to managing false negatives (Chesbrough, 2004). By false positive, we mean projects that are thought to be successful within the business model of the firm, but in reality result in significant losses. False negatives, on the other hand, are projects that do not fit within the current business model or projected needs, and the process of managing these requires a judgment call on the part of the innovation leadership team in determining whether or not further investment within the project will support the long-term growth strategy of the firm.

Consider the case of Procter & Gamble's strategy of Connect & Develop, which produces more than 35% of the firm's innovations (Huston & Sakkab, 2006). P&G's emphasis on open-innovation arose from the simple fact that R&D productivity had leveled off in early

2000, and their innovation success rate of 35% (while extremely high from an R&D productivity perspective), was not deemed to be sufficient to drive top-line organizational growth. P&G's innovation strategy was built around a strong understanding of customer (through their annual identification of the top ten areas) and market (through the identification of adjacencies, and playing technology game boards) needs. These needs could then be translated into action using either their own internal R&D network or through their technology entrepreneurs. For instance, the introduction of the 'Mr. Clean Magic Eraser' product, was first identified by a technology entrepreneur in Japan in 2001, evaluated in 2002, and finally launched in 2003. Since then, it has become one of the most successful products that P&G launched. Yet, the connect & develop strategy for P&G is not without its pitfalls, and requires both internal resources (for doing additional development work, and scaling up for production) and senior leadership commitment to making it successful. In addition to using an internal innovation network, firms can leverage open networks through organizations such as InnoCentive and NineSigma. InnoCentive's approach of connecting 'solvers' and 'seekers' (Allio, 2004) enables firms to pose difficult/challenging problems to the global community without necessarily revealing their own strategic positions.

As with technology scanning, open innovation can be applied to both the strategy formulation process, as well as the strategy execution process. For open-innovation to be successful, it has to be part of the organization's larger innovation strategy, and more importantly, it requires a significant change in the organization's culture (especially in organizations with a rich history of in-house innovation).

4.3 Rockwell Collins: An Innovation-Centric Enterprise

Having identified the three strategies that organizations use to create a system of innovation, we wanted to carry out a case study of an organization that we believe to have been successful at creating a system of innovation. Using publicly available archival data such as annual reports, magazines and news articles, we identify the factors that enable Rockwell Collins to successfully build a system of innovation.

Rockwell Collins is a global leader in aviation electronics and communication systems development. They became an independent company in 2001 when they were spun off from Rockwell International, and over the last eight years have become one the most respected avionics companies in the market. In fact, this year, Aviation Week and Space Technology ranked them the best company in the avionics segment (Velocci, 2009), beating out Honeywell, Thales and Raytheon. They have long been recognized for their operation excellence through their Lean Electronics™ program (George & Labedz, 2006), but it is only one element contributing to their success since becoming an independent company. Their emphasis on a balanced business model (generating revenue equally from the commercial and government sectors), establishment of shared services, and recognition of innovation as the key competitive differentiator have all contributed to that success. From the perspective of this chapter, we focus on the innovation aspects as it forms the foundation for their sustained growth and success. This is reflected in their five organization values (Rockwell Collins,

2009a) of *Teamwork, **Innovation**, Integrity, Customer Focus* and *Leadership*. Delving a little deeper, innovation is discussed as:

> *"We understand that the best source of growth is the creativity of our people. We support that creativity through investment, process efficiencies, professional development and knowledge management."*

4.3.1 Organizing for Innovation

Rockwell Collins recognized early that while it would be easiest to carry out all of their research and development in-house, it was not a sustainable strategy. As Clay Jones noted (2009),

> *"Innovation rarely happens by accident. Instead, it takes discipline, creative thinking, collaboration and careful preparation"*

They recognized that long-term effectiveness in bringing new products and services to the market faster than their competitors would need an alternative strategy – one that would leverage the strengths of their own talent base, as well as exploit the global brain. Their approach to internal R&D is through Rockwell Scientific to focus on basic research, the Advanced Technology Centers to focus on next generation products and services through applied research, and domain specific centers of excellence for specific products. This architecture is similar to what most large corporations do, but what makes Rockwell Collins unique, is their ability to leverage approaches such as technology scanning and open innovation to truly define their innovation system, and foster a shared understanding of the expectations between the various stakeholders involved in innovation.

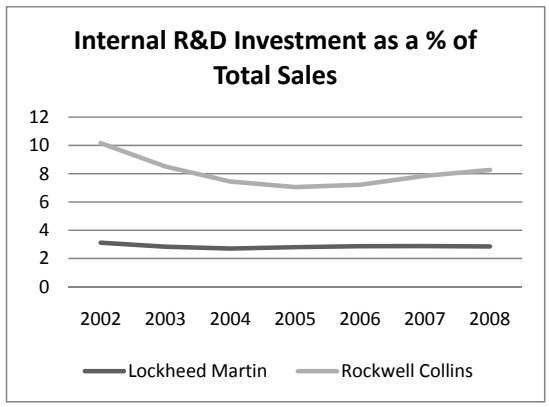
Nan Mattai, the senior VP for Engineering and Technology pointed out (Wojciechowski, 2007):

> *"We have to be smart innovators...We have to be willing to collaborate inside and outside of our own four walls in order to bring the best solutions to our customers. That's one of the ways in which we will continue to accelerate our growth and remain successful in this highly competitive marketplace."[emphasis added]*

At the heart of the innovation system at Rockwell Collins is the ability to have innovation be connected to the businesses, but not be driven by the businesses. This approach is exemplified by the technology enabled growth strategy adopted by the Advanced Technology Centers at Rockwell Collins. The metric they use, called sales growth leverage (SGL), is defined as the ratio of value expected by the businesses from the technologies developed by the advanced technology centers (ATC) looking out five years, over the amount spent in total by the centers on developing technologies looking back five years (Berger, 2007). When the metric was first used at Rockwell Collins the ratio was 8:1, and by 2007 had increased to 40:1. The importance was not in accuracy of the metric itself, but rather in the agreement it

fostered between the businesses and the internal R&D team on the value of innovation, and in the general trend that demonstrated the effectiveness of the team.

Source: AWST Top Performing Companies Data for Rockwell Collins and Lockheed Martin for Internal R&D Spending



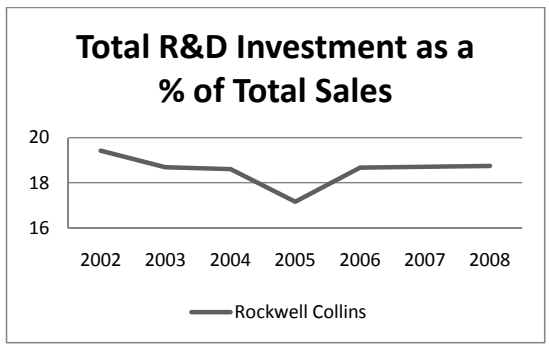


Figure 4-2 R&D Spending Analysis at Rockwell Collins

4.3.2 Funding Innovation

Innovation by its very nature is a hit-and-miss opportunity, and yet, Rockwell Collins has consistently invested between 18 – 20% of gross sales on R&D, a figure comparable to a high-technology consumer electronics company rather than an aerospace and defense company. Even accounting for the fact that about half of that expenditure is customer sponsored R&D, their R&D spending (as a percentage of revenue), far exceeds that of its peers in the aerospace and defense sector. Take for instance the R&D spending at Lockheed

Martin (LM) and Rockwell Collins (RC) using internal funds as shown in Figure 4.2 (top graph), Rockwell Collins outspends Lockheed significantly. It is important to note here that even through the various crises that the US market has faced, Rockwell Collins has maintained the funding level, with Clay Jones their CEO emphasizing that innovation is key to the organizations ability to deliver sustainable value to their shareholders (Jones, 2008).

One of the attributes of their R&D spending is that almost half is customer funded, as seen in the total R&D spending as a % of total sales in Figure 4.2 (lower graph). A good example of customer funded R&D is their Synthetic Vision Information Systems (SVIS) program, that was initiated with funding from the Air Force Research Laboratory, and then later matured with funded from the NASA Langley Research Center for Aviation Safety. The evolution of a project can be reconstructed by starting with a description of the concept (Etherington, Vogl, Lapis, Razo, Inc & Rapids, 2000; Theunissen, Rademaker & Etherington, 2001), followed by the proof-of-concept demonstrations and flight tests (Theunissen, Rademaker, Vogl & Postnikov, 2001), to the incorporation of elements of the system into the ProLine Fusion line of avionics products. The importance of the technology is further highlighted by Rockwell Collins winning the Prometheus Award from the Technology Association of Iowa (TAI) as the *2006 Innovator of the Year*.

4.3.3 Motivating and Driving Innovation

One of the challenges of having a strong innovation history, and successful products, is that organization potentially gets trapped in the approaches that it uses to tackle problems. The 10X program was designed to specifically foster breakthrough thinking on the part of their employees (Jones, 2006). The program was started in 2004 to encourage their employees to submit high-risk ideas that would result in at least an order of magnitude (10 times) improvement in at least one outcome of three parameters: cost, size or power requirements. Each accepted idea would be funded with $50,000 for proof-of-concept work, and once proven viable would join the traditional R&D pipeline. In addition to the recognition from their senior leadership, the engineer that proposes the concept is allowed to spend the next year developing the proof of concept. In 2004, the program was initiated with a budget of US$ 500,000 and resulted in the generation of 46 ideas, 8 of which were funded. In 2005, funding was increased to US$ 1 million with greater participation from other parts of the organization, leading to more than 70 ideas. In the third year, the program attracted more than 180 submissions, leading to it being institutionalized as an annual enterprise-wide effort with engineers participating from across all parts of the enterprise.

The other component to the Rockwell Collins innovation pipeline is the increased focus on technology scouting and open innovation. They established a technology area council to track emerging trends and technologies that could be applied to create immediate value. Rockwell Collins uses rewards based on financial incentives and peer recognition to support and drive innovation (Rockwell Collins, 2009b). These include the annual “Corporate Engineer of the Year” award and the quarterly “Open Innovator Award”. From all of the peer nominations for the Corporate Engineer of the Year award (Rockwell Collins, 2009c), 50 are selected as semi-finalists, and 10 as finalists. Of the finalists, three are chosen as winners. The open innovator

award was instituted to foster a cultural change within the organization to support open innovation. Since the program was first introduced in 2005, it has yielded benefits in areas ranging from cognitive radio (using Adapt4 technology) to global positioning systems (Wojciechowski, 2007).

4.4 Developing a System of Innovation

The innovation puzzle is not easily cracked by an organization using just one of the three approaches discussed in previous subsections. The current operational environment requires organizations to understand the strengths and weaknesses of each approach and craft a system of innovation that maximizes the strengths and minimizes the weaknesses.

Table 4-4 Analyzing the Strengths and Weaknesses of the three Innovation Approaches

Approach	Strengths	Weaknesses
Internal R&D	Knowledge sharing and transfer	Potentially developing core rigidities; Metrics
Technology Scanning	Tap into the larger community within an industry or domain	Managing large idea volume; Most alliances/acquisitions do not add value
Open Innovation	Tap into the global community both within and across industry or domain	Requires in-house capabilities to build the product/service; managing culture change;

In the endgame, the organization-specific system of innovation should contain elements of each of the three strategies. Through an analysis of the published literature and case studies of organizational innovation, and our own study of Rockwell Collins, we propose that in the ideal case an organization builds a system of innovation that is built around a shared value proposition between the internal innovation team and the rest of the firm; supported by a set of incentives and rewards that creates a culture that makes innovation everyone's responsibility; and last but not least, creating an infrastructure that supports idea capture, assessment, monitoring and knowledge management, as shown in Figure 4.3.

4.4.1 Shared Value Proposition

Consider the effectiveness of the average R&D department across ten key processes as assessed by Szakonyi (1994). With the exception of selecting R&D projects, and actually managing them, the average R&D department either underperformed (generating new product ideas, maintaining the quality of R&D processes and methods, coordinating between R&D and Marketing, transferring technology to Manufacturing), was only beginning to tackle (motivating technical people, establishing cross disciplinary teams, and linking R&D

to business planning) or performed dramatically poorly (long range planning). Six of the eight processes relate to establishing a shared value proposition with other stakeholders in the organization. For instance agreement between senior leadership and the innovation team regarding the success criteria ensures sustained funding for research efforts. Similarly, agreements with other functional areas both ensure ease of transfer of knowledge between the various functional areas, as well as participation in multi-disciplinary teams. When success criteria for the internal team are defined, then approaches such as technology scanning and open innovation can be leveraged to maximize the benefits for the organization.

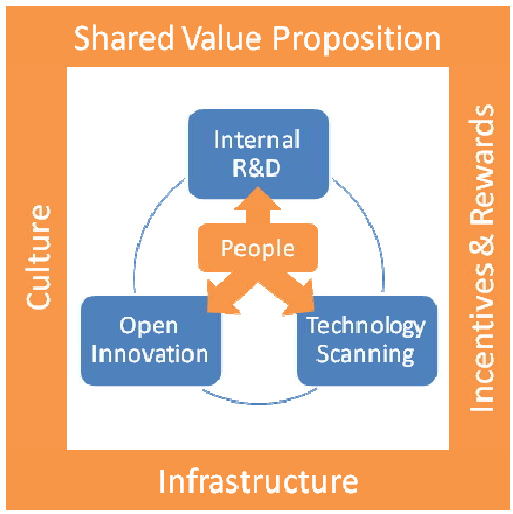


Figure 4-3 Creating a Strategic System of Innovation

In the case of Rockwell Collins, we saw that there was a clear mandate across the organization that innovation was a key to their long term success. In addition to statements to that effect, the organization spent a significant portion of their gross sales on R&D efforts. It is important to note that investing 18-20% of gross sales in R&D is comparable to the spending at a high-technology consumer electronics firm such as Nokia (Hira & Rosa, 2007). Another illustration of the shared value proposition between the innovation team and the rest of the organization is the use of "Sales Growth Leverage" as a metric for measuring the effectiveness of the organizations investment in R&D. While that metric is no longer currently being used, Rockwell Collins recognizes the importance of having a metric that both points to the general effectiveness of R&D, and also serves as a rallying point around which multiple groups can coalesce.

4.4.2 Culture, Rewards and Incentives

The issue of culture, and incentives and rewards is, to a large extent, intertwined. When the quality movement was first launched, the commonly used phrase was 'quality is everyone's problem'. A couple of decades later, now that quality has become embedded in most successful organizations, the next lever that will provide competitive advantage is innovation. It is not just the responsibility of the in-house R&D team to spur innovation, it is the responsibility of each and every member of the organization because they can spot opportunities and trends that are related to their own work, and through that, add increased value to the organization. An important enabler to building a culture of innovation and entrepreneurship is creating the right sets of rewards and incentives for members of the organization. As Hoskisson, Hitt and Hill noted in their analysis of 184 major US firms, managerial incentives that were based on a focus on short-term financial performance, were negatively correlated with R&D intensity (Hoskisson, Hitt & Hill, 1993). In reflecting on the state of the pharmaceutical industry, Garnier (2008) pointed out that a culture of 'wanting to make a difference' was needed, and GlaxoSmithKline had revamped their incentives in R&D - for example, by rewarding scientists (through bonuses) for their work, and not for the achievements of the organization as a whole. When it comes to demonstrating excellence, people in the organization are looking for rewards and recognition from their peers and from the organization.

Rockwell Collins embraces innovation as being one of the core elements of its brand identity, and with that, laid the foundation for embedding innovation into the cultural fabric of the organization. Even though there is a rich history of innovation going back to the roots of the company, Rockwell Collins had to establish its own identity after being spun off from Rockwell International in 2001. The rewards and recognition programs were designed to support incremental improvement (through their Lean Electronics™ program), out-of-the-box thinking (through the 10X program), and the "open-innovator award" and "engineer of the year" awards. In addition to the recognition provided by these awards, having one's 10X idea funded meant being able to devote a significant portion of time to prove that the concept was in fact feasible.

4.4.3 Infrastructure

The importance of having the necessary infrastructure to support idea capture, idea selection, and the monitoring of the idea as it transitions into becoming a finished product or service, is foundation to driving innovation. Consider the case of Nortel Networks (Massey, Montoya-Weiss & Tony, 2002), wherein the four-phase approach of idea qualification, concept development, concept rating, and concept assessment was supported through the Galileo process and the Virtual Mentor tool. They found that having a defined process and understanding people are prerequisites to developing technology, but without the right

tools/technology the project would not have been successful. Another good example that highlights the importance of infrastructure is the 'Innovation Jams' conducted annually by IBM. When it was first started in 2006, 150,000 participants used bulletin boards and the web to present, discuss, and debate technologies under development at IBM (Bjelland & Wood, 2008). In the second phase of the innovation jam, the 46,000 ideas that were generated were refined into a set of 36 ideas (using a combination of text analysis software and expert reviews), ultimately leading to the identification of the top 10 ideas that could be funded (Collins, 2007).The infrastructure was critical to enabling the entire organization to participate in the process, and in enabling the filtering of the ideas into an actionable set. While there is little publicly written about their innovation infrastructure, Rockwell Collins makes extensive use of communities of practice and learning networks to enable idea generation and selection.

4.4.4 The "Who" and the "How"

At the heart of the system of innovation is being able to determine the "who" and "how" of the innovation process. People are the source of innovation for any organization. For instance, Stoker, Looise et al. (2001) develop a model that connects leadership, individual and team characteristics to more effective R&D outcomes. In a similar vein, Stevens and Swogger (Stevens & Swogger, 2009a, b) describe a five step process adopted at the Dow Chemical Company which focused on ensuring that the right people were driving innovation, starting with creative leadership, finding the right mix of starters and finishers in leadership, finding appropriate rainmakers in middle management, and finishers for doing work. Leadership has a critical role in creating an environment where innovation can flourish. In Rockwell Collins, the senior leadership team is vocal in their support for innovation as seen in the comments from their CEO Clay Jones, and their senior VP of Engineering Nan Mattai. An equally important aspect is selecting and growing the right personnel to drive innovation within the organization. At Rockwell Collins, career trajectories have been developed that enable people to stay and grow within a technical role, and hence, take on greater responsibilities to drive, support and manage innovation efforts.

A system of innovation is not really a system until the "How" aspects are clearly understood. As an idea evolves into a finished product or service, the three mechanisms of internal R&D, technology scanning, and open innovation are used in varying combinations. Technology Scanning and Open Innovation are used dominantly in the idea generation and/or prototyping phase, while internal R&D provides the in-house expertise for productizing the idea. In the case of Rockwell Collins, they have both a traditional R&D infrastructure that covers basic, applied and product research, and at the same time, they embrace technology scanning through their technology councils, and open innovation both as a means of concept generation (Rockwell Collins, 2007), as well for creating additional value through modification, i.e. using open innovation as a 'Mod Station' (Nambisan & Sawhney, 2008).

4.5 Micro Perspective on Innovation

The discussion in the previous sections focused dominantly on the strategic aspects of creating a system of innovation. Even when the organization can generate great ideas that have the potential to be "game changers", they do not always do so. What needs to be discussed at that point is the micro perspective of why transitioning ideas into engineered products and services is difficult. This issue was first brought up in our discussions with the head of software engineering at SpaceCo, a boutique aerospace company focused on algorithm development. When we were discussing the challenges of innovating in the software context, she pointed out:

> *"When we write software to demo something, it is a wadded ball of graduate student duct tape. From the perspective of software engineering, it is a mess, but from an innovation standpoint, it is a beautiful thing!"*

When we followed up the comment, discussing the process used for development, she noted:

> *'Our people play in their sandbox, and they breadboard stuff, build some MATLAB™ models, and patch it all together to create a demo – this is not engineered – for the engineering the system we need sufficient scoping and documentation'*

On discussing the impact of having this kind of innovative development on the organization, she explained:

> *"Often, we have demoed code, and had the customer want to incorporate it into a product, however, when they are informed of the cost of engineering the software, i.e., making it robust for operational conditions, they go.. isn't what you just showed us software?... and the product goes into the next contract or project bid"*

When we followed up two years later with SpaceCo, the head software engineering had changed, but the problems still remained the same. A new element in the discussion was the challenges associated with introducing process improvement, as the new head said,

> *"We cannot apply the same processes as we use in our production environment, yet, we need to have some level of process maturity. Our attempt to introduce CMMI in our organization has to account for the variation in the classes of projects that people work on"*

These two interviews at SpaceCo prompted us to further our understanding on how software organizations transition ideas into production. Interestingly enough, there has been limited research in understanding how software organizations innovate, making the development of

the framework both a useful tool for practitioners who are trying to manage innovation in the software context, but also as a platform for launching future research.

4.5.1 Framework Development

In order to understand the problem of transitioning ideas into production, we have to address the problem at two levels: the planning level and the execution level. These levels were chosen from the perspective of the impact that they have on the organization, as well as to indicate which group of stakeholders would be best able to influence the dimensions clustered under them.

Table 4-5 Dimensions to Consider to support Innovation Planning

Focus	Dimension	Idea Phase	Production Phase
Planning	Nature of Innovation	Radical/Modular/Architectural	Incremental
	Expectation	Curiosity/ WOW! Factor	System Delivers as Expected
	Responsibility, Authority, Accountability (RAA)	Team	Organization
	Human Capital Dependency	Highly complementary	Fungible

From a planning perspective, the four key dimensions are the nature of innovation; the expectations from that phase; the human capital requirements; and the delegation of responsibility, authority & accountability (RAA); as shown in Table 4.5. When the idea represents basic or applied research, the nature of innovation is radical, modular or architectural. Radical innovations, by their very name, are game changers that have not been conceived of, and effectively executed on. Modular innovations involve the changing of the core design concepts of the technology, while architectural innovation forces a change in the product architecture without necessarily changing the underlying design concept or components (Henderson & Clark, 1990). While the inclusion of modular innovation may be surprising to some, the rapid pace of evolution of software technologies makes it a key part of the idea generation phase. In all three types of innovations in the idea phase, a software organization has to develop new capabilities. In the production phase, innovations are at best incremental.

The expectation from the idea generation phase is driven more by the curiosity/Wow factor, and a potential for success rather than a homerun. For instance, in the consumer goods industry, the success rate associated with the conversion of an idea into production is 15-20% (Lafley, 2009). On the other hand, once an idea has been successfully validated, the

expectation when it goes into production is that the product works every time, whether that is in terms of capability delivered by the system or in terms of the expected sales etc. One of the factors that was identified in our earlier work on successful technology adoption was the ability of the team to have the requisite responsibility, authority and accountability (Hines & Srinivasan, 2005). In the case of idea generation, the responsibility is scoped to that of the team working on the project, and to some extent to the larger innovation team. When a project transitions to the production phase, then the expectations of value delivery are higher, and consequently the RAA factor changes to an organizational scope. It is important to emphasize here that all three aspects of responsibility, authority and accountability must be present for successes both in the individual phases, but also in the transition phase. From a software organization perspective, this involves balancing team autonomy, customer access, and strategic alignment.

Table 4-6 Dimensions to consider for supporting Innovation execution

Focus	Dimension	Idea Phase	Production Phase
Execution	Problem	Wicked	Complex/Complicated
	Team Size	Small	Large
	Search Strategy	Exploration	Exploitation
	Knowledge Management	Tacit	Explicit
	Process	Fluid/Defined	Rigid
	Capabilities	Dynamic	Routines

From an execution perspective, the six dimensions that need to be accounted for are the nature of the problem being tackled; the size of the team; the search strategy used by the team to solve the problem; the associated knowledge management strategy, the process used by the team; and the resultant capabilities leveraged. Rittel and Webber observed that there are two broad classes of problems: tame and wicked. Tame problems are those in which the mission is clear, and it is easy to determine if a solution exists. On the other hand, wicked problems are characterized by unclear missions, and there is no easy way to determine whether or not a solution truly exists (Rittel & Webber, 1973). The distinguishing characteristics of wicked problems that they identify are equally applicable to software (for a more detailed discussion see (Srinivasan & Lundqvist, 2005)). In the idea generation phase, problems are wicked, but by the time they are ready to be transitioned into the production phase, they are either

complex or complicated. By complex, we mean that the problem is solvable with a combination of one or more techniques that are already known to the team, however, that solution has not yet been identified. Complicated problems on the other hand, already have a solution that can be applied directly. The nature of the problem strongly influences all the other dimensions of execution that need to be considered. When considering innovation, it is important to highlight the fact that idea generation is often done by an individual or a small team, and that the development of an engineered product or service often involves a large team. This impacts how the team searches for solutions, manages its knowledge, the process it uses, and last but not least the capabilities developed in the organization. The concept of exploration versus exploitation in organizational learning was first proposed by March (1991) and further refined by Levinthal and March (1993). The construct has now become commonly used in both the learning as well as the innovation literature. In the idea phase, since the problem itself is not well defined, the team has to use exploration as its search strategy, and creates new reservoirs of knowledge as the exploration progresses. Once the production stage is entered, the team exploits the knowledge that was generated in the idea phase. It is important to note that the search strategies are dominantly exploration or exploitation depending on the phase, but that does not exclude exploration from happening in the production phase or exploitation in the idea phase. In fact, we believe that for innovation to be truly successful, the organization has to be ambidextrous (Tushman & O'Reilly III, 2006).

As the organization carries out exploration using a small team, the knowledge generated remains largely tacit until the team has reduced the search space to a manageable set of options. The tacit knowledge generated is easy to share because the team size is conducive to support socializing the knowledge without having to codify it. On the other hand, as the team size grows larger, and the solution space narrows, the knowledge management strategy focuses more on managing explicit knowledge. In the case of software organizations this requires both conventional knowledge management systems, but also collaborative spaces and shared code repositories. An exploration driven process has to necessarily be fluid to support modification, but more often than not, is broadly defined by the nature of the problem. As the production phase is entered, the process becomes more rigid to maximize the exploitation of existing knowledge, and enforce creation and dissemination of explicit knowledge (through documents and other artifacts). In the end-game, the idea phase requires the creation of dynamic capabilities (Teece et al., 1997), while the production phase emphasizes the utilization of routines (Winter, 1995).

4.5.2 GameDevCo

In the previous chapter, we discussed GameDevCo in significant detail, mostly focusing on their agile adoption journey. We briefly mentioned that GameDevCo was running the sustainment of its current product Legacy_Game, and the development of the Next_Generation_Game in parallel, but were seemingly unsuccessful. The framework

discussed in the previous subsection provides a means of better understanding the reasons behind why it was not successful, as shown in Table 4-7 and Table 4.8.

Table 4-7 Innovation Planning at GameDevCo

Dimension	Next_Generation_Game	Legacy_Game
Nature of Innovation	Radical/Modular/ Architectural/Incremental	Incremental and Architectural
Expectation	Save the Product	Retain Customers
Responsibility, Authority, Accountability (RAA)	Organization	Team
Human Capital Dependency	Fungible	Fungible

Legacy_Game team members were constantly evolving the product, as is recommended in agile methods, as a means of ensuring that software could be maintained and enhanced easily. As one of the developers in Legacy_Game noted:

> *"The guys that created the mess are in Next_Generation_Game now! We have completely refactored the mess, and it is now more modular and elegant than Next_Generation_Game. We are delivering and improving Legacy_Game faster than they can catch up"*

Despite the expectation that the production phase would dominantly involve incremental innovation, the Legacy_Game team was also incorporating elements of architectural innovation. Next_Generation_Game on the other hand did not have a clear articulation of the nature of the innovation they were trying to incorporate. There were some elements that were almost purely incremental, some that were modular such as using Flash to program the user interface, some that were radical with respect to the core game engine architecture and some that were architectural, such as the payment engine creation. Some of those decisions were made by the chief architect based on his understanding of the product, and some were driven by customer/organizational needs. The impact of this lack of clarity on the nature of the innovation resulted in two launch date delays, and a product that was yet to demonstrate its capabilities for the organization.

From the expectations perspective, the characteristics are completely reversed. Next_Generation_Game was initiated in 2005 because Legacy_Game (circa 2005) was perceived to be difficult to maintain and evolve. At the same time, Legacy_Game was also

the dominant source of revenue in the organization, as a result, the mandate was to retain customers, not necessarily to grow the market. Next_Generation_Game on the other hand was expected to "save the company" and yet became a product that almost failed, forcing senior leadership to suspend development work on Legacy_Game (currently there are only two people working on bug fixes and maintenance patches) and move all the resources into Next_Generation_Game, essentially making it a production project even though it was not mature enough to be transitioned. Since Next_Generation_Game was seen as the future flagship project, the RAA factor was to the organization as a whole (since long term survivability of the organization was dependent on the project), while Legacy_Game became more insular and RAA focus was on the team from an improvement perspective.

The increased dissatisfaction in Legacy_Game and the seemingly increased focus on productization led to a large number of people leaving GameDevCo. This was in part driven by an assumption that key elements of their human capital were interchangeable, though in reality that was not the case. Consider the discussion between the head of game development at GameDevCo and a former project manager at Legacy_Game:

> *Legacy_Game_PM: We know that this problem has existed for almost two years and it is really costing us in terms of customer loss on a daily basis. One of my guys told me that we know the root cause of the problem, and that he needs two days to fix it.*
>
> *GD_BU_Head: We tried to fix this problem last year as well, and we dedicated a week of development time to finding and fixing the problem – and nothing happened*
>
> *Legacy_Game_PM: Oh, but my guy said that it would only take a day to fix, but we need system verification time*
>
> *GD_BU_Head: We dedicated that resource last time as well, but we didn't find anything – right now, we are hoping to get the Next_Generation_Game out with the problem solved.*

Among the reasons for the inability to find and fix problems were that the knowledge on why the problem was occurring was not available, and more importantly, the design knowledge was lost because the person who owned it had left the organization.

In looking at the dimensions relating to execution, the only dimension where Next_Generation_Game deviates significantly from the norm is in team size. Since there were multiple changes in technology and overall product architecture, the problem was in fact a wicked problem. It is only recently (Srinivasan & Lundqvist, 2009c) that the first business stories for the product were written, and the architecture stabilized and codified. The large team size was motivated in part by the aggressive release schedule that was originally planned, and the lack of clarity on the nature of the innovation being pursued. Even though exploration was expected, the loss of their domain experts, and the absence of both tacit and

codified knowledge from Legacy_Game slowed down the process dramatically. As a former Legacy_Game project manager noted:

> *"Initially, players had multiple mechanisms that they could use to achieve the same outcome – right now, we have standardized to this single mechanism and it has a bug – Apparently, we don't have the design knowledge that we need to effectively fix the problem."*

Table 4-8 Execution Enablers at GameDevCo

Dimension	Next_Generation_Game	Legacy_Game
Problem	Wicked	Complex/Complicated
Team Size	Large	Large
Search Strategy	Exploration	Exploration/Exploitation
Knowledge Management	Tacit	Tacit/Explicit
Process	Chaos → Stabilizing	Chaos → Stabilizing
Capabilities	Dynamic	Dynamic & Routines

One of the structural missteps that GameDevCo's senior leadership team made was that enforcing of a rigid separation between the two teams. As a result there was no incentive for the Legacy_Game team to share knowledge with the other team, and even worse, it triggered a "features war" that forced Next_Generation_Game to constantly evolve the features they were working on, just to keep up with Legacy_Game. To their credit, the senior leadership team at GameDevCo recently made a decision to create a single team that focused on Next_Generation_Game, thereby fostering increased tacit knowledge transfer across the teams. The adoption of Scrum as the development approach of choice is slowly moving towards stabilization and with it, the early chaos is receding. The impact of the process instability at GameDevCo goes beyond just the innovation perspective to the quality of organizational life as a whole. Unlike our expected behavior since Legacy_Game was continuously evolving the product, they were developing both routines and dynamic capabilities. One of the struggles with the Next_Generation_Game was the seeming assumption that just because the architecture was different, the competencies would necessarily be different, and the underlying routines had to be reinvented.

4.6 Conclusion

Innovation is a key for organizations to remain sustainable in today's turbulent environment. This is even more important for software organizations whose competitive advantage is built around being able to carry out knowledge work better than their competitors. Yet, there is little in the academic literature that connects the strategic and tactical perspectives on organizational innovation for software organizations.

In this chapter, we take a different approach for data collection, in that we rely on secondary data in the form of Annual reports, newspaper and magazine articles to construct the strategic framework for understanding how organizations architect their system of innovation. The data for the case study of Rockwell Collins can be found in the public domain, and most articles (other than those published by professional societies such as the IEEE, SAE, and ACM) are available on request. When transitioning from the strategic framework to the more tactical framework in the second half of the chapter, we rely on the academic literature for developing the theoretical framework and obtaining first pass validation from two expert interviews from SpaceCo. These interviews, along with feedback from senior leadership at GameDevCo, IndiaCo and EuroTel (where the framework was presented) gave us increased confidence in applying the framework to understand the challenges faced at GameDevCo with Legacy_Game and Next_Generation_Game. We recoded the instances where our interviewees discussed the tension between Legacy_Game and Next_Generation_Game using the attributes in the framework. This classification is not meant to be exhaustive or quantitative, but rather to act as a starting point for further research in applying the framework or falsifying the elements that need to be considered.

In this chapter we presented the macro and micro perspectives on innovation. We first developed a framework for understanding the macro perspective on innovation. We built the framework through a review of the literature, and an analysis of how Rockwell Collins developed their system of innovation. The framework developed highlights the need for every organization to built its own unique system of innovation that is centered around the people (since they are the primary sources of ideas), and enables them to leverage one or more of the three innovation strategies of internal R&D, technology scanning, and open innovation. However, this system of innovation would be ineffective in the absence of a shared value proposition on innovation in the organization, or a culture that is supported with the right mix of rewards and incentives to drive innovative behavior. Last but not least is the need for infrastructure both of the information technology variety (idea management, knowledge management), and the organization process variety (for example, hiring). The Rockwell Collins case provided us with an illustrative example of what a successful system of innovation would look like. The framework proposed provides senior leadership with guidance on the elements of an innovation strategy, and insights into the enablers that are necessary for successful innovation to occur.

Even when the organization does manage to put a system of innovation in place, the process of transitioning ideas into engineered products and services is not easy. It is in reality a chasm

that needs to be crossed to enable the organization to create a stream of successful and sustainable products and services. While the adoption of innovation in software organizations has been studied before (Zmud, 1984), the mechanics of actually creating a finished product from an idea had not been explicitly identified. Our articulation of the dimensions clustered across the planning and execution dimensions provides the factors that senior leadership should consider when managing and assessing their innovation portfolio. The case studies of SpaceCo and GameDevCo provide insights and validity within a specific context. The interrelationships between the dimensions in the micro perspective and the elements of a system of innovation at the macro level need more empirical work to be generalizable.

5 LEVERAGING GLOBALLY AVAILABLE CAPABILITIES

"Massive investments in infrastructure.. and cheap computing.. came together to create a platform where intellectual work, intellectual capital, could be delivered from anywhere. It could be disaggregated, delivered, distributed, produced, and put back together again-and this gave a whole new degree of freedom to the way we do work, especially work of an intellectual nature..."- Nandan Nilekani (Friedman, 2005) [Emphasis Added]

5.1 Introduction

The quote from Tom Friedman's book highlights the fluidity of knowledge work and the existence of pockets of expertise across the globe that organizations should tap into to gain a competitive advantage. Every organization today recognizes that it lives in a world where outsourcing and offshoring are the norm not the exception. While the traditional notions of software outsourcing for cost savings were true during and just after the dot-com bubble, the true driver for adopting these approaches is the availability and exploitability of capabilities. This chapter focuses on one instance of the larger problem of leveraging globally available capabilities by studying software maintenance in a product context. The research question that we answer in the chapter is:

How do we study the evolving customer supplier relationship in maintenance services outsourcing?

Our approach to answering the question was to use an embedded case design in which we independently studied each organization in the client-supplier relationship, and then compared across them to identify enablers and barriers in building a successful relationship. The overall road map for carrying out the research in this chapter is shown in Figure 5.1. Starting with senior leadership interviews at EuroTel, we conducted a workshop to revisit the EuroTel/IndiaCo relationship. We carried out focused case studies on both EuroTel and IndiaCo leading to knowledge sharing with top leadership team (TLT) of both organizations.

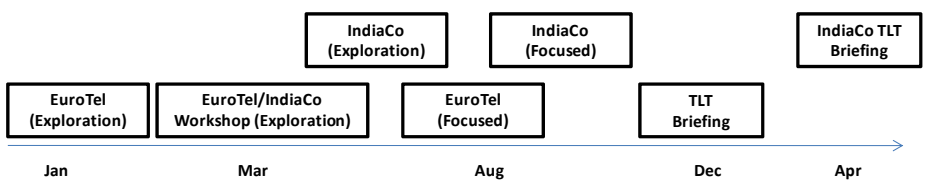


Figure 5-1 Research Roadmap

5.2 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, as shown in Figure 5.2. 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.

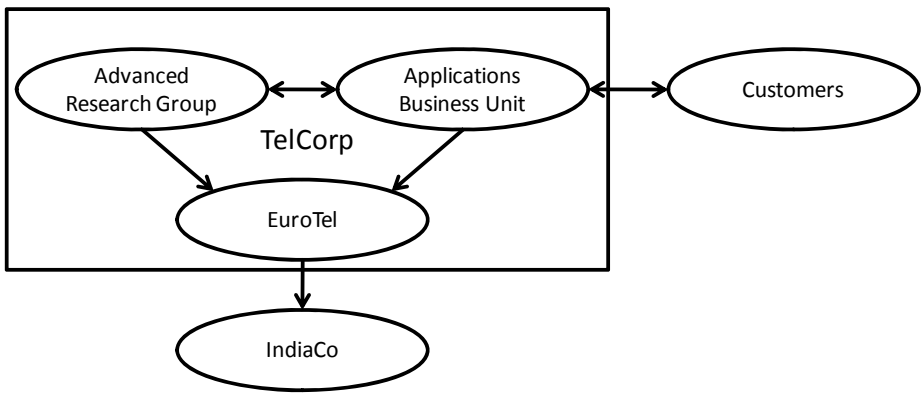


Figure 5-2 EuroTel Organization Structure

The focus of the case study is on EuroTel, as we cover our engagement strategy (Section 5.2.1), the preliminary round of fieldwork at EuroTel which yielded the organizational level challenges (section 5.2.2). The second round of fieldwork focused on project teams and yielded insights on the execution challenges (section 5.2.3).

5.2.1 Setting the Stage

The research used the approach of engagement, exploration, focused field work, and knowledge sharing, as discussed in Chapter 2. In the engagement phase, we contacted one of the senior managers at EuroTel to discuss how enterprises balanced agility and discipline, and to gather some more industry insights into whether the challenges were domain specific. In our first meeting, after having discussed both process improvement approaches and innovation, the senior manager pointed out that there was another way that organizations developing software systems gained agility – through the outsourcing of work to exploit cost differentials. 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 also some preliminary insights into the drivers underlying their evolving relationship with their outsourcing supplier - IndiaCo. In order to understand the dynamics of the relationship, and gain a deeper understanding of their strategic direction of the relationship going forward, 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). The findings from 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.

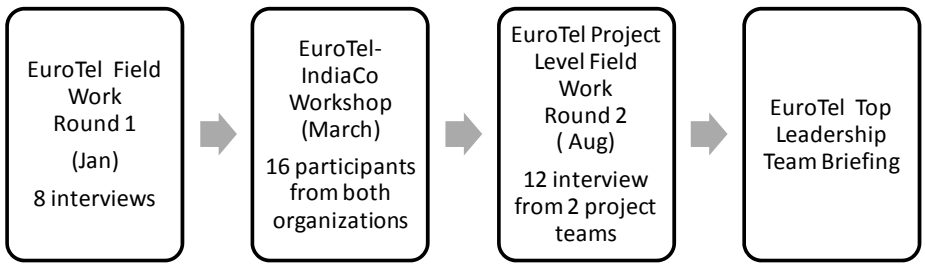


Figure 5-3 Evolution of the Research at EuroTel

5.2.2 Organizational Challenges

In the TelCorp enterprise, the end customers are the primary focus of value creation, with information flowing between the various business units as shown using directed arrows in Figure 5.1. While ARG and ABU have bidirectional information flows, ABU treats EuroTel as an internal supplier. What is even more damaging to EuroTel, is that they do not have direct access to the customer. Historically, EuroTel has had complete ownership of their product value stream, relying on internal expertise to create the core technology platforms, from the chip design and fabrication to software development. In 2001, TelCorp made a strategic decision to transition to commercial off-the-shelf (COTS) components, resulting in the slow marginalization of EuroTel. As the head of EuroTel noted:

> *"Right now, we are struggling as an organization to determine whether or not we should exist. From a corporate standpoint, we feel that our competitive advantage lies*

in being a systems integrator, and not in the development of the core technology. Either we turn this business unit around, or it is going to be split up and merged across the various product lines within ABU."

Yet, this situation in 2007 was an outcome of a series of events in the 2000s. In early 2001, a recession in the home market resulted in TelCorp severely downsizing its headquarters by almost fifty percent. This massive reduction in manpower resulted in the organization switching from its traditional growth focus to a 'save the company' mode. EuroTel was faced with an even greater shortage, as key personnel had either been fired or retired, with those leaving taking with them a significant amount of tacit knowledge. The resultant culture at EuroTel (post massive layoffs and a seeming erosion of core competencies in the early 2000s) was one of risk avoidance.

As a senior manager observed:

"A lot of managers now would rather not take a decision, and wait for instructions from higher levels – this is killing the innovation that was the core of the company. Yes, they are able to execute effectively when given orders, but they are not creating new value"

An alternative perspective was provided by a veteran manager, who is transitioning from EuroTel to the ABU. He said:

"In our company, you could always start something new. If you had the drive and could make the case for why your idea was important, management supported it. Now, we still have people that have the ideas and the drive, but do not necessarily have the support."

Part of this perceived gap in creating new products can be attributed to the fact that EuroTel is not close to the customer anymore. The relationship between ABU and EuroTel is an arms-length customer-supplier relationship, with the ABU generating requirements and handing it over to EuroTel for development. Even though they are part of the same organization, ABU does not share customer requirements or product schedules with EuroTel, creating significant tensions between the two business units. As one EuroTel product manager explained:

"We initially agreed to deliver the platform to ABU in July. Even though the requirements have changed dramatically, my team has been working round the clock to deliver the system on schedule. I just got a call from the project manager from ABU, telling me that they want the system delivered two months earlier – there is just no way, that we can deliver a quality product"

The three areas of concern that emerged from the analysis and severely impacted EuroTel's ability to evolve in an environment of constant change were:

- the strategic uncertainty about the existence of EuroTel,
- the risk-averse culture of middle-management,
- the strained relationship between EuroTel and ABU.

5.2.3 Execution Challenges

One of the challenges that EuroTel currently faces is with respect to project execution. The risk-averse culture of middle managers has led to the organization focusing on on-time project delivery as the dominant driver in determining project success. Historically, EuroTel has used a disciplined, plan-driven approach to software development. They followed a stage-gate approach (shown in Figure 5.4), with requirements captured in a system plan as the starting point. This system plan is refined using a pre-study that generates the road map for execution (in terms of software versions) and a set of implementation proposals for the actual execution of the projects. These implementation proposals include the estimated resource consumption measured as a factor of technical hours needed to execute the project, and the available team competencies. On reviewing the findings of the pre-study, the project moves to a feasibility study that determines whether or not the project moves forward into the execution phase. At any one of these gates, the project can be killed or sent back to a previous stage for more work.

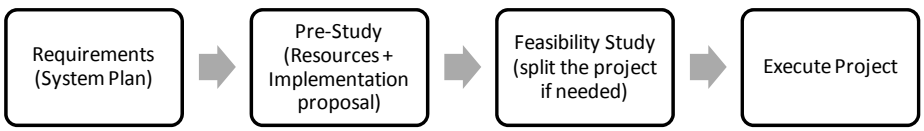


Figure 5-4 Project Planning at EuroTel

This discipline, however, has eroded over the last eight years, leading to the creation of two new improvement efforts focusing on managing version heterogeneity and providing visibility into the development process. The first improvement program focused on the need to manage version heterogeneity of their software its associated impact, – that of varying versions being concurrently worked on by different teams. Adding to the complexity of the problem is the fact that software runs on remote hardware that is owned by the clients, any trouble reports that come in have to be addressed in near real-time to ensure continued delivery of service – leading to multiple versions of the software being run at various client sites. This heterogeneity in software makes it extremely hard for EuroTel to maintain a common baseline which they can manage more effectively. The need to have greater visibility into the quality of the developed software, and to assess progress towards project goals, led to the creation of the second program. The notion of daily build and daily tests are critical to enabling the success of this effort. As one of the project mangers pointed out:

"Right now we are in the process of getting the tools in place to support daily build and daily test – given that we were the first people to do automated testing (even before it became the industry standard), it is frustrating that we don't have the right tools to do it now – we knew how to do it without all of the modern tools before, but we seem to have forgotten how to do it"

During our interviews in the second round of fieldwork, we asked the interviewees to list the characteristics of a successful project. We weighted their inputs with 5 for the most important, 3 for the second important, 2 for medium, and 1 for low. The weighted importance, as well as the count on the number of times the factor was mentioned, is shown in Table 5.1. The top five characteristics that emerged were: on-time delivery, meeting requirements, good quality, satisfied customer and compliance to the project plan.

Even though 'satisfying requirements' was mentioned less times it was ranked as the most important characteristic three times. Quality on the other hand, was not the most important factor in any of the interviews. Probing this definition further, we asked interviewees to list what they believed were the factors that enabled project success. Using the same approach for ranking and weighting, the top five factors that emerged were Schedule, Clear Understanding of Goals, Team Spirit, Resource Availability, and Team Competence, as shown in Table 5.2.

Table 5-1 Characteristics of Project Success

Characteristics of Successful Projects	# of interviewees	Weighted Importance
On Time Delivery	9	45
Meeting all the requirements	6	22
Good Quality	8	19
Happy/ Satisfied Customer	2	6
Compliance to Project Plan	2	4

Table 5-2 Factors Affecting Project Success

Factors affecting Project Success	# of interviewees	Weighted Importance
Schedule	5	22
Clear Understanding of Goals	4	16
Team Dynamics: Spirit, Cooperation and Small Size	4	14
Resource Availability	3	12
Team Competence	4	10

Each of these factors reflects the organizational level tensions that EuroTel is facing today. Two of the root causes are worth exploring further: Resource Constraints, and Scope & Schedule pressures.

Resource Constraints

EuroTel is still recovering from the impact of the recession, as one of the developers noted:

> *"Budget is tight as there not many employees. Hardware is good and easy to get. But the budget for hiring people is tight and it affects us. We need more people."*

From a manager's standpoint, the budget constraints have significantly impacted both their ability to plan for the future, as well as being unable to manage human capital losses. As one of them explained:

> *"Planning for each year is hard to do as you can't know in the beginning how much you can afford to do and we don't have time to do internal improvements in working processes and tools that we do ourselves as we would like to do"*

Since they are unable to cross train people, they are susceptible to single point failures. If someone assigned to a project is sick or in the worst case scenario, leaves the organization, then the project is hamstrung. Even if they are able to hire someone to replace the person who has left, they have to account for the learning curve that each individual has to go through. In the cases when sufficient project documentation does not exist, the project is poised for failure. The managers have tried to leverage this budget crisis by introducing more automation to the process:

> *"We try to do everything as efficiently and low cost as possible. For e.g., we have automated all testing. As we can't afford to have testers, all designers do their own testing with automated system"*

One of the outcomes of the resource constraints is that they have had to have people simultaneously working on multiple projects. As one of the developers pointed out:

> *"In my last project, it was just one project and no other projects or maintenance work. But usually you have a few person in project who are involved in some other projects and say they can't work on your project for 3 weeks now, as they have to work on some other project which is more important - it is hard to plan for people who come and go in the project"*

Furthermore people are used in mixed-mode fashion – they do new development work, while at the same time supporting maintenance activities on a product. The maintenance efforts could involve either an earlier version of the product that the developer had worked on

previously, or a totally different product. Often the maintenance activities would trump the development activities. As one of the team leaders noted:

> *"For the team we have a time plan when we have to deliver, and team members don't work 100% on projects. They are allocated like 70% to project and they have to work on other project like maintenance projects or Trouble Reports. So if an important Customer Trouble Reports comes during the project and designer has to work on it now, it influence and conflicts the project time schedule."*

The impact on the developers is even greater, as one of them said:

> *"You typically work 50-50 in two projects, but it gets hard, as expectations are like 100% or 75% time to be given to both projects. Currently, I am working as part-time System Administrator and Coder. There is a conflict between these two as it is difficult to switch between the two, especially in cases where a lot of thought has to be given in some problems with coding."*

Scope & Schedule Pressures

Team members from both teams that we interviewed had faced significant pressures with respect to project scope and project schedule over their tenure at EuroTel. When talking about the coupling between scope and schedule pressures, one of the project managers noted:

> *"If we change the scope of project and also change time schedules, then it is not a bad decision and project can be successful. But if we change requirements and push, in the same time frame as before, and the same quality requirements then it is not successful as it will not be delivered in time and reasonable quality."*

When discussing the issues with schedule pressures, one of the designers noted:

> *"Time estimations are initially made in Implementation Proposal (IP) and when we start execution we should do time estimation again and not just accept the one from the IP as system management estimated it and it may be too low. The designers make more realistic time estimations – but it is not easy to go against Project leaders and Team leaders. In reality, there is pressure so we tell them that we can do this, but with limited functionality or you get these functionalities but then there will be lot of errors, and then a lot of trouble reports, and you get the product with a bad quality."*

There is a perception however on the perspective of how project managers/ team leaders see their role in mitigating the schedule compression. As one of the project managers pointed out:

> *"My role is to divide and assign the work among team members; also to find replacements or back up persons and that is hard when we are too few people. The team analyzes the requirements in their areas and provides inputs for planning on new functions – who will work on which requirement, how much time is needed and*

then as Manager, I decide what we can promise to customers. Team is responsible for development of core functionality in their team responsibility areas and also for testing."

As was seen in the section on strategic context, the uncertainty about EuroTel's role in the larger TelCorp context makes it really difficult for the managers and project leaders to control the schedule constraints imposed on the project. In an accurate illustration of Brooks' law (Brooks Jr, 1995), one of the developers noted:

"If we have not changed the scope and we have too tight a time schedule, then the project is not in good shape and is not ready on time. These cases take much longer time and we spend much more resources to get things ready in time. We try a lot of shortcuts to get results and it is not successful and we have big drawbacks because of the short cuts - it costs much more then it should, the product is of low quality and it is not ready in time."

An unintended side effect of working on projects with compressed schedules is the 'decision coma' that it entails, as one team leader points out,

"In projects like these, you typically end up with a decision coma for the team member."

5.2.4 Learning from EuroTel

The challenges that EuroTel faces are both at the strategic and the operational levels. From a strategic perspective, EuroTel needs to have a clearer articulation of where they fit within the larger TelCorp value proposition. Since they are treated internally as a supplier, it hampers their ability to engage their own suppliers in a strategic partnership effectively (as seen in the subsequent section on their relationship with IndiaCo). Since the emphasis has shifted from having complete ownership of technology development to using COTS components, EuroTel is still in search of an identity, with respect to who and what they are. One of the challenges that they continue to face is the high uncertainty in their workforce, with respect to job security and job satisfaction. Their recent attempts to spur innovation within the organization indicate recognition of the problem, and can provide a potential solution. Yet, as their business unit head pointed out, they are on a very short lease – a "turn around or bust" scenario.

The constant scope & schedule pressures, combined with the lack of resources has resulted in a large number of projects entering into a churning cycle, as shown in Figure 5.5. From an execution perspective, they need to be able to manage the process heterogeneity within the enterprise, and provide some means of minimizing the constant pressure that their teams are under to deliver on time, under severe schedule and resource constraints. Their relationship with IndiaCo has mitigated some of the challenges, but has brought on additional communication and coordination challenges that they did not face earlier. The improvement

efforts, such as those focused on the development process, and effective version management, provide a means of achieving stability, a necessary condition to achieve sustainable improvement. There are still open questions with respect to their human capital management strategy – even though from a corporate perspective, they have had multiple knowledge management initiatives underway.

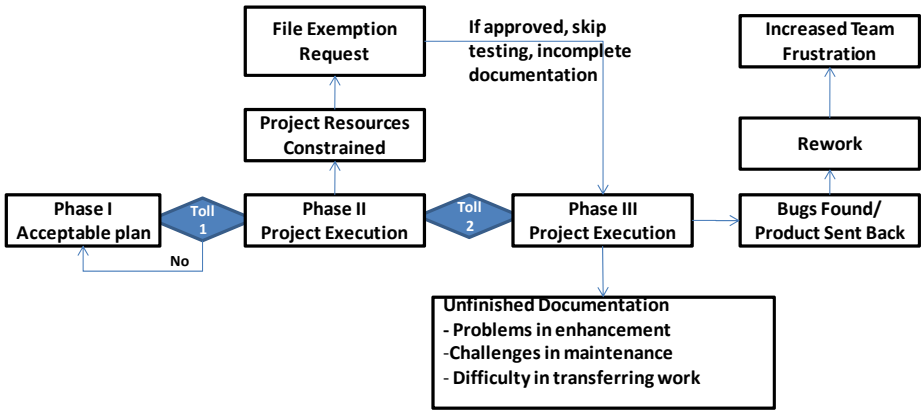


Figure 5-5 Project-Level Impact at EuroTel

When a project that is not in scope moves past the first project toll gate and enters development, the project team is already under significant pressure. Instead of completing the necessary verification and validation activities, the team files an exemption to move forward without ever passing through the second toll gate. Since the exemption is generally granted, the team moves forward with parallel development of the system, while it continues to undergo verification and validation. There are multiple unintended side effects – the most obvious one, is that the delivered product is flawed and requires rework. The impact is greater when it comes to product sustainment and evolution – incomplete documentation, incompatible/fragment mental models of the current team members (a lack of shared understanding of the product), make it very difficult to hand off the project to another team. Last but not least, this constant grind leads to increased team frustration. EuroTel has piloted the use of agile methods as a means of gaining better control over its development process, but has not been successful with it so far.

5.3 Growing the IndiaCo Relationship

Given that the reduction in manpower due to the recession in early 2000 did not result in an equal reduction on the total work that needed to be performed, EuroTel explored the possibility of creating a supplier base that would lower the cost footprint without impacting product quality. The organization that they tapped to take over the sustainment of one of their

products was IndiaCo. Since 2001, EuroTel has transitioned six products over to IndiaCo either in the form of complete product ownership or for just sustainment activities. Over the last year, the relationship between IndiaCo and EuroTel was slowly evolving to a state of maturity. While individual project managers in EuroTel vary vastly in their treatment of IndiaCo project managers, there is a strong sense of urgency within the senior leadership of EuroTel to evolve the relationship to a strategic partnership. This is apparent both in their transitioning of complete ownership of two products to IndiaCo, as well as in their joint participation in a workshop that was held at Mälardalen University as a means of developing a shared vision.

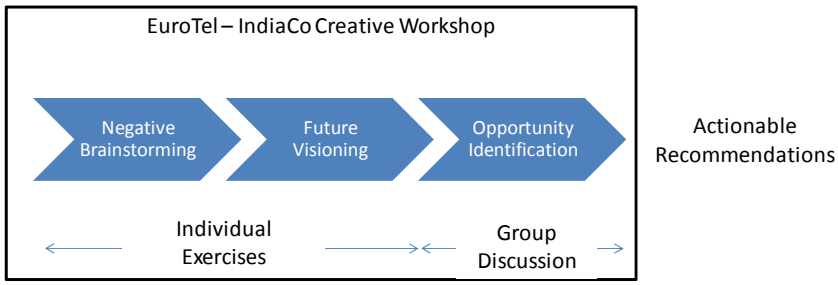


Figure 5-6 Creative Workshop Structure

The choice of workshop location was to provide a neutral setting, in which members from both EuroTel and IndiaCo could openly share current state challenges, and create a shared vision for the future. The workshop started with a set of lectures by faculty and researchers sharing best practices and cutting-edge research in the areas of software engineering, research and development management, and innovation, followed by a series of exercises that represent the basis for the findings presented in this chapter. The three exercises - negative brainstorming, future state visioning, and improvement opportunity identification - were carried out to develop actionable recommendations for improving the relationship between IndiaCo and EuroTel, as shown in Figure 5.6. The workshop exercises on negative visioning and future state visioning were designed by Annika Lofgren, as part of the innovation laboratory at MDH. In addition to building trust, the workshop provided rich data for understanding the relationship between EuroTel and IndiaCo.

5.3.1 Negative Visioning

The negative brainstorming exercise enabled the workshop participants to determine the key areas that would significantly degrade the relationship between EuroTel and IndiaCo. Each of the participants was asked to *identify his/her best ideas to make this cooperation as bad as possible*, as an individual exercise. We collected the data sheets and tabulated them to find common elements across the data. Although we did not explicitly distinguish between the personnel from the two organizations in this exercise, the data analysis, as well as the group

discussions showed that there was a strong overlap of viewpoints. There were five major grouping that emerged from the analysis:

- Overarching Strategy,
- Competence Management,
- Project Planning,
- Communication,
- Trust.

When the participants discussed the evolution of the IndiaCo-EuroTel relationship, there was a lack of agreement on whether the relationship was arm's length, contractual in nature, or if it was a trusted partnership. This arose from a lack of a clear vision of IndiaCo's role in the long-term strategy of EuroTel. As one attendee wrote:

> *"We can continue to outsource bits and pieces, with different, often conflicting strategies."*

This statement reflects the current state accurately, in that EuroTel has in so far highlighted the need to drive down costs and improve quality, but has not necessarily translated those short-term drivers to reflect long-term needs. IndiaCo, on the other hand, has demonstrated their ability to consistently drive down costs, but the monetary advantages of doing so are rapidly shrinking for them, as the EuroTel relationship represents only a minor portion of the total revenues that IndiaCo generates from its vast customer base.

One of the challenges that any organization outsourcing to the Indian market faces, is the high rate of attrition. As an IndiaCo member noted:

> *"one way to kill the relationship is to build teams with untrained people"*

This was a lesson that EuroTel learned through experience – one of their projects consistently had a high attrition rate, and was negatively impacting overall performance. Once the problem was identified, they were able to resuscitate the team to ensure that the attrition rate did not impact overall project performance. This issue of managing the competence of the IndiaCo talent base has become an area of focus within both organizations. IndiaCo recognizes the importance of delivering successful projects to grow the relationship. The notion of '*promising the moon and delivering peanuts'* recurred multiple times in the negative brainstorming. Ineffective project management, coupled with a lack of communication with the customer, were operational level factors that were pointed out in the negative brainstorming – the cues on communications included:

> *"Communicate only in the steering group meetings, don't provide any feedback or listen to feedback when provided, and hide bad news."*

When talking about creating an environment without trust, one of the EuroTel members said:

"Keep spying on them all the time, showing that we don't believe in them."

In a mirror comment, an IndiaCo member pointed out:

"Be policed all the time, and don't praise the partner, because they will raise the price."

The senior leadership teams of both organizations have worked very hard to build up an atmosphere of trust between the organizations, but there is still a long way to go before they truly become partners. For instance, recently, IndiaCo established a role, which we will call the 'safe communication channel'. In this role the IndiaCo staffer serves as a communication vehicle between the offshore team in India and senior EuroTel leadership. The need for this neutral broker emerged because mid-level managers at EuroTel were not communicating with their own senior leadership. As one of the IndiaCo project managers pointed out during our IndiaCo fieldwork,

"Their manager were telling us everything was fine, and at the same time, turning around and telling their senior leadership that the project was late because of us (IndiaCo)"

5.3.2 Future State Visioning

The future state visioning exercise was carried out using the template shown in Figure 5-7. This template was designed as part of the Innovation work at MdH, and served as a data gathering instrument. We collected fifteen usable inputs, and clustered the contents based on each of the four quadrants. There was broad consensus across the board that the future relationship would be characterized by:

- Accurate and Timely Communication,
- Establishing a Win-Win Trust-based relationship,
- Shared Product Ownership,
- Building Competencies and Spurring Innovation,
- and Effective Management.

Expanding on the notion of a win-win trust based relationship, one of the participants noted:

"'I am EuroTel not IndiaCo', I treat EuroTel's customers as my customers, rather than as a third party. Every activity is done to enhance value to customer and to make EuroTel successful – in the end we are their most trusted partner."

The underpinnings of that trust-based relationship are the ideas of joint product ownership, innovation and effective management. One of the side effects of the downsizing was in EuroTel's loss of domain knowledge – the people that remained at EuroTel were so busy trying to keep the organization working that innovation suffered. In order to stimulate innovation, EuroTel has to shift more than just the maintenance aspects to IndiaCo. The knowledge transfer process that takes place when IndiaCo takes on a project, ensures that they have sufficient domain knowledge to effectively take complete product ownership. IndiaCo further builds on this competency using a mix of informational interviews, document generation, and developing internal communities of practice.

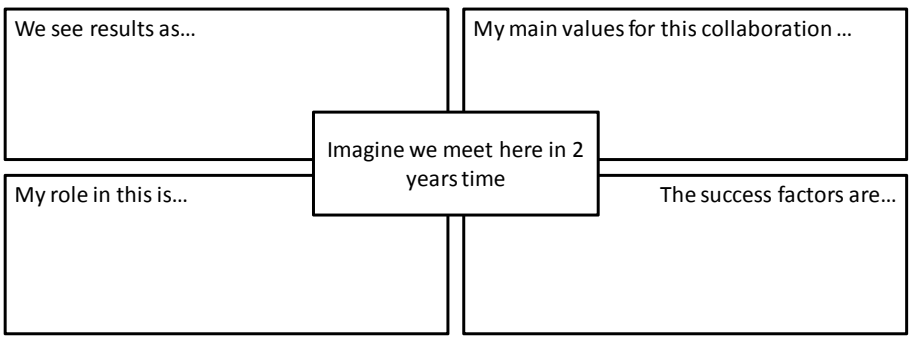


Figure 5-7 Visioning Template

The current model of taking product ownership that was jointly designed by IndiaCo and EuroTel ensures that the risks associated with IndiaCo taking over complete product ownership are identified and mitigated against. As one of the participants' envisioned:

> *"We become world class, are learning from each other, we have a common innovation lab and exchange students and technology - we help each other to get additional market share."*

As we noted in the negative brainstorming section, IndiaCo has to constantly manage against talent attrition, while at the same time growing the internal domain competencies and overall capabilities of the organization. The notion of building competencies is not just associated with IndiaCo – EuroTel relationship. There is a significant challenge in educating other members of EuroTel (and to some extent the larger TelCorp enterprise) about the management challenges, impact, and overall effectiveness of working with IndiaCo. Their managers need to bridge across cultural (both internal to EuroTel, as well as in working with an Indian company) boundaries.

5.3.3 Triangulation Survey

In addition to these exercises, we conducted a brief survey of the sixteen participating members to assess the nature of the relationship between EuroTel and IndiaCo. In this exercise, we explicitly differentiated between the customer and supplier to see if there was any significant variation based on the groups, and to provide validity to the recommendations. Each of the participants was asked to rate on a scale from 1-7 (with 1 being strongly disagree, and 7 being strongly agree) their perception of the current state relation across nine areas, as shown in Table 5.3. We discussed the survey with the participants to ensure that everyone had the same understanding of the questions, and that the ratings scale was consistent. Furthermore, we had to discuss the responses with two of the participants to ensure that we could include their assessment in the overall findings. Across the board, the survey showed that there were opportunities for improvement. With the exception of two areas, there was strong correlation between the people from both organizations about the overall health of the relationship. The two areas of contention were in requirements generation and long-term plan sharing.

Table 5-3 Workshop Survey Questions

	Survey Questions
1	Overall, our relationship is trust-based
2	Our communications are timely and accurate
3	The requirements generation process is timely and accurate
4	Our KPI's are constant and consistent
5	EuroTel shares their long term plans with IndiaCo
6	EuroTel invests in IndiaCo's success
7	IndiaCo constantly changes their team composition
8	IndiaCo's human capital is growing to meet EuroTel's needs
9	EuroTel provides IndiaCo with sufficient time to plan, reflect and learn

Given that a majority of the projects are sustainment projects, the requirements generated are often in the form of user filed trouble reports. These reports have varying levels of information content and the IndiaCo team has to refine the requirements prior to actually making any changes. This problem was further exacerbated when IndiaCo does not have direct access to the people who actually filed the trouble report. This lack of access has to do with the reluctance on EuroTel's part to let its end customers (the people generating the trouble reports) communicate directly with a supplier. While they want their end-customers to have a seamless experience, treating IndiaCo as a supplier does not support that intent. IndiaCo has repeatedly stated that they want to be viewed as a partner, and have direct access to the sources of the trouble reports themselves. As one of the attendees noted:

"IndiaCo will have full product lifecycle management in place, and delivers results at a cost < 80% of the original assignment."

The other major gap that emerged was in sharing long term plans – IndiaCo believes that on the products that they sustaining, they have developed domain knowledge to become contributors to the long-term strategy. As one of the workshop attendees explained:

"IndiaCo is helping EuroTel sell our platforms to other customers, always on time and content with quality to exceed expectations."

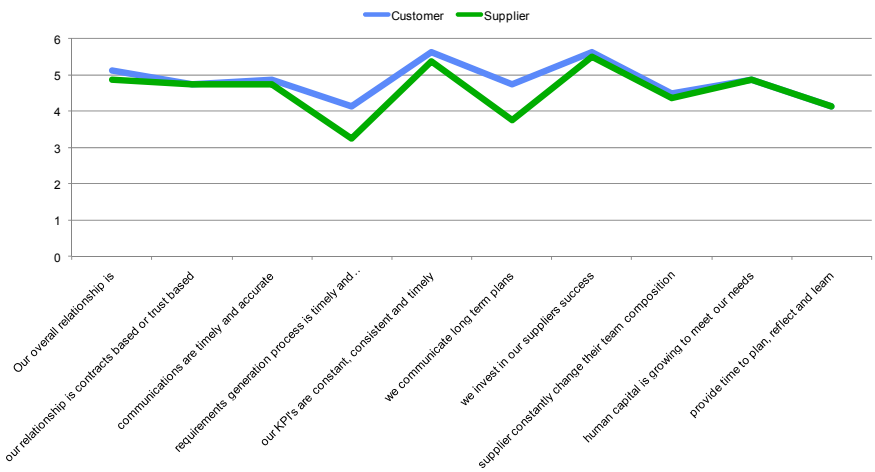


Figure 5-8 Survey Findings from the Workshop on Relationship Building

5.3.4 Learning from the Workshop

The workshop resulted in the identification of improvement opportunities that would support the growth of the IndiaCo-EuroTel relationship. In addition to the traditional focus on improving process and project capabilities (cost, schedule, and quality), the major improvement opportunities included:

- Leadership Development
- Strategy Co-Creation

- Establishing Competence Engines

Both IndiaCo and EuroTel have been successful at growing their leaders internally. The focus on leadership development is to enable them to groom the next generation of leaders who would understand the requirements, and can act as ambassadors for both organizations. This can be achieved through job rotations where managers from IndiaCo work on internal EuroTel projects and vice-versa. An important enabler to this leadership development exercise is building the competence engine that allows both organizations to identify, evolve and focus on their core competencies.

As a result, IndiaCo sees itself as a strategic partner to EuroTel, providing value added software services, and potentially doing greater segments of the engineering work associated with product development. EuroTel would like to leverage these capabilities to effectively improve both the current performance, as well enhance their focus on creating innovative products and services. This requires effective knowledge management & transfer strategies, wherein EuroTel can pass on domain knowledge to IndiaCo's core team.

One of the recommendations that emerged from the workshop was the creation of a competence engine that would focus on hiring, educating, and retaining talent in the IndiaCo-EuroTel relationship. This competence engine would also be used to directly support innovation through a greenhouse for piloting ideas and carrying out joint product development.

The most important outcome of the workshop was the increased recognition in both organizations to co-create their long-term strategy. They both recognized the importance of having a strategy in place that had the buy-in of key stakeholders in both organizations. IndiaCo's focus on obtaining a larger portion of product ownership and EuroTel's willingness to allow them do so, has been tempered by cultural and operational challenges. One of the recommendations that came out of the workshop was to look at the entire portfolio of projects, not through the Profit/Loss lens, but rather in terms of goals, competencies, and effectiveness. This would enable both organizations to focus on long-term effectiveness without necessarily sacrificing short-term objectives.

5.4 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 match a 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 discussing the case, we start with the researcher evolution at IndiaCo, followed by a discussion of the themes that emerged from the data analysis.

5.4.1 Setting the Stage

In the IndiaCo case study, we used the approach of engagement, exploration, focused field work, and knowledge sharing, as discussed in Chapter 2. We engaged the senior-leadership team of IndiaCo at the creative workshop, and obtained their buy-in to initiate the first round of fieldwork on-site in India in the month of March. Both the creative workshop and the first round of fieldwork fall within the exploration phase. While in India, we conducted sixteen interviews in total with twelve project managers, two senior leaders, and two developers. The focus on project managers was chosen 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.

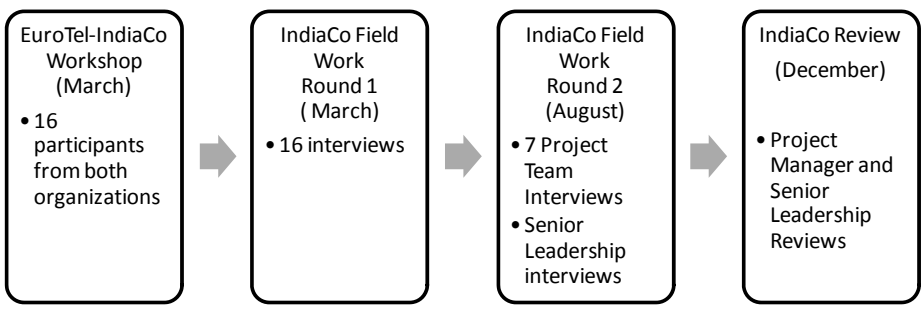


Figure 5-9 Research Evolution at IndiaCo

In addition to the interviews, we were also given access to the training materials used internally in IndiaCo, and to some project data. Once the preliminary data analysis was complete, we did a second round of fieldwork in august of the same year, when we focused on team interviews with members of all seven projects that formed the core portfolio at IndiaCo. These interviews gave us deeper insights 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 final round of fieldwork, working primarily with the senior leadership team to share the lessons learned, and to discuss challenges imposed by the recession.

5.4.2 Focusing on Long-Term Performance

IndiaCo's business model is built around three broad classes of projects: Product Sustainment, Product Ownership, and Knowledge-Work, as shown in Figure 5.10. Of the seven projects that are currently in their portfolio, four are sustainment only (with EuroTel retaining product ownership), two are owned and sustained by IndiaCo, and one involves purely knowledge work. IndiaCo's first engagement with EuroTel was through a knowledge-work project in Australia. IndiaCo recognizes that projects in this class enable them to gain greater exposure to other ongoing projects at EuroTel, in which they can have greater participation. The downside of this class of projects is the uncertainty in resource loading – IndiaCo manages the volatility by rotating personnel between the other projects.

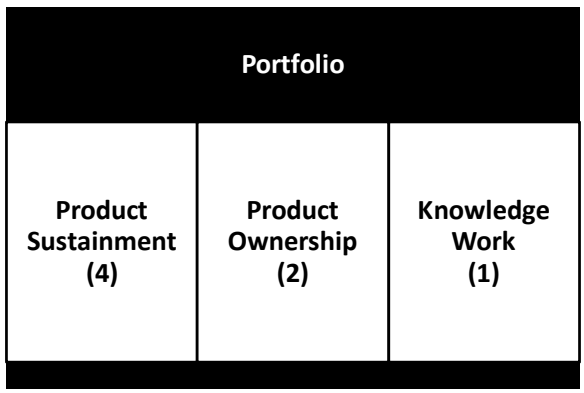


Figure 5-10 IndiaCo Project Portfolio

Yet, none of these projects would ever have emerged if not for a highly strategic move on the part of the senior IndiaCo leadership in 2000. It is important to mention that they had a more traditional consulting role with EuroTel's parent TelCorp, as the head of IndiaCo noted:

> *"Our senior leadership team basically went in to TelCorp's leadership team during the 2000 recession, and told them that we understood that were suffering, and that we were voluntarily going to reduce our rates to help them get through this difficult time – that is what got us our first contract with EuroTel"*

The first project that IndiaCo executed for EuroTel was built around a core team of people, all of whom had deep domain knowledge in the telecommunications industry. It was the performance of this team (whose members were drawn from other ongoing projects, a definite hit to the short-term efficiency of IndiaCo), that resulted in the establishment of the offshore development center. As noted earlier, they recognized that the primary motivation, from their customers perspective, was (and to large extent still is) cost savings. They have demonstrated continuous reductions in costs and improvements in the quality of their end product through process improvement. Their efforts don't just stop there – even though the offshore development center is only a small portion of the total revenue of IndiaCo, it is seen as a strategic relationship that will last well into the next decade. The focus of IndiaCo's leadership team on long-term performance is further demonstrated by their creation of a special internal team to look at potential product improvement opportunities. They realized that under a time-and-materials contract, it was impossible to dedicate billable resources to identify the task, and at the same time, felt it was necessary. As one senior project manager put it:

> *"We have the best insight into product improvement opportunities as we are carrying out the maintenance operations. Over the last two years, we have consistently improved both the documentation and the quality of the final product. Our internal team identified twenty improvement opportunities, and executed on some of them using our own funds. As a result, the customer is now funding some of the improvements. Our goal is to have complete product ownership, and we have to demonstrate value in order to do that."*

Both the sustainment and ownership classes of projects require effective knowledge transfer and knowledge management to ensure success.

5.4.3 Knowledge Creation and Transfer

The issues of resistance to outsourcing have been well documented, whether it is in the trucking industry, or in software development. IndiaCo recognized the resistance on the part of EuroTel members during their first project, and created a knowledge transfer mechanism to capture system characteristics. In reminiscing about the project, one of the first team members told us:

> *"We were not given sufficient anything... People would give us just enough information to stay out of trouble, and when we wanted more, they would say that it was all they had. My team made sure that our leadership knew both who was causing us difficulty and what we were looking for in terms of data or access. My project manager at the time recognized that in some cases, the client really didn't have the data or documentation, and tasked us to essentially recreate it. It was a grind - we would generate the documentation and then out brief everyone else on the team on a daily basis. In the end, we had a better understanding, and codified knowledge about*

the system, than the client. We have since institutionalized process for all of our engagements."

Even today, IndiaCo does not have access to product documentation from either their offsite location in India, or through their onsite team members. They have to rely on their client-side counterparts to supply the necessary information. This 'request and wait' cycle introduces significant delays in the development lifecycle, and has placed even more importance on the knowledge creation activities prior to the start of a project. The process that has been institutionalized by IndiaCo today is shown in Figure 4. It beings with a review of existing documentation at EuroTel, followed by formal knowledge transfer sessions when EuroTel experts share their knowledge.

Figure 5-11 Knowledge Creation and Transfer Process

Any identified gaps in knowledge are filled by creating new documentation and socializing the knowledge between IndiaCo team members. Each transfer team member essentially becomes an expert in some aspect of the system that they are taking over, and shares his or her learning with the team through daily meetings. When the team feels that they have sufficient information, they transfer the knowledge to the offshore team, which continues to build and refine the knowledge base. When IndiaCo first developed the process, it took six months – this time period has now been compressed to a twelve week process. This compression has been driven by the senior leadership in both organizations. As one of the project mangers noted:

"We are at our roots a services company – our leadership doesn't completely understand the challenges of working in a product environment, and they will not pass up an opportunity to gain more business. The compression of the knowledge transfer schedule has some unexpected side effects"

A case in point is the challenges that IndiaCo has faced with one of the recent products of which they took complete ownership. The product was a tool that was used to load embedded software onto a target hardware platform. When we were doing the second round of field-work in August, there was a trouble report that was filed by their client that the tool was not working, as their attempt to load a block of software onto a target board resulted in a checksum error. Interestingly enough, another block of software was not affected when the IndiaCo tool was used.

In discussing how the problem was handled, the project manager in charge of the product said:

> *"We received the trouble report on Thursday (August 14th) and started to look into the problem, since August 15th was national holiday in India followed by weekend, we didn't get a solution back to the client on Friday. The issue was escalated by the client to our senior leadership on Sunday August 17th. On Monday, 18th August we created the task force to look into the problem with high priority and had the technical team setup the communication with the client organization"*

The initial investigation found that IndiaCo did not have the required hardware. The hardware itself had not been upgraded in the last few years as the product was stable. Since there had been no major trouble reports involving hardware in the two years leading up to the problem, the hardware was not transferred to IndiaCo. IndiaCo requested access to hardware at the client-side using a remote login available into the client's test environment. This would allow IndiaCo's team to load software from India and have an on-site team support their debugging activities.

As part of the problem solving process, they observed that old version is being used in basic testing at test site and hence, as a workaround suggested that it be replaced with the current version to see if the problem still persists. Since the problem was isolated to one block of code (say X block), and not the other (X1 block), they double checked the input files for the problematic block again. Since it is rare for the tool to work with one block and not another, they explored the possibility that the problem was in the input file that was being sent to the tool itself, as opposed to the in code block. They analyzed the tool using trace points in the client test environments, but found no faults. They explored the other permutations and combinations such as:

- Problem might be in updated source code – they verified that it was not the case by loading an old version of X block file with the tool, and found that the tool worked as expected.
- Problem might be in Build environment used to create the input (intermediate) files. If the input files were wrong, then the tool would not generate the required output. This should be checked with Build manager. Apart from this, building the source code for the X block file can be done with build support tool from latest release toolbox to generate the .chk files to be loaded. This appeared to be a problematic area, and the client team had to investigate the problem further, as their build tool appeared to be introducing the error.
- Problem might be in the operating system. This was verified by the client to show that the problem did not exist in the operating system.

Upon analysis completion, it was shown that the problem was not in the tool itself, and the client had to investigate their own tool chain to determine where the source of the error was. Among the lessons learned from this crisis, was that they were unable to investigate the problem due to a confluence of timing and lack of hardware. The lack of hardware was partially attributed to the compressed schedule during the transfer of knowledge. A recommendation that came out of this investigation was that for all old/new tools, the hardware/software dependencies should be checked and made available to prevent future problems. From a human standpoint, this crisis resulted in the team spending almost two consecutive days in the office in order to respond to their client's escalation.

This root cause analysis that they carry out for every project is captured using a structured template, as shown in Figure 5.12

Root Cause Analysis Template
Problem Description:
Background Information:
Approach:
Analysis:
Conclusions:
Lessons Learned:

Figure 5-12 Root Cause Analysis Template

5.4.4 Growing Leadership & Building Human Capital

Leadership is a critical element in successful software development in general. Athey's leadership competency framework (Athey, Consulting & Collions, 1998) identifies the ten areas of business literacy, technology vision, cross-functional orientation, strategic partnership management, customer relations, total quality discipline, market decisiveness, technical teamwork, knowledge development, and leadership versatility, as being the needs of the software industry. All of their project managers have to take and pass a project management course that teaches them the core and soft skills needed to effectively manage their teams and their relationships with their clients, prior to being appointed to that role. IndiaCo grows its leadership internally – in fact, most of their project managers have been part of the organization since its inception. They are extremely selective about lateral entries into leadership roles of personnel who are not part of the offshore development center, even if they are from within IndiaCo. In essence, IndiaCo is 'a respected community of peers'

(Borchers, 2003), making it hard for someone who has not grown up within the organization to be effective. IndiaCo's emphasis on open and honest communications applies both internal to the organization, as well as in their relationships with their client. It is driven by the senior leadership and strongly reinforced by their quality and project management systems. As the head of IndiaCo pointed out:

> *"I am not going to hold you responsible for bad news, as long as I know about it with enough time to engage the client if needed. That is not to say that you can consistently make mistakes – every time there is an issue with a project (cost, schedule, resource issues), the project manager is required to file a root cause analysis report. I look at them and see if there is a pattern of behavior emerging"*

This attitude is also reflected by the project managers:

> *"If I see a problem coming, I would rather let the client know ahead of time, rather than have it come up in the review meetings"*

The measurement system that IndiaCo has established enables them to track project progress across multiple dimensions, even though they most often report only cost and schedule metrics to their client (This is driven by the tracking mechanisms used by the client, and not information hiding on the part of IndiaCo).

IndiaCo has been remarkably successful at managing their human capital base. They recognize that people are their greatest asset. In the Indian market for talent, which is 'a seller's market', it is important to note that their attrition rate in early 2008 of 11% was much lower than the national average of 18%. A recent CiteHR.com poll on 'Who is responsible for attrition?' in the Indian software industry, 382 HR professionals voted across the three areas of Employee (7.59%), Supervisor (38.22%), and Compensation/Job Profile (54.19%). These results further support the foresight of IndiaCo's leadership in eliciting and meeting the value propositions of their employees. As the head of the IndiaCo offshore development center noted:

> *"We are very cognizant of the socio-economic pressures that our team members face"*

Discussing the evolution of a typical team member, he highlighted the importance of understanding that the needs and expectations of each of the team members evolves over their career. Consider the evolution of a young professional, fresh out of college, at 21 (as shown in Figure 5.13.), he/she is extremely excited to work in new technologies and grow their skills. They all come in knowing that they need to gain experience before they can go abroad. Three years into their career, they put in a request to their supervisors to work on the client's site. This request is not so much governed by the nature of the work, as it is by the social pressures from their families – since the expectation, when you work in India's software industry, is that you will be sent abroad. If this person does not go abroad, he/she can walk out of the door with all of the skills provided by IndiaCo, and have their expectations met by

their competitors. The international assignment is also financially lucrative, as it enables them to buy a new home for themselves (often, this is their first home – though the current credit availability in India has changed some of that). After completing their two or three years of international assignment, the social pressures come into play again. At 26 or 27, they are considered marriageable, and parents in particular apply consistent pressure to enter into matrimony. This is the second point when people tend to leave the organization, typically because their spouses are located elsewhere. Once married, he/she would like to take their spouse abroad, resulting in increased pressure on the management team to send them on an international assignment. Within a year, the professional determines whether or not he/she wants to continue to stay abroad, mostly driven by the preferences of the spouse. This is where the challenge of managing middle managers comes to the forefront, because now there is an experienced person with 3-4 years of client-side experience not wanting to work on an international assignment. At around 30, they decide to have children. The competition to get into good schools is so fierce right now that you have to get your child into the school of your choice at the kindergarten stage itself or risk not having them go to a good school. Now there is a situation where a professional with 10 years of work experience does not want to travel on international assignments. We capture this evolution across the five phases of entry, 1st International Assignment, Status Change, 2nd International Assignment, and Stabilization (pre child, and post child).

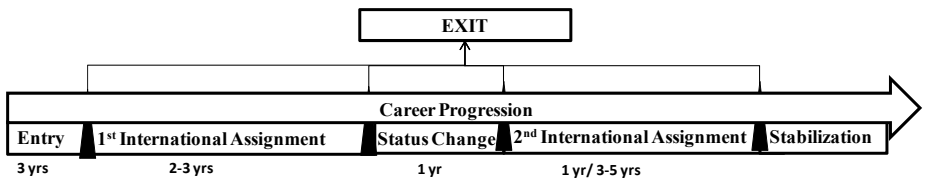


Figure 5-13 Challenges of Managing Personnel

IndiaCo's leadership team effectively manages the socio-economic pressures on their employees and has built a robust organization through their selection of team members, growing of leadership internally, communicating their expectations and last but not least, offering compensation packages that are competitive in the Indian marketplace. The importance of managing human capital at IndiaCo was brought to the forefront when one of their project teams had significant attrition in 2005. The attrition was so significant that it was escalated as a constant red flag in IndiaCo's project review meetings with EuroTel. As a means of managing the risks that arise from high team attrition, IndiaCo and EuroTel worked together to determine the root causes of the attrition. The analysis highlighted the need for:

- career planning at IndiaCo,
- the creation of a metric that captured the knowledge needs of the project team,

- the importance of articulating the differences between working in a product environment, as opposed to an application development environment.

During the second round of field-work, the manager of the project that suffered significant attrition noted that every member of the team now had a career plan, and it was his responsibility to ensure that those career plans were kept up to date.

5.4.5 Building Commitment and Competencies

Software development is in many ways, one of the purest forms of knowledge work. There have been multiple studies that emphasize the importance of managing knowledge and supporting organization learning in software development. IndiaCo has put in structures and processes to support learning at both the individual and organization levels. In addition to the 20 hours per year of training that is mandated for all IndiaCo employees, teams self-organize additional seminars and courses for their own members. Every team member is evaluated by his or her immediate supervisor to determine if there are any gaps that need to be filled. They work together to create a learning plan that will allow the individual to obtain the necessary skills to make them effective in performing their role. Meeting these learning goals is part of the annual assessment process within IndiaCo. In addition to the assessment, these learning credits get accumulated as part of one's expertise rating, providing social incentives as well.

As one project manager noted:

> *"The hardest part with the young professionals is keeping them focused. I am required to review everyone's learning plans and sign off on their applications for taking classes. Since the plan is online, and a student may request as many courses as he/she wants (prior to obtaining approval), HR often sees that there are unapproved courses, and starts harassing me to either approve or reject the request. There are times when I know someone needs to take the course, but we are too busy with project work for me to spare them during work hours. Our learning center is great in scheduling classes during the evenings or weekends. It also allows me to teach in these courses (something that I am required to do by my boss, and something that I enjoy doing"*

IndiaCo's personnel rotation between design/development activities, verification activities, and support documentation generation, enables them to create a shared mental model within the team about all aspects of the project. This strategy for disseminating tacit knowledge across the team is ably supported by a formal content management system that is accessible by all team members. The quality management process forces project managers to reflect on, assess the status, and codify both the root causes and expected challenges for each project. This structured process results in a growing base of codified knowledge, as well as increases the project manager's ability to guide and manage his/her project. Another instance of IndiaCo's commitment to organization learning is in their participation in student exchange programs from European universities. IndiaCo's leadership recognizes the advantages of

exposing the next generation of leaders to the unique strengths (and weaknesses) of working in or with an Indian firm. They have recently started reverse pollination, wherein they are sending students to study in European universities for higher studies, and working with the universities to design programs that enable their students to work closely with or at the client's site.

IndiaCo ensures personnel stability through a rigorous selection process, and augments that with a contractual obligation to remain in the offshore development center (ODC) for three years. This contractual obligation applies even to people transferring into the ODC from other parts of IndiaCo. IndiaCo instituted the selection process when they realized, early in their evolution, that people who joined the team without understanding the challenges of working in a product-centric environment, either requested transfers to other parts of IndiaCo, or left the organization completely. There are two tracks within the selection process: one for young professionals joining the offshore development center straight from school, and the other for people requesting transfers into the offshore development center. The young professionals track focuses on educating them about the nature of the work and the level of commitment required for excelling within the project. As one of managers noted:

> *"Most young hires want to work on sexy technologies such as Java or .NET that will immediately improve their market value in six months. We make sure that they understand that it will take them at least a year to understand the product itself, and that they would be using customer proprietary programming languages and platforms. The value that they get is in the training in problem solving and improvement in the fundamentals such as data structures and algorithms – not in the learning the hottest new thing"*

The EuroTel team has the reputation within IndiaCo as being one of the best places to work. As a result, they get a large number of people requesting transfers into the team. The selection procedure for these people includes a detailed analysis of their technical and soft skills in other projects. An interview with the person is held by the project manager or team leader to determine if the person is a cultural fit for the organization. A transfer into the offshore development center does not necessarily mean that the person gets to retain their earlier designation (they do get to stay at the same pay scale that they were at prior to joining it). As one project manager said:

> *"You have to earn your place within the team. I had one person wanting to join my team, with over ten years of experience. In my discussions with him, I emphasized that fact that he would be starting as a project leader, and not a project manager. If he proved his abilities to the team as a whole, then we would definitely fast track him. That person spent more than six months as a project leader, and really did not want to transition into the project manager role"*

The three year contract requirement further serves to reinforce the nature of the commitment expected to be part of a product sustainment operation. One of the issues that IndiaCo faced

with their client was the approach that the client adopted to measuring team-level competence. Given the high attrition rates early in their existence, the client imposed a required level of competence metric to be reported out on a regular basis. The issue however, was in the measurement of the competence – which was based on the years of work experience and the exposure to specific technologies – it did not capture the domain knowledge of team members. Through a joint investigation of the problem, with their client, they were able to agree upon a metric that measured the competence of 'key' project personnel as opposed to the entire team. This enabled IndiaCo to grow younger professionals and rotate key people across projects within the organization to gather deeper domain expertise.

From a project execution perspective, IndiaCo internally has both stability and flow obtained through effective project planning and deterministic release planning. Team members involved in projects are rarely there on the weekends. As one of the senior leaders noted:

> *"If I see our team members in the campus on a weekend, I make it point to find out why they are here. Often it is because they are exploring something else, and the air conditioning in the office makes it a more comfortable place to be. If they are working on a client project, that raises a red flag, and I take it up with the project manager"*

Given that most of IndiaCo's projects are in the maintenance and evolution arena, the number of projects they have ongoing concurrently is variable. Furthermore, since the request for work is generated in a distributed manner on the client's side (client side project managers have discretion to request work to be performed, without getting central approach within their organizations), there is significant variation in the expectations with respect to delivery. The project managers are responsible for shielding their teams from this instability. Team members within a project often take on documentation and minor bug fixing activities in other projects during slack time to build up overall system knowledge. This becomes invaluable during crunch times since multiple members within IndiaCo can contribute to the project.

5.4.6 Creating a System of Governance

IndiaCo's onshore-offshore model of software development represents an internal interdependency, and suffers from the communication, coordination, and control challenges imposed by distance. The coordination complexity is alleviated, to a large extent, by an effective partitioning of labor between the two teams, with the requirements generation, specification development, and software transfer being carried out by the onshore team, and the core development and verification being carried out by the offshore team. Additionally, the rotation of key personnel between the offshore and onshore teams ensures that there is a collective understanding of the challenges faced in both teams. From a communications and control standpoint, the monthly meetings between the offshore team and the onshore account manager ensure that there are no critical impediments to project success. Another challenge

that IndiaCo faces, originates from the variation in process maturity between them and their client. As one project manager explained:

> *'There is a challenge in that our processes are more mature than those of our client, and while they expect stringent performance on our side, their own processes are often lacking'*

The approach adopted within IndiaCo was to 'reasonably' tailor their standard process framework (that is deployed across all of IndiaCo), to meet the process specifications of their client. Their ability to morph their process, without losing its essence, allows them to meet the same corporate-wide audit requirements on process compliance and quality monitoring. More importantly, they are able to transform this reporting burden into a sensemaking tool, as they have to track a larger set of process and performance metrics than mandated by their client. This disparity in process maturity is understood by their clients as well, who have initiated their own process improvement efforts as a means of reducing the gap.

5.4.7 Learning from IndiaCo

IndiaCo's success can be attributed to their leveraging of an effective business model that focuses on long-term organizational performance. This focus is driven through a very effective knowledge capture and transfer process, that is strongly supported by their approach of organizational learning, and their strategy for developing and maintaining competencies. Their growing of leadership internally, and strong governance structure has enabled them to build a culture that is slowly evolving into a product culture as opposed to purely a services culture. This enterprise approach provides overarching guidelines around which senior leadership can build a set of reinforcing practices to guide their organization's transformation. Our intent is not to present this as the ideal case, rather to provide an example where an evolutionary process of continuous transformation results in the onus shifting from being leadership-driven to becoming self-sustaining and self correcting. The IndiaCo case allows us to present concrete examples of how this has been achieved. Their journey is by no means complete, but IndiaCo is in many ways, a world-class software organization.

5.5 Conclusions

The ability to explore both the client-perspective and the supplier-perspective within the context of a research study provided us with insights into the challenges of outsourcing maintenance:

- Awareness of competence (added/lost): one of the issues that emerge from outsourcing the maintenance of a product is the potential lock-in with the supplier. More importantly, it forces the firm to make a conscious choice on what its core competencies are, and determine where it can find the best competencies.

- Disparity in maturity: the EuroTel/IndiaCo case represents a high variability in the maturity of the supplier and the maturity of the client. In our case, IndiaCo's maturity is far greater than that of EuroTel.
- Constructing a Collaborative value proposition: constructing a collaborative value proposition requires significant engagement on both sides and this engagement has to be reflected in positive action.

6 TOWARDS A THEORY OF ENTERPRISE AGILTY

6.1 Introduction

In earlier chapters, we identified the three mechanisms that software organizations can use to obtain enterprise agility – namely software process improvement; creating systems of innovation; and leveraging globally available capabilities. We contend that there are four organizational enablers (stakeholder alignment, employee empowerment, group & organizational learning, and governance mechanisms), have to be present to some degree before an organization can develop enterprise agility, as shown in Figure 6.1.

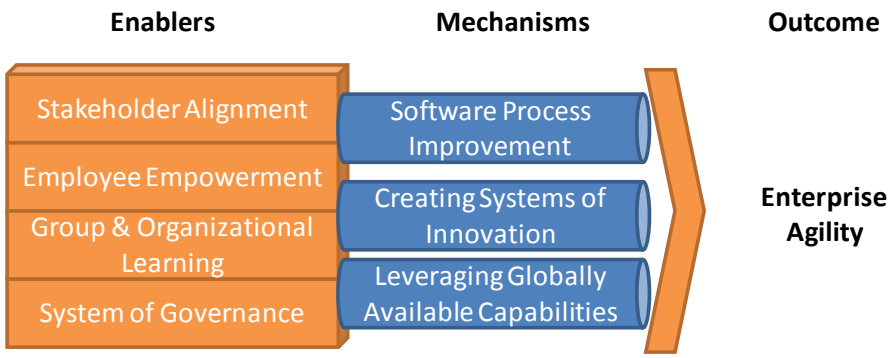


Figure 6-1 Developing a Theory of Enterprise Agility in Software Organizations

We discuss each of the organizational enablers separately in the subsequent sections, and pull together the disparate elements to propose a theory of enterprise agility in software organizations.

6.2 Stakeholder Alignment

From the perspective of achieving stakeholder alignment in software projects, the best known model is the win-win spiral model (Boehm, Egyed, Kwan, Port, Shah & Madachy, 1998; Boehm & Ross, 1989). The first two phases focus on identifying the next level of stakeholders, and determining the win-win conditions for these stakeholders. The emphasis in the third phase is in determining the feasibility of meeting those stakeholder expectations. The three following phases are focused on enabling execution, and the seventh and final phase focuses on obtaining stakeholder consensus. As Boehm (1994) points out, the emphasis of stakeholder commitment to shared systems objectives provides the organization with a collaborative framework for helping people and organizations cope with change. The

importance of software organizations managing the needs of their key stakeholders has been emphasized at the project level from the perspectives of architecture (Clements et al., 2002), requirements management (Damian, 2007), globally distributed development (Damian & Zowghi, 2002), just to name a few.

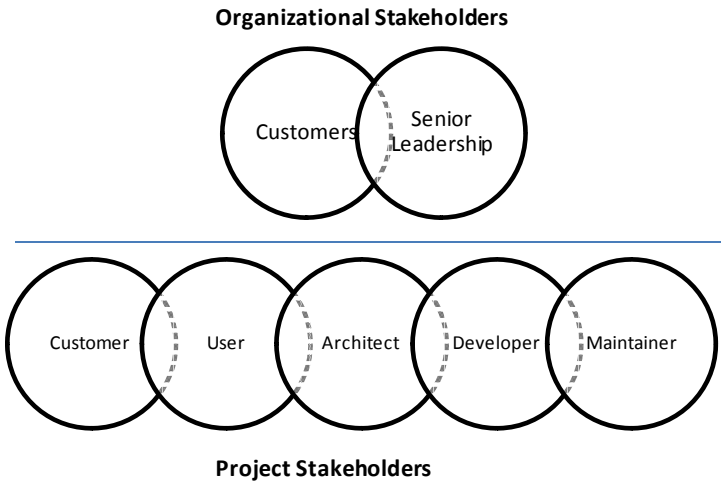


Figure 6-2 A Bi-Level Perspective on Stakeholder Alignment

The win-win spiral model however, does not explicitly address the needs of organization-level stakeholders. An extension to the model is proposed here, to account for the fact that software organizations have to necessarily take a bi-level perspective, addressing the needs of senior leadership, and customers (both internal and external) at the organizational level, and meeting the needs of customers, developers, architects, users and maintainers at the project level. In effect, the model views a software enterprise as a portfolio of projects, that each has a set of stakeholders, who are often distinct from the organizational-level stakeholders, as shown in Figure 6.2.

6.2.1 IndiaCo Stakeholder Alignment

From the perspective of alignment to its external stakeholders, IndiaCo has strong alignment to the values of its senior leadership, as well as with that of their external customer. Given that IndiaCo has a strong policy of promoting internally, the basic values across the project teams and the senior leadership are strongly aligned. In addition to the status reports presented in the monthly internal meetings, an escalation mechanism has been put into place, to allow senior leadership to detect potential problems ahead of time. From a customer perspective, at an organizational level, IndiaCo actively manages customer expectations – a

case in point, is their voluntary reduction of service rates when their customer was undergoing significant downsizing. This not only brought them significant goodwill, it also demonstrated that they were aware of the challenges that their customers were facing, and that they viewed themselves as a value-adding partner, not just a supplier. Senior leadership ensures that they communicate their expectations to the project teams, both in terms of long-term strategy, as well as short-term objectives. As the head of the organization noted:

> *"We have a responsibility to demonstrate to the customer that by offshoring they are not only getting superior performance at a lower price point, they are also gaining expertise that they cannot find elsewhere."*

From the project perspective, IndiaCo's project teams are completely disconnected from the end user. The only source of information that they have about end user challenges/expectations comes from the trouble reports that are filed by the end user, and routed through their customer. This disconnect impacts the overall project performance in the case of incomplete/ill-defined trouble reports, as the developer now has to perform two additional steps of verifying that the recreated problem was in fact correct, and then determine whether or not the problem was introduced by the software of which he/she has ownership. While this disconnect has been identified and raised to the senior leadership on the client-side, little action has been taken to address it. The other source of misalignment is with respect to project-level customers. Given that IndiaCo has a portfolio of projects that range from total product ownership to knowledge worker support, there is a significant variation in expectations from middle-level managers on what is expected from the project teams. While the senior-leadership on the client side emphasizes a 'partnership' role, some of the middle-level managers treat the project teams as suppliers, creating intrusive monitoring systems. As one IndiaCo project manager put it:

> *"When we transitioned to a different site, the local project managers sent a person down to the local site in India, to ensure that we were actually doing what we said we were doing. The person stayed on site for six months, did nothing, and eventually went back. Now they are transitioning product ownership to yet another site, and we have to go over this process again."*

This disconnect between the project teams and their customers is an outcome of a lack of uniform policy guidance on the client-side on what is expected from IndiaCo. Again, this disconnect has been identified to the senior leadership, and documented in earlier work (Srinivasan, 2008a, c). The alignment between the architect and the project team again is a function of the organizational affiliation of the architect – when the architect is internal to IndiaCo, the alignment is strong, however, when it is a client-side architect, the alignment is a function of whether the person has been involved earlier in the IndiaCo relationship in some capacity or if they have to build trust in the efficiency and effectiveness of the project team. The lack of alignment between the architect and the software team influences the learning strategies that the project teams adopt.

6.2.2 EuroTel Stakeholder Alignment

At the organization level, there is significant misalignment with senior leadership as well as with their customers (both internal and external). As was noted in the introduction section, EuroTel is currently treated as a supplier within the larger corporate structure; as a result they do not have access to the end-customer. This problem is further exacerbated by a lack of strategic vision for EuroTel. As the head of EuroTel noted:

> *"Right now, there is no clear vision for whether we should exist – corporate (with my inputs), will make a decision on whether we should split EuroTel up and embed the capabilities within the customer-facing organization, or truly elevate it into a core part of the TelCorp enterprise."*

Since EuroTel is currently being treated as a supplier to other business units within TelCorp, their relationship with their internal customers is often adversarial and arms-length. EuroTel project teams feel the brunt of this disconnect, as they are forced to execute in an environment of constrained budgets, and compressed schedules. One of the strengths of the EuroTel is the strong alignment between the architects and developers, most of whom have spent their entire careers in the same organization, often working together on multiple projects. Over the last eight years, they have outsourced a significant portion of their product maintenance efforts to global offshore centers, some of which are wholly owned by EuroTel. The offshoring strategy has resulted in some misalignment, mostly due to a lack of experience in managing global software development. The lack of alignment with project teams involved in maintenance can be traced back to the massive downsizing and subsequent outsourcing that occurred at the EuroTel in early 2000. The disconnect occurs when product owners and line managers focus on 'protecting their rice bowls' during the knowledge transfer to the offshore team, and subsequently, limit the support they provide to that team, if not actively hinder it. In addition to the socio-cultural aspects of the problem, the EuroTel policy of not providing their non-EuroTel offshore sustainment teams with access to the corporate network (and thereby restricting access to project artifacts), further fosters project-level stakeholder misalignment.

6.2.3 GameDevCo Stakeholder Alignment

As GameDevCo transitioned from start-up mode to a full-fledged product company, most of the founders left the organization, and new senior leadership was brought in, creating the first set of misalignment between project-stakeholders and senior leadership. The recent acquisition of GameDevCo's by a global conglomerate has further resulted in misalignment with their corporate customers. The root causes of the misalignment are:

- a limited understanding of GameDevCo's product (and by extension, the market they serve),
- the lack of visibility into GameDevCo's processes,

- and corporate ownership of the high-level product roadmap.

This transition also resulted in a more hierarchical organization structure, wherein more layers were inserted between the project teams involved in the product lifecycle (design, development and sustainment) and their customers. At the organizational level, even within GameDevCo, there is misalignment between the various business units. A case in point is the lack of alignment between the Infrastructure & Support (I&S) business unit and the game development (GD) business unit that develops the games. The lack of alignment here arises from the absence of a structured release process for product upgrades, straining an already limited I&S staff that is constantly juggling patches, and trying to establish version control.

At the project-level, there is a strong alignment between the customers and developers in GD, as the developers have a strong passion for the game for which they developed the product. The misalignment with the architects/product owners is driven by shift in emphasis from 'new technology development' to 'business value'. As the head of GD noted for a product that has been in existence for almost six years:

> *"We only wrote our first business story six months ago."*

The lack of alignment between the developers/maintainers and the architects is caused by both, the constant churn in the development process, as well as the perception of the lack of effectiveness of the architect/product owner. As one of the developers in the first generation product noted:

> *"The people designing the new system created the mess in the first product – over the last two years, we have had to constantly redesign and refine the product, while keeping it in service – getting it to the maturity we have."*

The challenges faced by the maintainers are similar to those faced by I&S stakeholders – they lack sufficient information to effectively support the product. Since the adoption of Scrum as the development methodology of choice three years ago, the rapid execution cycle time has led to poor fidelity artifacts (when they exist). These artifacts tend to decay rapidly, as the collective problem solving activities that are carried out to resolve issues/make design changes are not captured. As one developer bemoaned:

> *"We are sprinting, not doing scrum"*

6.2.4 AgileCo Stakeholder Alignment

Of the four cases under study, AgileCo possesses the strongest stakeholder alignment at both the organizational level, as well as the project team level. AgileCo's senior leadership believes both in the process they have adopted to develop software, as well as in their overarching strategy for value creation. Their philosophy of maximizing value delivered to the customer includes the YAGNI principle (Erdogmus & Favaro, 2003) – being able to tell

their customer that 'you are not going to need it'. These trust-based relationships are nurtured from the beginning by AgileCo's senior leadership. Their unwillingness to accept projects that do not fall either into their core competence or their long-term vision ensures that there is strong alignment at the organization level. This philosophy of alignment extends both top-down and bottom-up at AgileCo.

At the project-level, they mandate the presence of an on-site customer (or an effective proxy), who has, as Boehm and Turner so effectively summarize it, CRACK qualities (Boehm & Turner, 2003), to ensure alignment. Their uses of the practices of pair-programming, role rotations, iteration planning meetings, and daily stand ups, ensure that everyone on the project team has the same shared understanding of, both the current state of the project, as well as where it is headed. Unlike conventional software development teams, every member of the team has access to the entire code base and is responsible for the integrity of the software that he/she incorporates into the system.

As a senior project manager noted:

> *"One of the advantages of writing acceptance tests is that you know when a given feature is going to break the build. I had a young developer working on a feature for a system that was supposed to go live on Monday … Both of us knew that the acceptance test would fail – I left early on Friday evening, and told him to make sure that the acceptance tests passed before checking in the code – when I came back late Friday night, I saw that he had checked it in – when I asked him about it, he told me that was going to fix it by coming in Saturday – instead of blowing up – I used it as a teaching tool for the entire team."*

Table 6-1 Assessing Stakeholder Alignment in the Four Cases

Stakeholder Alignment	Organization-Level Stakeholders			Project-Level Stakeholders				
	Senior Leadership	Customers		Customer	Architect	Developer	Maintainer	User
		External	Internal					
IndiaCo	●	●		●/○	●/○	●	●	
EuroTel	○	○	○	○	●	●	○	○
GameDevCo	○	○	○	●	○	●	○	○
AgileCo	●	●	●	●	●	●	●	●

Legend ● – Strong Alignment ○ - Weak Alignment

6.2.5 Discussion

The comparison of stakeholder alignment across the four cases is shown in Table 6.1. The two organizations that have demonstrated success at aligning their key stakeholders at both organization and project levels, have established an environment of trust with their key

stakeholders, and built upon those foundations using well-defined and well understood software processes. In the case of EuroTel and GameDevCo, they have pockets of strong alignment. Since the values of senior leadership and customers are either not articulated to the project stakeholders, or in some cases plain misunderstood, the project level stakeholders in both organizations face friction both internally and externally.

6.3 Employee Empowerment

The concept of employee empowerment appeared as part of the management vernacular in the 1980's and rapidly became an often used, yet poorly understood concept. As Randolph (2000) noted, *empowerment remains one of the most promising, yet mystifying, concepts in business*. There are multiple constructs that are associated with empowerment (Conger & Kanungo, 1988): a relational construct at the organizational level, in which the principle source of power that an actor has over the organization arises from the actor's ability to provide some performance or resource that is valued by the organization or the actor's ability to cope with important organizational contingencies or problems, as defined in Pfeffer (1977). A relational construct at the interpersonal level often implies the leader/manager sharing power with their subordinates, and a motivation construct wherein empowerment focuses on enabling employees.

As Aoyama (1998) points out, one of the goals of agile/iterative software development is to transition from a culture of enforcement to a culture of empowerment. In looking at the cultural assumptions underlying the adoption of the Software Engineering Institute (SEI)'s Capability Maturity Model (CMM)-based improvement efforts (which are predominantly used in organizations that use plan-based development approaches), Ngwenyama and Nielson (2003) note that empowerment is implicit in organizations that have a developmental orientation, in which the focus is on human development. From the perspective of enabling software organizations to create enterprise agility, employee empowerment is essential. People act as the primary sense making mechanism in software organizations, and an unempowered employee cannot effectively be a part of creating or contributing to enterprise agility. Empowerment is seen as a critical factor in enabling software process improvements (Dyba, 2000),but actually creating empowered employees is difficult (Baddoo & Hall, 2002). The four characteristics of empowered people are (Quinn & Spreitzer, 1997):

- a sense of self-determination (free to choose how they do their work);
- a sense of meaning (care about what they are doing);
- a sense of competence (confident about their ability to do the work);
- and a sense of impact (ability to influence their work unit).

These four characteristics provide the structure that we can use to better understand employee empowerment in the four case studies, as shown in Table 6.2.

6.3.1 IndiaCo Empowerment Strategy

In adopting an internal growth and promotion model, IndiaCo has been successful at creating a culture that supports empowerment. Their software lifecycle processes are designed to enable their employees to manage day-to-day activities as they see fit. Every member of IndiaCo has a focus on delivering customer value, which in the services business is key to enabling long-term sustainability. Their approach to selecting employees focuses on both their domain knowledge, as well as in fostering a deeper understanding of the challenges of operating in a 'product domain'. As one of the project managers in IndiaCo noted:

> *"Most young hires want to work on sexy technologies such as Java or .NET that will immediately improve their market value in six months. We make sure that they understand that it will take them at least a year to understand the product itself, and that they would be using customer proprietary programming languages and platforms. The value that they get is in the training in problem solving and improvement in the fundamentals such as data structures and algorithms – not in the learning the hottest new thing"*

The tension that arises when considering the competence aspect of empowerment is due to IndiaCo's business model. Since they are an offshore services provider, members of the client organization often feel threatened by their presence, and sometimes take an adversarial stance when transferring technical knowledge to the offshore team. When reflecting on the challenge of ensuring successful product transfer, one of the team members noted:

> *"We were not given sufficient anything... People would give us just enough information to stay out of trouble, and when we wanted more, they would say that it was all they had.."*

From an impact perspective, internal to IndiaCo, the team members all feel that they have a strong impact on the organization's ability to meet its strategic objectives. This impact aspect is further fostered through the use of approaches such as the Innovation week, where team members can contribute their product/process improvement ideas. The most recent innovation week was held in December 2008, with 100 ideas being generated and distilled down to 35. These ideas are currently being prioritized for implementation. More important to this discussion is the coupling of these activities to individual incentives where the best three ideas are rewarded and the initiator is recognized. There is a gap in impact when the teams do not have complete product ownership.

6.3.2 EuroTel Empowerment Strategy

EuroTel is still struggling to recover from the massive downsizing that they went through in early 2000. A well defined process has enabled employees to manage their day-to-day activities within the bounds specified. The downsizing has had significant repercussions with respect to the meaning and competence aspects of empowerment. Given that a large

percentage of the workforce had operated under the assumption of lifetime employment at EuroTel the across-the-board downsizing (the number most commonly used was 50%) has resulted in a risk-averse culture. As one senior leader noted:

> *"A lot of managers now would rather not take a decision, and wait for instructions from higher levels – this is killing the innovation that was the core of the company. Yes, they are able to execute effectively when given orders, but they are not creating new value."*

Two of outcomes of the downsizing were the loss of the deep domain knowledge that was resident in the employees who they lost, and the resultant consolidation of core competencies within EuroTel. While this consolidation is definitely visible in their use of offshoring and outsourcing to gain cost efficiencies, its exact nature still remains unclear, both internal to the organization, as well as to the external observer. One of most significant downsides over the last eight years has been the loss of perspective on the impact employees have on the organization. As a senior project manager transitioning out of EuroTel explained:

> *"In our company, you could always start something new, if you had the drive and could make the case for why your idea was important, management supported it. Now, we still have people that have the ideas and the drive, but do not necessarily have the support."*

There is strong recognition within EuroTel for the need to rekindle the spirit of innovation, as evidenced by their creation of an innovation mentorship program. This program remains at its infancy and the true effects of this program on employees perception on their ability to impact the organization's growth remains to be seen.

6.3.3 GameDevCo Empowerment Strategy

While the adoption of agile processes, in general, increases the autonomy of the employee, and thereby enables them to make better project-level decisions, GameDevCo's adoption of Scrum as the development approach of choice, has brought some interesting organizational issues to the forefront, primarily among them:

- the absence of training for contract employees
- lack of an effective tool to support the rapid development cycle time.

Their explosive growth over the last four years has resulted in GameDevCo's heavy reliance on contract employees to ensure project progress. However, these employees are not trained in the SCRUM process. As one contract employee noted:

"I have been here for a year, and I still don't know what SCRUM is."

This problem is further exacerbated within the organization, as there is a wide-ranging understanding of what SCRUM is, and how it should be implemented. As a result, the nature of the day-to-day development activities is strongly influenced by the project manager (or scrum master). As was said by one senior project manager:

"We spend more time talking about what SCRUM is, and what it should be, rather than focusing on what it should be doing for us."

This internal process diversity, coupled with the lack of an effective governance structure, has resulted in churning on the part of the employees, when it comes to the development process. Another challenge that impacts their ability to execute day-to-day work is the lack of a requirements management tool that supports the rapid development cycle time. The corporate mandated tool was designed to support plan-based development, and is too cumbersome to support effective situational awareness. As a workaround to the tool, development teams started using a wiki-based tool, however, even there, the artifacts decay rapidly. From the perspectives of meaning and competence, GameDevCo has world-class technology competence, and across the board, the development teams are passionate about the game for which they developed the product.

GameDevCo's growth has led to a greater focus on the business of selling games, and a perception of dilution on technology aspects of game development. This growth has also created a more hierarchical organization that has resulted in functional silos, as opposed to the cross-disciplinary, cross-lifecycle teams that existed in the initial phases of their growth. This problem has been further exacerbated by the move to a common office in a major city, which has exposed their technical staff to a larger, and in some cases, more rewarding job market. From an impact perspective, GameDevCo's approach of using independent teams to develop the next generation product and sustain existing products, has resulted in frustration for the team that maintain the current product. As one of the team members involved in sustainment noted:

"There is always a ghost in the room – we don't really know what our role is when the next-generation product will finally be fielded."

6.3.4 AgileCo Empowerment Strategy

AgileCo represents the best-in-class when it comes to employee empowerment. They have institutionalized the values espoused in the agile manifesto within their organization, ensuring that their employees are both aligned to overarching objectives of the organization, and have the maximum leeway to execute their day-to-day activities. There is a common emphasis on team and customer values that is reinforced by the team. While technical excellence is a pre-requisite for working in AgileCo, the ability to self-organize and self-govern is key to their sustained improvement. As one of the developers noted:

"If you under perform, then nobody wants to pair with you. We had the case of one developer who always seemed to have an excuse for why they were late in delivering their story cards – and invariably, it had to do with their partner. Over time, none of us wanted to pair with that person, and after the first iteration, that person left the team to take on a more managerial role – good riddance is all we had to say – it was a collective effort on the team's part because it was affecting overall project delivery."

From an impact perspective, AgileCo's employees extend beyond just their organization boundaries through an organization-wide focus on community service, and through an almost evangelical approach to teaching their approach to software development, and sharing lessons learned.

6.3.5 Discussion

In all four cases, we found varying degrees of empowerment, as shown in Table 6.2. With EuroTel and GameDevCo, there was a sense of powerlessness on the employees' side. The roots of the powerlessness, however, were different.

Table 6-2 Assessing Employee Empowerment in the Four Cases

Empowerment Characteristics	Self Determination	Meaning	Competence	Impact
IndiaCo	●	●	●/○	●/○
EuroTel	●	●/○	●/○	○
GameDevCo	●	●	●/○	○
ArgileCo	●	●	●	●
Legend ● – strong alignment ○ – weak alignment				

EuroTel is only now beginning to regain the trust of its employees after a massive downsizing. GameDevCo's challenges arise from their evolution from a technology development company to a product company. In IndiaCo's case, the gaps in empowerment were caused primarily by a lack of alignment with their customers (at the senior leadership, at peer-to-peer cross-organizational interactions, or in some cases, a combination of both factors). AgileCo represents the best example when it comes to empowerment – a large part of their success arises from their ability to effectively architect their relationships with their customers, and the willingness to protect the team's ability to self-organize

6.4 Group and Organizational Learning

Software development is in many ways, one of the purest forms of knowledge work. There have been multiple studies that emphasize the importance of managing knowledge (Desouza, 2003; Rus & Lindvall, 2002) and supporting organization learning in software development (Adams, Day & Dougherty, 1998; Faraj & Sproull, 2000; Fichman & Kemerer, 1997; Sanchez & Mahoney, 1996). For understanding the four cases, we use the three lenses of

individual learning (Cohen, 1991), group learning (Argote, Gruenfeld & Naquin, 2001), and organizational learning (Dodgson, 1993; Huber, 1991).

6.4.1 IndiaCo Learning Strategy

IndiaCo has established structures and processes to support learning at the individual, group and organization levels. In addition to the 20 hours per year of training that is mandated for all IndiaCo employees, they organize additional seminars and courses to support learning. Every team member is evaluated by his or her immediate supervisor to determine if there are any gaps that need to be filled with respect to their competencies. The team member and supervisor then work together to create the learning plan which will allow the individual to obtain the necessary skills to make them more effective in the project. Meeting these learning goals is part of the annual assessment process within IndiaCo. In addition to the assessment, these learning credits get accumulated as part of one's expertise rating, providing social incentives as well.

As one project manager noted:

> *"The hardest part with the young professionals is keeping them focused. I am required to review everyone's learning plans and sign off on their applications for taking classes. Since the plan is online, and a student may request as many courses as he/she wants (prior to obtaining approval), HR often sees that there are unapproved courses, and starts harassing me to either approve or reject the request. There are times when I know someone needs to take the course, but we are too busy with project work for me to spare them during work hours. Our learning center is great in scheduling classes during the evenings or weekends. It also allows me to teach in these courses (something that I am required to do by my boss, and something that I enjoy doing"*

Another instance of IndiaCo's commitment to individual learning is their participation in student exchange programs with European universities. Their leadership recognizes the advantages of exposing the next generation of leaders to the unique strengths (and weaknesses) of working in or with an Indian firm. They have recently started reverse pollination, wherein they are sending students to study in European universities for higher studies, and working with the universities to design programs that enable their students to work closely with or at the client's site.

IndiaCo's personnel rotation between design/development activities, verification activities, and support documentation generation, enables them to create a shared mental model within the team about all aspects of the project. This strategy for disseminating tacit knowledge across the team is ably supported by a formal content management system that is accessible by all team members. The IndiaCo quality management process forces project managers to reflect on, assess the status, and codify both the root causes and expected challenges for each project. This process results in an increasing codified knowledge base that supports overall

organizational learning, and at the same time enhances the individual project managers ability to guide and manage his/her project.

6.4.2 EuroTel Learning Strategy

EuroTel's learning strategy currently focuses on the individual and group levels. The significant loss of human capital over the last eight years has brought an increased urgency to develop an organization-wide learning system. However, this system has yet to be put into place. Given the disciplined development process employed in EuroTel, there are a large number of project artifacts that are generated for each project, including design documents, architecture specifications, release notes, and trouble reports. These artifacts provide a rich store of codified knowledge, all of which are available on the internal intranet. Members within the organization have access to these documents and can access them to support individual learning. Operating in a resource constrained environment has resulted in EuroTel suffering from 'single point failures'. As one developer noted:

> *"We have Single Point of Failures – if people assigned to do some work are sick or away then we have no replacements and it also come for support. People are experts on areas and on short notice it is not easy to find replacements. It is vulnerable situation…. Both budget and competence are problems here and we can't afford two experts in same area, which might be possible in bigger setup."*

From a corporate perspective, EuroTel has a corporate mandate to define their current and future knowledge needs, as measured by competencies. Competencies of each individual are assessed using a four point rating scale that ranges from Basic to Exceptional, across the professional, human and business dimensions. While the end goal of this system is to enable better recruitment as well as to retain critical competencies, its impact again remains to be seen.

From a group learning perspective, EuroTel makes extensive use of dependency diagrams and a rigorous project review process to ensure that all team members have the same shared understanding of the current state of the project. Among the challenges that EuroTel currently faces with respect to enabling group learning, is the emphasis on project completion and the allocation of personnel to multiple projects. As one developer said:

> *"People usually have maintenance responsibility and it is a big problem. The development team shouldn't have to work on maintenance projects, but it happens. It should be as little as possible. It is not only Trouble Reports (TR) you have to spend time and it takes away time from your project but it is very disturbing to get interrupted from what you are doing and do something else"*

Another mechanism that EuroTel uses to enhance its learning capabilities is partnering with local universities. By supporting student projects and sponsoring industrial PhD students,

they gain access to cutting edge academic research, as well as analyses of how the organization itself is currently performing.

6.4.3 GameDevCo Learning Strategy

GameDevCo focuses primarily on creating mechanisms to support individual and group learning. Given their main challenge of standardizing to the Scrum process, every member of the permanent staff went through basic training in the process. As was noted earlier, this training was not provided to contract staff, who represent a significant portion of their total human capital. To support their transition efforts, they retained an external consultant to mentor their Scrum masters, however, the consultant left prior to institutionalizing the knowledge, resulting in significant variation in the development process. As one scrum master pointed out:

> *"We spend more time arguing about what the scrum books say about the process, rather than in trying to figure out what the process is trying to do for us"*

The growth of the organization has resulted in a separation of game development and systems operation, further limiting the opportunities for individuals to learn from the debugging of server related problems and the creation of release packages. Group learning at GameDevCo currently occurs through the use of informal communities of practice that meet to share best practices and lessons learned. The challenge, however, is that these communities of practice do not meet on a regular basis due to schedule pressures. Another approach to fostering group learning is the use of retrospectives at the end of an iteration, which by definition, are meant to provide the team with an opportunity to reflect on both the project and the process. However, these meetings have now become a pro-forma ritual. As one developer explained:

> *"Our retrospect starts with the scrum master saying, What didn't work in this sprint? What can we improve in the process?... We look around and in 5 minutes we are done"*

6.4.4 AgileCo Learning Strategy

AgileCo's emphasis is on individual and group learning. Every one of their recruits is put through a rigorous training program that teaches them the standard processes that are used within the organization. In addition to classroom lectures, the new recruits work in teams to create software solutions to problems that have already been solved for actual customers by the organization. This exposes them to both the mechanics of the process, as well as gives them increased technical skills. Given that Indian schools form their largest recruitment center for fresh graduates, AgileCo's parent company has centered their training program for all of their new recruits in India. As their director for training noted:

> *"We expect to do a lot more work in India, so it's good for the young people recruited outside India to come experience Indian culture, learn the process from expert teachers, and build their own learning in a 'safe environment'."*

Another means of fostering individual learning is through teaching. AgileCo encourages its team members to teach at workshops, and share their knowledge with the larger community. The process of teaching their processes and sharing lessons learned enables them to gain a deeper understanding of both the organizational and technical challenges faced. AgileCo's use of a hybrid mix of XP and Scrum includes the use of pair-programming, pair rotations, and daily standup meetings. These three processes enable project teams to have a shared understating of the current state of development, and more importantly, have the capability to step in when a team member is not available. In addition to the communities of practice that have developed over the life of the organization, AgileCo has a strong mentoring process, with both experienced members acting as mentors to younger members, and peer-to-peer mentoring. When discussing mentoring, the expression that was often used was:

> *"We are a village without doors, and you are really not considered to be a senior member of the company until people ask for your help."*

6.4.5 Discussion

Each of the four cases represents a mix of systems, structures, and processes that are used to enable learning, as shown in Table 6.3. The organizations that were largely successful (IndiaCo and AgileCo) created a system of learning that bridged across individual, group and organizational levels. One of the points that is often made when discussing organizations that adopt agile methods is the large amount of tacit knowledge within the organization – i.e. knowledge that is resident in the heads of individuals.

AgileCo leverages the steps in the development process to socialize the knowledge across the team, codify it using the artifacts that are generated as part of the project, and then disseminate it through teaching, mentoring, and most importantly, by using leaders as teachers. IndiaCo's plan-based development approach results in greater codified knowledge, hence, they use of more structured knowledge sharing approaches, such as personnel rotation and their integrated project management system to create and capture lessons learned. They combine this approach of learning from codified knowledge, with formal and informal communities of practice, and with an expertise location system that allows individuals/groups to identify key knowledge sources. In the case of EuroTel, the mechanisms needed to support and drive organizational learning are present, but they have not yet been coupled to create a system of learning. With GameDevCo, the challenges arise from the lack of organizational infrastructure to support learning, as well as the simple fact that the organization is executing too fast to learn.

Table 6-3 Comparing Learning Strategies Across the Four Cases

Learning Strategies	Individual	Group	Organizational
IndiaCo	Ramp-up Training, Mandated Annual Training, Sponsored Students,	Personnel Rotation, Project Artifacts, Communities of Practice	Project Management Process, Content Management System, University Collaboration, Leaders as Teachers, Expertise Locators
EuroTel	Industrial PhD Students	Project Artifacts	University Collaboration, Competence Assessment, Multiple web-based tools
GameDevCo	Process Training	Informal Communities of Practice, Standup meetings, Retrospectives	University Collaboration
AgileCo	Ramp-up Training, Teaching, Mentoring	Pair Programming, Pair Rotations, Standup meetings, Retrospectives, Communities of Practice	Leaders as Teachers

6.5 Systems of Governance

The notion of governance in software organizations has received little explicit focus from the academic community until recently – a case in point being the First Workshop on Software Development Governance that was held as part of ICSE'08. The summary of the workshop identifies the perceptions of 21 participants on what they believed was part of governance process. We coded the raw data to extract common areas of consensus, as show in Table 6.4. While the understanding of governance is skewed to the individual's perception based on their location in the organization and prior experience (in current and previous organizations), there is strong consensus that governance bridges strategic and operational aspects of software development, and it includes some aspect of project management. Our analysis of the four cases builds around the same structure, addressing first the strategic aspect of governance, and then, the day-to-day activity of project management.

Table 6-4 Perspectives in Defining Governance

Themes	Example Quotes
Bridging Strategy and Execution	''SW product focus / effectiveness"; "Governance is about ensuring corporate expenditures of resources are used effectively and responsibly towards satisfying corporate objectives"; "Control feedback between development and management"
Project Portfolio Management	''allocation of resources for projects"; "monitoring process of projects"; "multi-project management"
Allocation of Decision Rights	''Who decides what? What is right?"; "The structure of managerial relations that are within an organization that is developing software systems"; "responsibility and accountability"
Project Monitoring and Control	''Mechanisms and organization structure to monitor project goals and performance"; "Accountability of SW development activities"; "Monitoring and (adaptive) control of progress towards achieving objectives"

6.5.1 Governance in IndiaCo

IndiaCo's senior leadership has set clear vision for where they expect the organization to go – as a software services company that was started with a focus on product sustainment, the overarching goal remains increasing their share of product ownership, and evolving into a strategic partner for their clients. Their planning process accounts for the variability in their customer's needs/demands. The alignment to a product-services/ownership business model has been further supported through the creation of organizational structures and processes that are tailored to their unique environment. For example, the Onshore-Offshore model for services is extremely well understood; however, the usage of that model in product sustainment was not well understood. IndiaCo created a process (which has since been institutionalized) for knowledge to be transferred from their clients to their transition team. This team is responsible for codifying and internalizing sufficient knowledge to ensure that the transition is invisible to their client's customers. In addition to this startup phase, the offshore-onshore teams coordinate during monthly meetings (some involving their clients) to ensure that projects are, in fact, performing as expected. According to one project manger, this transition to a product services/ownership model remains challenging:

> *"At our core, we are a services company – we are still learning what it takes to do products!"*

The project portfolio itself is designed to provide sufficient variety for IndiaCo to retain its key employees and, at the same time, transition the organization to complete product ownership. From the perspective of partitioning decision rights, IndiaCo's team members are connected to client team members, often of the similar rank/designation, to support rapport building, create a shared understanding of the project state, and ensure that the decisions are made collaboratively, and in a timely manner. They have put into place an escalation mechanism on both the IndiaCo side, as well as the client side to ensure that senior leadership is apprised of issues in a timely manner.

IndiaCo has strong project management capabilities. As an organization that has been assessed to have a maturity at CMMI Level 5, their processes and policies are both well defined and well documented. Even in working with a 'less mature' client, IndiaCo has been successful at tailoring their processes to meet client-specific process requirements, without losing compliance to their internal policy specifications. Their project management system, which also serves as the performance measurement system, is integrated into their quality management system, which makes reporting, analysis, and reflection a part of the daily routine of project managers. Such a system not only supports the traditional functions of monitoring and control, it also serves to identify improvement opportunities, and acts as an effective communication tool.

6.5.2 Governance in EuroTel

EuroTel's challenges from a strategic governance perspective are rooted in the lack of a strategic vision. It is not yet clear, whether they are seen as a source of competitive advantage (which they should be), or as just suppliers of commodities (which they are currently treated as). As a direct result of this, it is not clear how the decision rights are partitioned – EuroTel team members are constrained in terms of the people they can hire. As one developer pointed out:

> *"There are never enough people – the budgets are tight and we cannot hire new people to meet the project needs, but we have budgets for software/tools – hence, our intensive use of automated testing."*

Even in the uncertain environment that EuroTel finds itself in, they have been successful at maintaining a strategically relevant portfolio of projects. This portfolio of projects represents the knowledge that is critical to the long-term sustainability of the EuroTel enterprise.

From a project management perspective, EuroTel has had long standing policy and process specifications for the software lifecycle. This process is very well understood and has been adopted by organizations in other industries, as the development process of choice, for its rigor and the ability to gain insights into understanding how the organization is currently functioning. GameDevCo, for example, adopted the process and then rejected it as being too cumbersome to meet their cycle time requirements. The challenge faced by EuroTel is actually complying with the specified processes – their deviation from the process resulted in the implementation of the two process improvement initiatives to manage design diversity, and the challenges of having to constantly patch software. The emphasis on delivery has resulted in teams optimizing to meet that deliverable, often at the cost of quality and content.

6.5.3 Governance at GameDevCo

GameDevCo is in the process of figuring out how governance should be carried out. Their original business model, which was focused on servicing individual clients, has since been expanded to services and infrastructure provision. Moreover, their interactions with corporate headquarters appear to be 'mandate-based', as opposed to collaborative tailoring. Their transition to Scrum placed the onus of decision making on the development teams themselves. However, the corporate process for project management is built around the traditional monitoring and control paradigm that relies heavily on stable technology roadmaps, and a stage-gated product development process. Further exacerbating the mismatch between the specified process and the enacted process is the lack of mapping between the roles specified in the corporate project management process, and the roles used within GameDevCo. As one senior project manager pointed out:

> *"We have raised this lack of clarity on the roles multiple times, but there has been no action taken by corporate to resolve it."*

Given that the corporate office does not really understand how GameDevCo's processes work, it becomes critical for them to have an effective project management system. The Scrum methodology, by definition, comes with an implicit project management strategy that has a strong emphasis on self-governance, with the Scrum masters (SMs) and Product Owners (POs) acting as the interface to senior leadership. As discussed earlier, there is still no consensus within GameDevCo on exactly what their implementation of Scrum is. In addition to lack of clarity on the process itself, there is significant heterogeneity on the 'who' and the 'how' aspects of governance as well. The internal reporting by project teams is done using burn-down charts, but the translation to the larger enterprise is unclear. Similarly, there is significant overlap in the roles of the SMs, POs and Architects.

6.5.4 Governance in AgileCo

AgileCo believes that they are collaborative problem solvers with their clients. They ensure that their model for providing services is acceptable to the client prior to taking on a services assignment. Their willingness to educate their clients, coupled with a commitment to the agile manifesto's ideal of constantly delivering value to the customers through short development cycles, enables them to align their services model to their operational model. They are limited in their ability to manage their portfolio, particularly in India, where they have been working on projects mostly in the financial services domain. From a decision rights perspective, the clients specify the features that they want in a given project, for a given delivery, and it is up to the project team to be successful. Client inputs are solicited and incorporated into the larger roadmap, but the details of project execution and local decisions are bounded to within the team.

From a day-to-day project management perspective, AgileCo leverages their process to provide self governance at the team level. The project managers act as the medium between the teams and senior leadership to ensure that project execution is not hampered by management issues. As one senior project manager noted:

> *"One of things that I learned to do was to protect the team from the customer and executive management. We told the client that we were going to ideal hours, and instead of being adversarial with my team, I told them to give me ammunition to sell their ideas. Over a couple of iterations, I developed a conversion factor that effectively translated a story point to a standard number of hours. By keeping the story boards updated prior to a standup, I had all the information that I needed for the leadership calls, without having to gather 'overhead' data from my team."*

6.5.5 Discussion

IndiaCo and AgileCo both excel at addressing the governance challenges at the strategic and operational levels. More importantly, they successfully bridge across the two levels to ensure that there is no 'lost in translation' effect that we see in the other two cases, as shown in Table 6.5.

Table 6-5 Analysis of Governance Strategies Across the Four Cases

Governance Analysis	Strategic Governance			Project Management	
	Business Model Alignment	Project Portfolio Management	Decision Rights	Policy/Process	Tool Support
IndiaCo	●	●/○	●	●	●
EuroTel	○	●	○	●/○	●
GameDevCo	○	○	○/●	○	○/●
AgileCo	●	●/○	●	●	●
Legend ● – Strong practices observed ○ – Weak or no practices observed					

IndiaCo and AgileCo are polar opposites with respect to their choice of development processes – yet, their underlying governance mechanisms are the same. They are built on a clear strategic vision to create a system that supports monitoring and control, but more importantly, enables the identification of opportunities for improvement, enhances the motivation to their key stakeholders, and serves as a communication channel. GameDevCo and EuroTel both are in the process of revamping their project management strategies, but they have to address the challenges in strategic governance to ensure that they are sustainable in the long run.

6.6 Theory Development

All four organizations IndiaCo, EuroTel, GameDevCo, and AgileCo use multiple approaches for gaining enterprise agility, as shown in Table 6.6. In the case of IndiaCo, continuous improvement is driven both by their corporate process improvement group, but also by the project teams tailoring processes to meet unique client-specific requirements. When it comes to innovation, IndiaCo has built a system of innovation that leverages the individual contributions of their project team members (through planned events to foster innovation, as well as independently generated ideas) and seamlessly combines those ideas with that generated by their 'innovation' team, and mandates from their corporate offices. The planned innovation activities are funded by IndiaCo and are targeted at client specific problems, as opposed to general capability development and enhancement. In addition to the nature of the work that they do, IndiaCo also leverages globally available capabilities from the human capital perspective.

Table 6-6 Approaches Used to Gain Enterprise Agility

	Software Process Improvement	Creating Systems of Innovation	Leveraging Globally Available Capability
IndiaCo	X	X	X
EuroTel	X		X
GameDevCo	X	X	
AgileCo	X	X	X

They hire personnel globally, continuously scanning to find the best talent available. EuroTel's search for strategic and operational stability has had mixed results: there is a lack

of strategic stability - they are an organization that is still in search of an identity. From the operational perspective, the process harmonization and version control systems are beginning to be institutionalized and will potentially provide stability.

The analysis of the organizational enablers further showed that, in the cases of organizations that were successful, the enablers were present in greater strength than in the cases where it was weak or absent. 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 of people that has to be governed effectively. In the case of leveraging globally available capabilities, stakeholder alignment is critical to prevent fear within the organization. Furthermore, learning and governance are foundational support for long term sustainability.

In summary, our research is built on the premise that software organizations have to evolve in order to remain sustainable in the long run. We observe that this 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, it requires, to some degree, the presence of all four organizational enablers.

It also leads to some hypotheses that need to be tested:

- Can an organization achieve enterprise agility in the absence of any of the four enablers?
- What additional mechanisms can an enterprise leverage to gain enterprise agility?

7 CONCLUSIONS

Enterprise agility: 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.

7.1 Introduction

The research presented in this thesis started with the recognition that the factors governing competitive advantage for software organizations had fundamentally changed in the new millennium. Continued advances in computing technology, reliable telecommunications infrastructure, and increased availability of well-trained, well qualified people globally, have diminished the dominant roles of distance and region-specificity in determining the effectiveness of software organizations. Firms that will thrive and grow in the current environment are those which develop the capabilities needed to be successful under a model of Schumpeterian competition.

By framing the problem of software development as that of finding the unique blend of people, processes, and technology to solve a real-world problem, we shift the research impetus from looking at the technical challenges to addressing the people & process issues. Since software organizations do not fall into a single 'industrial sector' (as seen in the discussion of NAICS code in Chapter 1), the classification into shrink-wrapped products, software services, and software-intensive systems provided sufficient granularity in guiding our selection of case studies. As we progressed along the research, we iteratively defined and refined the key questions that needed to be answered, that would enable us to identify, explore in greater detail, and explain the mechanisms that provided enterprise agility, as seen in Table 7.1.

Table 7-1 Connecting Questions, Contributions and Thesis Structure

Contribution / Question	C1	C2	C3	C4 & C5
Q1 SPI	Chapter 3			Chapter 6
Q2 CSI		Chapter 4		Chapter 6
Q3 LGAC			Chapter 5	Chapter 6

In this chapter, we briefly discuss the mechanisms that software organizations use to develop capabilities, and map how our case studies provide insights into developing enterprise agility. We follow that with an overview of the research approach developed and highlight some of the lessons learned in applying the methodology. We present the findings that answer our research questions, and identify areas for future research.

7.2 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 an organization can build or acquire capabilities: Software Process Improvement; Creating Systems of Innovation; and Leveraging Globally Available Capabilities. 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 set of capabilities in the long run.

Table 7-2 Mapping Case Studies to Mechanisms

Phase	Case Study	SPI	CSI	LGAC
Pilot	BankCo	X		X
	FinServicesCo	X		X
	SpaceCo	X	X	
In-Depth	AgileCo	X	X	X
	GameDevCo	X	X	X
	IndiaCo	X	X	X
	EuroTel	X	X	X

When the organization has a sense of the capabilities that it needs in the future, 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 time investment on the part of the organization. In contrast to the two mechanisms, is the use of 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 in the form of an offshoring effort. 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. Our organization-specific case studies, provided us with useful coverage of the three mechanisms, and supported our theory generation efforts, as shown in Table 7.2.

7.3 Developing Methodology

The theory generating research that we carried out in this thesis requires the mixing of multiple data gathering and analysis approaches to create a rich/thick description of the organizations studied. The rigorous data analysis and cross-context theorizing emphasizes the need for engaging key stakeholders in defining the research problem and ensuring that they are active participants in the research process. This engagement of stakeholders enables us to increase the validity of the findings, and more importantly, construct a value proposition that is beneficial to both the researcher and the organization under study. The mixed methods

approach (shown in Figure 7.1) that we adopted, tested and validated, enabled us to gain access to the research site, obtain useful data, and generate usable theories of software organizations.

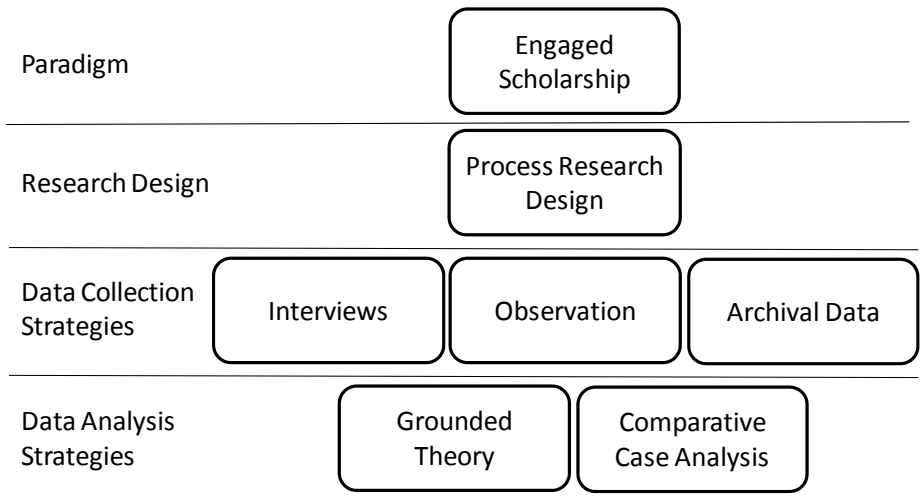


Figure 7-1 Elements of the Research Design

The research approach uses engaged scholarship as the overarching paradigm for execution. The engagement phase involved the researcher setting up meetings with key project stakeholders (especially the project sponsor) to level-set expectations from the study, gain some preliminary insights into the challenges that the organization currently faces. Successful engagement will allows the researcher(s) to more easily explore areas of interest with a select group of interviewees. These interviews were selected through a snowball sampling approach wherein the project sponsor/champion(s) identified key people to interview and facilitate access to them for the exploratory and focused research phases. After the exploration phase was complete, the focused research at the organization's site was governed by the nature of the research question and the availability of resources to execute the research. We recommend a minimum of three rounds (including the exploration phase) to establish organizational context; use focused interviews/observations/archival data analysis to gain deeper insights; and another round of fieldwork for clarification and reflection. The analysis approach combines grounded theorizing at the individual case study level and comparative case analysis at the theory generation stage.

One of our contributions to the software engineering body of knowledge comes in the form of this validated approach. In addition to applying it in this dissertation, it has been used in Sharma (2008) to study goal conflict in software teams, and in Gagne (2009) to study the

dynamics of Free Libre Open Source Software (FLOSS) communities. Our experiences in using this approach led to seven lessons learned: three of which pertained to establishing the context for the research, and the remaining four focusing on the execution of the research.

7.4 Achieving Enterprise Agility

7.4.1 Software Process Improvement

There are two broad streams of improvement approaches that software organizations adopt as a means of building capabilities: top-down, policy-driven approaches (which build off a normative model such as CMMI) and more organic, evolutionary approaches (such as through the adoption of agile methods). Given the extensive research that currently exists on plan-based SPI, we focused on how organizations adopt and sustain the use of agile methods. The two pilot studies of BankCo and FinServicesCo, highlighted the importance of four aspects:

- aligning the agile team to overall organizational objectives,
- the criticality of mentoring in adopting and sustaining agile methods,
- the role of organizational culture,
- and the human capital dimension.

The in-depth cases of AgileCo and GameDevCo provided richer descriptions of those four aspects, and more importantly provided a basis for understanding how software organizations adopt and sustain the use of agile methods. When we compare the two cases against the values and the principles stated in the agile manifesto, we see that the organization that we saw as being successful demonstrated greater adherence to the same.

7.4.2 Creating Systems of Innovation

Software organizations in the software services and software systems sectors often exploit incremental innovations (often obtained through process improvement efforts). The area in which they struggle (unlike their counterparts in the shrink-wrapped software sector) is in the area of radical innovation. In this dissertation, we articulated the strategies that an organization could employ to effectively build a system of innovation by finding the right mix of internal R&D, open innovation, and continuous improvement. The key elements to creating this system of innovation are to have:

- an understanding of the organization's core competencies and capabilities;
- a clearly defined and agreed upon value proposition between the innovation group and the rest of the organization;
- a culture that supports experimentation and learning;

- a mechanism to assess progress .

The challenge with maximizing the effectiveness of this system of innovation lies in addressing the critical bottleneck of transitioning ideas into engineered products and services. In looking at the literature, we found limited frameworks that articulated the dimensions of this particular problem, and provided examples of the problem had been successfully solved. In this thesis, we develop a framework that clearly articulates the dimensions of the problem, ranging from the nature of the problem being solved, to the impact it has on the organization. This framework provides senior managers and leaders with a means of assessing their innovation portfolio, and thereby improving organization performance. In addition to explaining the theory underlying the framework, we provide concrete examples through the discussion of GameDevCo's effort to develop the next generation of its product.

7.4.3 Leveraging Globally Available Capabilities

Every organization today recognizes that it lives in a world where outsourcing and offshoring are the norm not the exception. While the traditional notions of software outsourcing for cost savings were true during and just after the dot-com bubble, the true driver is the availability and exploitability of capabilities. This thesis provides insights into the larger problem of leveraging globally available capabilities in the software product maintenance context. Using the embedded case study of the EuroTel and IndiaCo relationship, we highlight the importance of environmental factors in the extent to which an organization chooses to leverage globally available capabilities.

The EuroTel story highlights the strategic and operational challenges that an organization faces when it makes the decision to outsource. The IndiaCo case provides a set of best practices that most organizations can adopt effectively. The ease of adoption is driven largely by the presence of contextual information about how and why IndiaCo adopted/developed a certain best practice – see for example their challenges with human capital management.

The case also provides insights into the effective management of an outsourcing relationship, and the potential pitfalls to be avoided from both the client side and the supplier side. The workshop approach for building relationships provided a set of tools that every organization could adopt.

7.5 Organizational Enablers for Enterprise Agility

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 to better understand the case studies. Our analysis of the organizational enablers showed that in the cases of successful organizations, the enablers were present to a greater degree than in organizations that were not that

successful. 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.

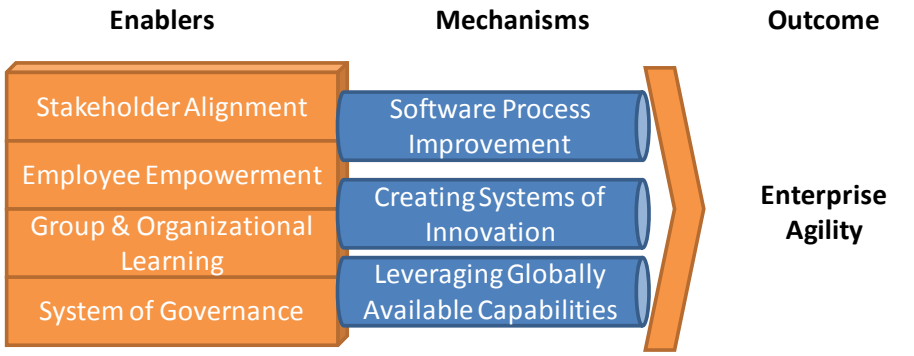


Figure 7-2 Towards a Theory of Enterprise Agility in Software Organizations

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 varying degrees. In summary, the five contributions made in this thesis are:

Contribution 1: *Identified the success factors to enable the adoption and sustainment of agile methods by software organizations*

Contribution 2: *Developed a framework for creating a system of innovation that addresses strategic and tactical issues*

Contribution 3: *Provided a deeper understanding to the evolution of the customer-supplier relationship in maintenance services outsourcing*

Contribution 4: *Identified the organizational antecedents of enterprise agility in software organizations*

Contribution 5: *Proposed a theory of enterprise agility in software organizations*

7.6 Future Work

The work presented here is a first step towards understanding how software organizations gain enterprise agility. 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 into engineered products and services, 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.

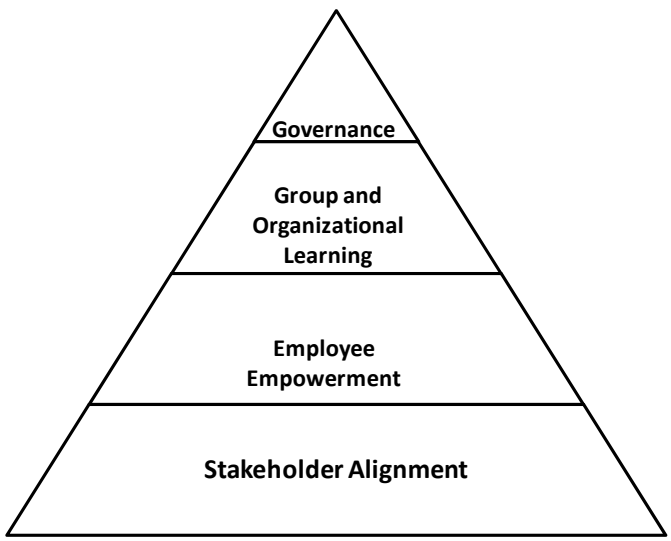


Figure 7-3 Relationship between the Enablers

5. The research approach itself needs to be tested further to see its applicability outside of the contexts in which we have applied it.

Appendix A: Papers Relating to the Thesis

This thesis draws upon eight conference and workshop papers, one book chapter, and one technical report, in addition to two conference papers, and two journal papers that are under review. In this appendix, we present an executive summary of each of the published/accepted papers, and highlight the contributions made as the first author.

Paper #1: Srinivasan, J. (2008). Studying Customer-Supplier Relationships in Global Software Development. Workshop on Studying Work Practices in Global Software Development, ICGSE 2008.

Abstract: The preliminary argument used by most organizations that choose to source their software from other organizations is the perceived cost savings. The nature of the relationship between the customer and supplier has to necessarily evolve in order for it to remain mutually beneficial over the long run, i.e., the 'arms-length' relationship becomes one of strategic partnering. Studying this evolving relationship requires a set of methods that capture the context within which these organizations exist, make explicit the gap (if any) between the actual and articulated nature of the relationship between the two organizations, and can be used to create some useful constructs for managing/ evolving the relationship. This paper illustrates how the paradigm of engaged scholarship has been applied to studying the EuroTel-IndiaCo relationship

Contributions: I am the author of this paper.

Paper #2: Srinivasan, J. (2008). Architecting a Lean Software Enterprise: The TODC Story. Innovation in a Flat World, Special India Conference. Hyderabad, India, Strategic Management Society.

Abstract: The Indian software industry has become a global powerhouse over the last two decades, evolving from supporting lower value segments of the software value chain, to taking complete ownership of key sections of the value chain. Indian software organizations are in the midst of transitioning from competing solely on cost differentials as arms-length suppliers, to becoming strategic partners that are integral to the value proposition of their partner organizations. This transition requires a focused transformation effort to ensure long-term sustainability both in terms of growth rates, as well as profitability. In this paper, we present the seven principles of lean enterprise thinking, and illustrate their applicability through a case study of one of the

telecommunications offshore development centers within a large Indian software services firm.

Contributions: I am the author of this paper.

Paper #3: Srinivasan, J. (2008). Driving Open Innovation: Learning from Three Cases. Workshop on Open Innovation in Services. Cambridge, UK.

Abstract: Open Innovation has become a mantra in recent times with its promise to reduce time to market and increased effectiveness in transitioning ideas into finished products and services. Outside the retail sector, we are yet to see widespread use of open innovation. This paper presents three exploratory case studies on understanding the innovation systems in three knowledge-intensive organizations, drawn from the aerospace, telecommunications and software sectors respectively. Our analysis across the three cases highlights three factors that need to be considered when crafting an innovation strategy: alignment of business model and innovation strategy; leadership and change management in support of the innovation strategy; and last but not least, engaged and empowered employees. We illustrate the presence and absence of these factors in the three case studies, and identify areas for future research.

Contributions: I am the author of this paper.

Paper #4: Srinivasan, J. and K. Lundqvist (2009). Using Agile Methods in Software Product Development: A Case Study. 6th International Conference on Information Technology : New Generations, Las Vegas, IEEE Computer Society.

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.

Contributions: I was the initiator and primary author of this paper. Prof. Lundqvist provided advice and feedback.

Paper #5: Srinivasan, J. and K. Lundqvist (2009). Organizational Enablers for Agile Adoption: Learning from GameDevCo. 10th International Conference on Agile Processesand eXtreme Programming in Software Engineering, XP 2009, Sardinia, Italy, Springer.

Abstract: Adopting agile methods requires an understanding of both the mechanics and the dynamics of value creation in software organizations. From a mechanics perspective, successful agile adoption is about ensuring that project stakeholders are aligned toward a common project objective, employees have the ability to make decisions at the right level of abstraction, that there is effective project management, and an environment exists that supports individual and group learning. The dynamics of value creation require an understanding of organizational-level stakeholders and their value propositions, the development of an organizational learning system, and last but not least, an effective governance strategy. This paper uses the lessons learned a case study of GameDevCo to illustrate these organizational enablers for agile adoption.

Contributions: I was the initiator and primary author of this paper. Prof. Lundqvist provided advice and feedback.

Paper #6: Srinivasan, J., R. Dobrin, K. Lundqvist, (2009). 'State of the Art'in Using Agile Methods for Embedded Systems Development. 2nd International Workshop on Industrial Experience in Embedded Systems Design (IEESD 2009). Seattle, IEEE.

Abstract: Agile methods hold a significant promise to reduce cycle times and provide greater value to all key stakeholders involved in the software ecosystem. While these methods appear to be well suited for embedded systems development, their use has not become a widespread practice. In analyzing the state-of-the-art, as captured in published literature, we found that there are technical issues (requirements management, and testing), as well as organizational issues (process tailoring, knowledge sharing & transfer, culture change, and support infrastructure). In this paper, we build preliminary guidance for firms around these six areas and presented as a framework that will enable understanding the expected adoption trajectory.

Contributions: I was the initiator and primary author of this paper. Dr. Dobrin and Prof. Lundqvist provided advice and feedback.

Paper #7: Srinivasan, J., C. Norström, K. Lundqvist (2009). Exploring the Sources of Enterprise Agility in Software Organizations. 2nd International Engineering Systems Symposium. Cambridge, MA.

Abstract: 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.

Contributions: I was the initiator and primary author of this paper. Prof. Norström and Prof. Lundqvist provided advice and feedback.

Paper #8: Srinivasan, J., A. Löfgren, C. Norström, K. Lundqvist (2009). Lessons Learned from a Workshop on Relationship Building. International Conference on Global Software Engineering, Limerick, Ireland, IEEE Computer Society.

Abstract: Openness and trust are key elements to sustaining any successful client-supplier relationship. When the relationship is transitioning from being arms-length to evolving into a true partnership, it is critical to establish a shared understanding of not only the current state, but also of the expected future state. A workshop organized and facilitated by a neutral party, with the senior leadership of both organizations provides an ideal means for articulating

implicit assumptions and surfacing hidden challenges such that an actionable vision can be created. Using a recent workshop held with both EuroTel and IndiaCo, the key elements of the workshop are discussed, along with the lessons learned. Moreover, this workshop provides further insight into the mechanics of the evolution and governance of outsourcing relationships.

Contributions: I was the initiator and primary author of this paper. Ms. Löfgren created some of the templates used in the workshop, and served as the primary facilitator during the workshop. Prof. Norström and Prof. Lundqvist provided advice and feedback.

Paper #9: Srinivasan, J. (2009). Creating a Lean System of Innovation: The Case of Rockwell Collins. 2nd ISPIM Innovation Symposium,

Abstract: Lean and Innovation have both been touted as transformational strategies that are essential to long term survival of organizations. The question of whether the two constructs can be used simultaneously remains unanswered. This paper is the first step at deriving a theory of lean systems of innovation that combines the notions of lean transformation with that of innovation. We focus on developing a descriptive understanding of the system of innovation that has been deployed at Rockwell Collins. This descriptive understanding enables us to identify the key elements of a strategic system of innovation. By placing human capital at the heart of the framework, we emphasize the importance of knowledge and people to a successful innovation system. The three mechanisms of internal R&D, open innovation and technology scanning have to be managed in concert to ensure maximum value to the enterprise. Most importantly, we emphasize the need to creating a shared value proposition between the innovation team and the rest of the enterprise - this shared value proposition in the presence of a culture that supports innovation and the appropriate rewards and incentives ensures that the innovation architecture is both implementable and sustainable

Contributions: I am the author of this paper

Book: Srinivasan, J. and K. Lundqvist (2010). Agile adoption in India: Success Factors and Barriers, Agile and Distributed. D. J. Smite, P. Agerfalk and B. M. Nils, Springer. (2nd round of review)

Abstract: The family of software development approaches that constitute 'agile methods' have forced a paradigm shift in the mechanics and management of software development. We use the word paradigm shift in the same vein as (Kuhn 1970), because agile methods have resulted in what is essentially an epochal change. The four values and twelve principles espoused in the agile manifesto that was published in 2001 challenge the conventional notions of how software should be developed and managed. In this chapter we contribute to both theory and practice by identifying the success factors and barriers for adoption and sustainment of agile methods by Indian software organizations. Drawing on our fieldwork in Indian software organizations that have adopted agile methods, and triangulating the findings with the published literature, we identify and illustrate success factors such as training and hiring strategies, as well as barriers such as culture.

Contributions: I was the initiator and primary author of this book chapter. Prof. Lundqvist provided advice and feedback.

Appendix B: Research Protocol

Exploring the Sources of Enterprise Agility in Software Organizations

Proposal for Research at the
ORG NAME

Jayakanth Srinivasan
Doctoral Candidate
javakanth.srinivsan@mdh.se

Kristina Lundqvist
Professor of Dependable Software Engineering
kristina.lundqvist@mdh.se

School of Innovation Design and Engineering
Mälardalen University

1/09/2009

Sweden

Introduction

Thank you for your interest in possibly participating in this research project! Your participation in this project will enable you to better exploit sources of enterprise agility by understanding the mechanisms that organization uses to gain sustained competitive advantage, and the associated organizational enablers that support the achievement of that objective. The three mechanisms of interest are software process improvement, creating systems of innovation and leveraging globally available capabilities. In addition to better understanding the mechanisms and enablers in your own organization, you will better understand how your organization compares to other organizations both within and across industry sectors with respect gaining enterprise agility.

This document describes the research we would like to conduct in your organizations. Below is a list of the proposal contents. First, there is a quick overview of the research. Next, the proposed research steps are outlined. Following this, the proposed research tools are highlighted, namely the senior leadership interview questions, the subject interview questions, the background survey and archival data requested. These research tools are open for modification, discussion, and clarification. The final section discusses the legal issues related to this research. This includes information on a proprietary information agreement, an explanation of COUHES, and a consent form for subjects to sign.

Thank you for your time. We look forward to possibly working with you!

Proposal Contents

Research Overview

Exploring the Sources of Enterprise Agility in Software Organizations

Jayakanth Srinivasan, Doctoral Candidate, jayakanth.srinivasan@mdh.se
Kristina Lundqvist, Professor, kristina.lundqvist@mdh.se

Overview

This study explores the sources of enterprise agility in software organizations, wherein enterprise agility is the outcome of an organization's ability to sense changes in its environment, and both efficiently and effectively reconfigure its resources and capabilities to respond to those changes. A better understanding of enterprise agility provides a foundation that senior organization leaders can use in formulating their strategies for achieving sustainable competitive advantage. By explicitly identifying and managing the organizational enablers, they are able to successfully manage and drive change in their organizations. The study seeks both qualitative and quantitative data on how the organizations have achieved (or have not achieved) enterprise agility.

Motivation

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. Our definition of enterprise agility is predicated on the ability of an organization to evolve/adapt its capabilities. Given that the drivers of competition are many, and the associated capabilities equally large, we have bound the scope of the problem. We do so by asking three simple questions relating to:

- How does the organization try to improve the way it does work?
- What does the organization choose to do to create value?
- Who does the work?

Answering these questions lead us to the three mechanisms that we are interested in studying in greater detail: Software Process Improvement, Creating a System of Innovation, and Leveraging Globally Available Capabilities.

Research Protocol

The proposed research aims to carry out comparative case studies across multiple industry segments to identify the mechanisms that software organizations use in practice to gain enterprise agility and determine the organizational enablers that support the generation of enterprise level agility. The mixed methods approach involves both quantitative and qualitative data collection. The quantitative data is gathered through an analysis of past project data in terms of:

- Metrics,
- Process Improvement Strategies and Indicators
- Successful transition of "sand box" development into products,
- Development approach used, and
- Policy regarding process improvement, innovation, and outsourcing/offshoring.

The qualitative data is gathered through interviews with members across all levels of the software development organization, ranging from senior leaders, to project managers, software engineers, and other members. The interviews will be semi-structured, lasting anywhere between 60-90 minutes. The data will be analyzed using a qualitative data analysis tool such as NVivo to identify common themes.

In a series of host enterprises, a point-of-contact from each organization will be contacted to identify a panel of senior leaders in that enterprise. Depending on the opportunities in the company, this senior leadership panel of approximately four individuals will be interviewed individually in order to understand the motivation behind the use of the three mechanisms of software process improvement, systems of innovation, and leveraging globally available capabilities.

Next, the senior leadership panel will be asked to identify interviewees in three groups: (1) project managers, (2) software engineers, and (3) other engineers. Each subject will then be asked to complete a background survey and the investigator will then carry out a detailed interview each with participant. Approximately 75-90 minutes hours will be required of each subject, as the background survey requires 15-30 minutes, and the core interview requires one hour.

Comparisons will be made of differences between the three groups. The interview data and background survey data will generate qualitative information about the current and future states of organization's approach to gain enterprise agility.

Potential Benefits

This research will help companies understand how to gain enterprise agility, wherein agility is defined as the outcome of an organization being able to sense changes in its environment and efficiently and effectively respond to those changes. Companies who participate in this research will better understand how their effective their strategies to gain competitive advantage are, as well as how they compare within and across industry segments.

Proposed Research Steps with Your Company

Note: These suggested steps are flexible and open for discussion.

By telephone and email

1. The researcher identifies a ***point-of-contact*** within the company.
2. The researcher ***introduces the research topic*** to the point-of-contact in the company.
3. The researcher ***signs a non-disclosure agreement*** (if needed) for the handling of company proprietary information.
4. The point-of-contact and the researcher ***identify a senior leadership panel*** in the company. Depending on the opportunities in the company, this expert panel of approximately four individuals may consist of an enterprise leader (someone who is responsible for the overall strategy of the company), two senior leaders (someone discuss the three mechanisms in greater detail), and perhaps a human resource representative who leads the human capital development. Interviews with the expert panel are set up.
5. The researcher ***interviews the members of this senior leadership panel*** to understand each person's background and the organization's strategy for obtaining enterprise agility, as well as the current state successes and challenges.
6. The researcher, company point-of-contact, and the expert panel work together to ***adjust the research tools*** to be used in the company.
7. The researcher, company point-of-contact, and the expert panel ***identify subjects in three groups***: (1) project managers, (2) software engineers, and (3) others. Preferably, this would be a random sample of eligible employees from each group.
8. The researcher, company point-of-contact, and the expert panel ***set-up a site visit***.

On-Site Visit

9. At the company, the researcher has each subject complete ***a background survey***
10. At the company, the researcher ***interviews each subject***. Approximately 75-90 minutes will be required of each subject, since the survey requires 15-30 minutes, and the interview requires one hour.

Back at MDH

11. Back at MDH, the ***researcher assesses the data.*** Results will be combined with data from other companies. The researcher makes comparisons of differences between the three groups, differences between companies, etc. The interview data, background survey data, and comparison data will generate information about the mechanisms and organizational enablers of enterprise agility in software organizations.
12. The aggregate ***findings will be communicated*** back to the host companies by whatever method is preferred by the host company (email, telephone, report, conference presentation, or an on-site presentation to the company).

Research Tools

Following are several of the proposed research tools for use in this study. These tools are open for comments, discussion, and changes.

Senior Leadership Questions

1. Can you tell me a little bit about your background and your evolution into your current role?
2. What are the challenges facing your enterprise with respect to long term sustainability?
3. What have been the major changes in the organization since you have been part of the organization?
4. How do you improve organizational processes?
5. What is the system of innovation in your organization?
6. What is your strategy for outsourcing/offshoring?
7. What are the challenges you are facing with respect to Q3- Q5?
8. What can the organization do better?
9. In your experience, what enablers or barriers have you seen to the organization gaining enterprise agility?
10. In your opinion, what has been effective, and why?
11. Other thoughts, people I should speak to?

Survey of Your Background

Thank you for agreeing to take this survey. Please complete the following survey by entering the correct information in the blank space provided or by checking the appropriate box.

1. Name _____

2. Phone number (for research purposes ONLY) _____

3. Email (for research purposes ONLY) _____

4. What is your gender? ☐ Female ☐ Male

5. What is your current country and state/province of residence? _____

6. What is your level of education? Please check the highest level attained.
   a. _____ High school graduation or GED equivalent
   b. _____ Two-year post-secondary or technical certificate
   c. _____ Bachelor's degree or equivalent
   d. _____ Master's degree
   e. _____ Doctoral degree

7. If applicable, please list the educational institutions you attended and your major(s) at each.

Degree	College Attended	Major(s) and thesis title, if appropriate
Bachelor's degree		
Master's degree		
Doctoral degree		

8. Who is your current employer? _____

9. How many years have you worked for your current employer? _____

10. Please provide a BRIEF title and job description for your current position.

11. Please list the job positions you have held during your career.

Job Position	Company	Length of Time

13-15. Have you participated in the following types of training programs?

Training Program	Yes or No?	Comments	Total Time in Training
13. Training program with job rotations		Years:	
14. Software engineering training program		☐ Internal, company training program ☐ University certificate program ☐ University degree program ☐ Other Year:	
15. Process improvement training, such as Lean or Six Sigma training		Year:	

Subject Interview

Thank you for participating in this interview! During this interview, I hope to learn more about you, your experience with software processes, innovation, and outsourcing, and how these mechanisms have been implemented in your organization.

1. Can you tell me a little bit about your background and your evolution into your current role?
2. What are the challenges facing your enterprise with respect to long term sustainability?
3. What have been the major changes in the organization since you have been part of the organization?
4. How do you improve organizational processes?
5. What do you think are the enablers and barriers to process improvement in your organization?
6. What is the system of innovation in your organization?
7. Why was this system created?
8. How does it currently function?
9. What is your strategy for outsourcing/offshoring?
10. What can the organization do better?
11. In your opinion, what has been effective, and why?
12. Other thoughts, people I should speak to?

Quantitative DATA

Quantitative data is requested in the form of:

a. Metrics used to track project success
b. Published policies regarding software development
c. Historical data on past projects

Legal Issues

Multiple legal issues may need to be addressed during the course of this research. First, the researcher and company may want to sign an agreement on how proprietary information will be handled. Second, the researcher has obtained approval of the research design at the school using an approach similar to MIT's Committee on the Use of Humans as Experimental

Subjects (COUHES). Finally, before each subject participates in the research, a consent form should be signed. These items are described further in the following sections.

7.6.1 Proprietary Information Agreement

If your company would like to have a proprietary information agreement in place, the investigator may sign as an individual. The investigator does not have signature authority to sign anything as a representative of MDH or as a representative of the PROGRESS Center which supports this search at MDH. Additional legal issues may be negotiated with the investigator.

Consent to Participate in Non-Biomedical Research

Exploring the Sources of Enterprise Agility in Software Organizations

You are asked to participate in a research study conducted by the PROGRESS Center at Mälardalen University. You were selected as a possible participant in this study because of your current position in your organization. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

- **PARTICIPATION AND WITHDRAWAL**

Your participation in this study is completely voluntary and you are free to choose whether to be in it or not. If you choose to be in this study, you may subsequently withdraw from it at any time without penalty or consequences of any kind. The investigator may withdraw you from this research if circumstances arise which warrant doing so.

- **PURPOSE OF THE STUDY**

The purpose of this study is to better understand the challenges of gaining enterprise agility in software organizations. Software 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. Our definition of enterprise agility is predicated on the ability of an organization to evolve/adapt its capabilities. Given that the drivers of competition are many, and the associated capabilities equally large, we have bound the scope of the problem to the three mechanisms of software process improvement, systems of innovation and leveraging globally available capabilities.

- **PROCEDURES**

If you volunteer to participate in this study, we would ask you to do the following things.

Please complete the survey. If you volunteer to participate in this study, we would ask you to answer the survey questions to the best of your knowledge. The survey will be administered

to a sample of employees from three groupings: (1) Project managers, (2) software engineers, and (3) others. The survey will take about 15 minutes to complete.Subjects will then be asked to participate in semi-structured interviews. If you volunteer to participate in this study, we would ask you to answer the interview questions to the best of your knowledge.

- **POTENTIAL RISKS AND DISCOMFORTS**

There are no foreseeable risks or discomforts due to participation in this study. All research will be confidential, non-attributable, and reported as aggregate data.

- **POTENTIAL BENEFITS**

Companies who participate in this research could better understand the mechanics and enablers of obtaining enterprise agility. This could lead to more efficient processes, indirectly benefitting employees through improved company performance. This study could further the knowledge in the areas of software engineering, organization design, and strategy.

- **PAYMENT FOR PARTICIPATION**

There is no payment for participation.

- **CONFIDENTIALITY**

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law. Data will be stored securely until research is complete, at which time any attributable data will be destroyed. Reported data will be non-attributable.

- **IDENTIFICATION OF INVESTIGATORS**

If you have any questions or concerns about the research, please feel free to contact:

Jayakanth Srinivasan, jayakanth.srinivasan@mdh.se

or

Prof. Kristina Lundqvist, kristina.lundqvist@mdh.se

- **RIGHTS OF RESEARCH SUBJECTS**

You are not waiving any legal claims, rights or remedies because of your participation in this research study. If you feel you have been treated unfairly, or you have questions regarding your rights as a research subject, you may contact Prof. Christer Norström, christer.norstrom@mdh.se

SIGNATURE OF RESEARCH SUBJECT OR LEGAL REPRESENTATIVE

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

_____

Name of Subject

_____

Name of Legal Representative (if applicable)

_____ _____

Signature of Subject or Legal Representative Date

SIGNATURE OF INVESTIGATOR

In my judgment the subject is voluntarily and knowingly giving informed consent and possesses the legal capacity to give informed consent to participate in this research study.

_____ _____

Signature of Investigator Date

Appendix C: Exemplar Data Capture, Reduction and Analysis

Recognizing that it is not practical to have all of the raw data and intermediate analysis steps taken to arrive at the conclusions incorporated in the thesis document, in this appendix we use illustrative examples from the case studies. These examples are intended to provide snapshots at various phases such that the reader has increased confidence in findings of the thesis, as well as to enable other researchers to leverage some of the techniques. The intent is not to have replicability in the classic sense of the word, rather to present the steps taken in building the argument and make explicit any biases embedded in the analysis.

EuroTel-IndiaCo Relationship

The evolution of the EuroTel-IndiaCo relationship is one of the core elements of Chapter 5 on Leveraging Globally Available Capabilities. While the paradigm of outsourcing/offshoring has been studied in great detail from either the client perspective or the supplier perspective, there have been very few studies that take a bilateral perspective on the issue.

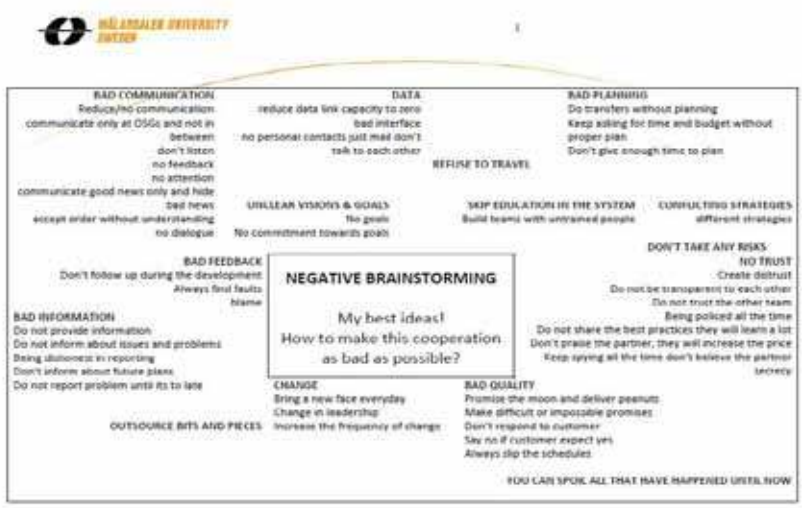


Figure C1 Raw Data Captured from Participants

From a data capturing standpoint, the workshop that was carried out involving both EuroTel and IndiaCo provided codified data from the participants of the exercises in the form of filled

in templates or post-it notes, as well as observational data captured in the form of field notes. At the end of every activity in the workshop, we gathered data from the team members and consolidated them as shown in Figure C1.

"Quotable Quotes"
Negative Brainstorming
EuroTel Senior Leader: *"We can continue to outsource bits and pieces, with different, often conflicting strategies."*
IndiaCo PM: *"one way to kill the relationship is to build teams with untrained people"*
IndiaCo Team Member: *"Communicate only in the steering group meetings, don't provide any feedback or listen to feedback when provided, and hide bad news*."

EuroTel PM: *"Keep spying on them all the time, showing that we don't believe in them."*
IndiaCo Team Member: *"Be policed all the time, and don't praise the partner, because they will raise the price."*
IndiaCo PM: "*promise the moon and deliver peanuts"..."that is probably the easiest way to destroy our trust. As it is, we have to convince other parts of EuroTel (outside Sweden) about our effectiveness"*

Future State Visioning
"We become world class, are learning from each other, we have a common innovation lab and exchange students and technology - we help each other to get additional market share."
"'I am EuroTel not IndiaCo', I treat EuroTel's customers as my customers, rather than as a third party. Every activity is done to enhance value to customer and to make EuroTel successful – in the end we are their most trusted partner."

"IndiaCo will have full product lifecycle management in place, and delivers results at a cost < 80% of the original assignment."
"IndiaCo is helping EuroTel sell our platforms to other customers, always on time and content with quality to exceed expectations."

Figure C2 Notes from Post Exercise Participant Reflections

By consolidation we mean the capturing of all the raw data into a single repository, and not any form of data reduction. During the reflection periods, we captured key insights in the form of quotes (Figure C2) as well as observations from the actual execution of the exercise itself (Figure C3). It was possible after the workshop to go back and verify the observations since one of the workshop assistants had used my camera to record the actual conclusions of the domino game itself. To summarize the four data capturing mechanisms that we used in the workshop:

1. Templates and post-it notes that the participants used during the exercises

2. Quotes from the reflection sessions
3. Observations during the actual execution of the exercises
4. Videos of the domino game

During the dominoes game, we asked the team to self organize into groups of 5-8 people, with the goal of taking the complete bag of dominoes and arranging them into a shape such that when one of the dominoes is toppled, all of the dominoes should topple. The only additional requirement was the arrangement was to be a closed figure - i.e. the starting and ending dominoes must be able to touch each other.

Interestingly enough the group self organized into the senior-leadership team, technical team, and others.

I may be biased here, but the senior leadership team took the most conservative strategy of laying the dominoes on the longer edge down. While they spent a significant amount of time planning what they were going to do (a good thing), their implementation was somewhat sloppy, and even with multiple tries, they could not show success.

The technical team on the other hand took the riskiest approach but did not spend the time planning as well as they should have. Their structure was not well put together and as they were executing the exercise imploded a number of times. Exciting but disastrous

The "other" team consisted of middle managers, trainers, and external observers. They seemed to find a harmonious balance of individuality (they did not try to make just a plain circle), stability, and execution (even they needed three tries to successfully demonstrate completion)

[It was good fun and the teams seemed to bond effectively - pointed out to Annika about the dynamics of self organization - something we need to account for and discuss during the reflections session.]

Figure C3 Observations from the Domino Game

While the dominoes gave us insights into the dynamics of the team, and was a fun story to talk about, it's primary use in this thesis is in the form of providing further support to our theory in terms of stakeholder alignment as an organizational enabler.

Given the raw data, our approach to data reduction was to gain a first past understanding of the data using simple word count based visualizations, such as those provided by Wordle (Figure C4).

Figure C4 Word Count Based Visualization for Data Reduction

1. Create critical mass of management & to involve the pool in various EuroTel as project managers
2. Start common leadership development
3. Identify set of people based on certain criteria and train them
4. all trained managers should be provided opportunity to work as project managers for EuroTel project
5. create a goal that every trained manager will get to practice the learning, for example learning for "x" number of days
6. Develop a strategic joint management recruitment that for example integrates such things as job rotation, etc
7. open up for E managers to work for IndiaCo and IndiaCo managers to work for E on 1-2 year basis
8. Identify the areas where this exchange works best and come out with action plan
9. capacity improvements (competence & people).
10. Identify the competency requirements for each program & create a training plan
11. identify future competency requirements for each program & create a training plan.
12. define concrete action for the next coming project that will contribute to the strategy (product ownership)
13. identify, train & set clear goals for project manager, system manager, quality manager and technical leaders.
14. Develop a plan for the next step towards product ownership.
15. Start growing IndiaCo persons who will take more and more product ownership.
16. Identify the gaps & arrange the necessary training.
17. Identify the gaps & arrange specific technical training programs & bring in guest speakers and other professional competencies
18. Develop core competencies.
19. do proper analysis and then plan for transferring the entire product. Piecemeal approach of only design/innovation should be avoided
20. make a transition plan for the "former" main architects to simplify transfer of the product.
21. prepare classroom training for the competence and have F2F team building session with EuroTel and IndiaCo personnel
22. have continuous/intermittent checkpoints to measure the competence development
23. Identify a component to outsource.
24. Allocate system architects to the organization owning the component. Allocate the product architect to the ordering organization. Explain responsibility.
25. Define proper responsibility and have buy-in at all levels.
26. Ensure that agreed responsibilities are handled and tracked timely and any deviations are communicated timely.
27. Proper action plans are derived from deviations and tracked to closure.
28. Competence engine – a self sustaining mechanism to build, maintain and extend competences.
29. Knowledge transfer from EuroTel to IndiaCo. IndiaCo executes the work and train more people OJT (On the Job Training). Attrition is inevitable but at any point of time knowledge is guaranteed. OJT will ensure knowledge propagation in the team.
30. Have clearly defined transition process with outcome from each process. Have EuroTel management involvement especially when different vendors involved and track the progress of transition. Come up with forecast of attrition and plan properly to get the knowledge transfer. Conduct playback lesion with incoming people to assure whether he is ready to work .
31. Seek the knowledge from all sources and don't wait to set served.
32. Start a technical network with our experts to speed up learning.
33. Identify the knowledge gaps in product. Assess "as is" competency in the teams. Identify the clean list of items to achieve "to be" goal. Work together to build the gaps.
34. Identify core critical mass of people, retain the knowledge capture the knowledge in knowledge base and build KM system. So any new entrant can work on.
35. Trust your own competence and teach each other.
36. Create an EuroTel Greenhouse for competence built up around EuroTel product and processes. Utilize MDH to learn more about coming technologies like design-driven and open innovation.
37. Identify the talents and invest in them ensuring they remain in the organization long-term.
38. To further arrest risk make knowledge sharing sessions so that the organization is not hugely dependent on the individual. However the critical mass of people having the knowledge exists always in the organization.

Figure C5 Raw Inputs for Further Actions

For the first pass of the visualization for future actions, excluding the commonly used English words, the visualization in Figure C4 (upper part allowing for 150 words) highlights *identify,*

knowledge, competence, people, product, organization, work, manager, plan etc. This can be further clarified by reducing the number of words displayed to 25 (lower part of the figure). Yet this only provides a starting point for the data reduction and analysis, not the conclusions.

We take the 38 unique inputs (Figure C5) and cluster them into codes (Figure C6), develop associated memos, and refine the codes during focused coding, we are able to develop the three top level themes (in section 5.3.4) of leadership development, strategy co-creation, and developing competence engines (Figure C7).

Initial Coding	Coding Instances
Management Competences	2
Grow Talent	15
Cross-Train Across the Relationship	11
Identify Talent	3
Understand/Build Competencies	19
Nurture Learning	10
Competence building Mechanisms	1
Connect Strategy to Actions	16
Co Create Strategy	1
Growing human capital	3
Assess performance	1
Strategic innovation plan	2
Responsibility, Authority, and Accountability	3
Proper Plans	7

Figure C6 Initial Coding for Future Vision Action Items

Themes	Coding Clustering
Co-Create Strategy	Strategic Innovation Plan, Co-Create Strategy, Connect Strategy to Actions, Proper Plans, Nurture Learning, RAA, Identify Talent,
Leadership Development	Management Competence, Cross-Train Across the Relationship, RAA, Assess Performance
Competence Engines	Growing human capital, Competence building Mechanisms, Understand/Build Competencies, Identify Talent, Cross-Train Across the Relationship, Grow Talent,

Figure C7 Focused Coding Leading to Themes

Bibliography

1996. Agility and ice cream. *Modern Materials Handling, 51(2)*: 7.

Aaen, I. 2003. Software process improvement: Blueprints versus recipes. *IEEE software*: 86-93.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. 2002. *Agile software development methods Review and Analysis*: VTT Technical Research Center.

Adams, M., Day, G., & Dougherty, D. 1998. Enhancing new product development performance: an organizational learning perspective. *Journal of Product Innovation Management, 15(5)*: 403-422.

Aitken, J., Christopher, M., & Towill, D. 2002. Understanding, Implementing and Exploiting Agility and Leanness. *International Journal of Logistics: Research & Applications, 5(1)*: 59-74.

Allio, R. 2004. CEO interview: the InnoCentive model of open innovation. *Strategy & Leadership, 32(4)*: 4-9.

Alter, A. 2007. Making Agility an Ability. *CIO Insight*: 14-17.

Aoyama, M. 1998. Web-based agile software development. *IEEE software, 15(6)*: 56-65.

Archer, M., Bhaskar, R., Collier, A., & Lawson, T. 1998. *Critical realism: essential readings*: Routledge.

Argote, L., Gruenfeld, D., & Naquin, C. 2001. Group learning in organizations. *Groups at work: Theory and research*: 369-411.

Arora, A., Arunachalam, V., Asundi, J., & Fernandes, R. 2001. The Indian software services industry. *Research Policy, 30(8)*: 1267-1287.

Ashry, N., & Taylor, W. 2000. Requirements Analysis as Innovation Diffusion: A Proposed Requirements Analysis Strategy for the Development of an Integrated Hospital Information Support System. IEEE Computer Society Washington, DC, USA.

Athey, T., Consulting, F., & Collions, F. 1998. Leadership challenges for the future [of the software industry]. *IEEE software, 15(3)*: 72-77.

Athreye, S. 2005. The Indian software industry and its evolving service capability. *Industrial and Corporate Change, 14(3)*: 393-418.

Auerbach, C., & Silverstein, L. 2003. *Qualitative data: An introduction to coding and analysis*: NYU Press.

Baddoo, N., & Hall, T. 2002. Motivators of Software Process Improvement: an analysis of practitioners' views. *The Journal of Systems & Software, 62(2)*: 85-96.

Bailetti, A., & Liu, J. 2003. Comparing software development processes using information theory. *Management of Engineering and Technology*: 309-315.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., & Jeffries, R. 2001. The agile manifesto. *The Agile Alliance*.

Bellinger, A. 2006. The agile trainer is an easy step. *ITTraining*: 30-30.

Berger, D. 2007. Conversations on the Cutting Edge. *Innovate 1st*

Bjelland, O., & Wood, R. 2008. An Inside View of IBM's 'Innovation Jam,'. *MIT Sloan management review*: 1532-9194.

Blackler, F. 1995. Knowledge, knowledge work and organizations: an overview and interpretation. *Organization studies, 16(6)*: 1021.

Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., & Madachy, R. 1998. Using the winwin spiral model: A case study. *Computer, 31(7)*: 33-44.

Boehm, B., & Ross, R. 1989. Theory-W software project management principles and examples. *IEEE Transactions on Software Engineering, 15(7)*: 902-916.

Boehm, B., & Turner, R. 2003. Using risk to balance agile and plan-driven methods. *Computer, 36(6)*: 57-66.

Boehm, B., & Turner, R. 2005. *Balancing agility and discipline: A guide for the perplexed*: Addison-Wesley.

Borchers, G. 2003. The software engineering impacts of cultural factors on multi-cultural software development teams. *25th International Conference on Software Engineering* 540-545.

Brereton, P. 2004. The Software Customer/Supplier Relationship. *Communications of the ACM, 47(2)*: 77.

Breu, K., Hemingway, C. J., Strathern, M., & Bridger, D. 2002. Workforce agility: the new employee strategy for the knowledge economy. *Journal of Information Technology (Routledge, Ltd.), 17(1)*: 21-31.

Briand, L., Eman, K., & Melo, W. 1998. AINSI-An Inductive Software Process Improvement Method: Concrete Steps and Guidelines. University of Maryland.

Brooks, F. 1987. No silver bullet: Essence and accidents of software engineering. *IEEE computer, 20(4)*: 10-19.

Brooks Jr, F. 1995. *The mythical man-month (anniversary ed.)*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Broy, M., Kruger, I., Pretschner, A., & Salzmann, C. 2007. Engineering automotive software. *Proceedings of the IEEE, 95(2)*: 356-373.

Bryant, A. 2002. Grounding systems research: re-establishing grounded theory. 3446-3455.

Carmel, E., & Sawyer, S. 1998. Packaged software development teams: what makes them different? *Information Technology & People, 11(1)*: 7-19.

Charmaz, K. 2006. *Constructing grounded theory: A practical guide through qualitative analysis*: Sage Publications Ltd.

Chatzkel, J., & Saint-Onge, H. 2007. Quantum leap breakthrough performance in acquisitions. *Management Decision, 45(9)*: 1457-1469.

Chesbrough, H. 2003. The era ofopen innovation. *MIT Sloan. Management Review, 44(3)*: 35-41.

Chesbrough, H. 2004. Managing open innovation. *Research-Technology Management, 47(1)*: 23-26.

Chin, G. 2004. *Agile Project Management*: American Management Association International.

Chrissis, M., Konrad, M., & Shrum, S. 2003. *CMMI Guidlines for Process Integration and Product Improvement*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Clark, K., & Wheelwright, S. 1992. *Managing new product and process development: text and cases*: Free Press.

Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., & Little, R. 2002. *Documenting software architectures: views and beyond*: Pearson Education.

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

Cohen, M. 1991. Individual learning and organizational routine: Emerging connections. *Organization Science*: 135-139.

Cohn, M., & Ford, D. 2003. Introducing an agile process to an organization. *Computer, 36(6)*: 74-78.

Collins, L. 2007. Innovation everywhere-Organisations under pressure to innovate are realising that they need to look beyond the R&D department for help, finds Luke Collins. *Engineering Management Journal, 17(2)*: 14-18.

Conger, J., & Kanungo, R. 1988. The empowerment process: Integrating theory and practice. *Academy of Management Review*: 471-482.

Cox, B. 1990. Planning the software industrial revolution. *IEEE software, 7(6)*: 25-33.

Coy, P. 2000. The creative economy. *Business Week, 28*: 75-214.

Cusumano, M., & Selby, R. 1995. *Microsoft secrets*: Free Press New York NY.

Damian, D. 2007. Stakeholders in global requirements engineering: Lessons learned from practice. *IEEE software, 24(2)*: 21-27.

Damian, D., & Zowghi, D. 2002. The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. 319-328.

Desouza, K. 2003. Barriers to effective use of knowledge management systems in software engineering. *Communications of the ACM*.

Diaz, M., Sligo, J., Inc, M., & Scottsdale, A. 1997. How software process improvement helped Motorola. *IEEE software, 14(5)*: 75-81.

Dibbern, J., Winkler, J., & Heinzl, A. 2008. Explaining variations in client extra costs between software projects offshored to India. *Management Information Systems Quarterly, 32(2)*: 333.

Dictionary.com. 2009.

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

Dodgson, M. 1993. Organizational learning: a review of some literatures. *Organization studies, 14(3)*: 375.

Dougherty, D. 1992. Interpretive barriers to successful product innovation in large firms. *Organization Science*: 179-202.

Dove, R. 1996. The voice of industry speaks on agility priorities. *Automotive Production, 108(3)*: 16.

Dove, R. 2001. *Response ability: the language, structure, and culture of the agile enterprise*: Wiley.

Doz, Y., & Kosonen, M. 2008. The Dynamics of Strategic Agility: NOKIA'S ROLLERCOASTER EXPERIENCE. *California Management Review, 50(3)*: 95-118.

Drucker, P. 1985. The discipline of innovation–the innovative enterprise. *Harvard Business Review, 63(3)*: 67–73.

Dyba, T. 2000. An instrument for measuring the key factors of success in software process improvement. *Empirical Software Engineering, 5(4)*: 357-390.

Dyer, L., & Shafer, R. A. 1998. *From human resource strategy to organizational effectiveness: Lessons from research on organizational agility*: Center for Advanced Human Resource Studies, Cornell University, ILR School.

Eisenhardt, K., & Martin, J. 2000. Dynamic capabilities: what are they? *Strategic management journal*: 1105-1121.

El Emam, K., Melo, W., & Drouin, J. 1997. *SPICE: The theory and practice of software process improvement and capability determination*: IEEE Computer Society Press Los Alamitos, CA, USA.

Erdogmus, H., & Favaro, J. 2003. Keep your options open: Extreme programming and the economics of flexibility. *Extreme programming perspectives*: 503.

Etherington, T., Vogl, T., Lapis, M., Razo, J., Inc, R., & Rapids, C. 2000. Synthetic vision information system. *Digital Avionics Systems Conference*.

Ethiraj, S., Kale, P., Krishnan, M., & Singh, J. 2005. Where do capabilities come from and how do they matter? A study in the software services industry. *Strategic management journal, 26(1)*: 25-45.

Ettlie, J., Bridges, W., & O'keefe, R. 1984. Organization strategy and structural differences for radical versus incremental innovation. *Management Science*: 682-695.

Faraj, S., & Sproull, L. 2000. Coordinating expertise in software development teams. *Management Science*: 1554-1568.

Fayad, M., & Laitnen, M. 1997. Process assessment considered wasteful. *Communications of the ACM, 40(11)*.

Fichman, R., & Kemerer, C. 1997. The assimilation of software process innovations: An organizational learning perspective. *Management Science*: 1345-1363.

Fine, C. 1998. *Clockspeed: Winning industry control in the age of temporary advantage*: Basic Books.

Fisher, K. 2007. where is your organization going? *hfm (Healthcare Financial Management), 61(7)*: 100-101.

Fowler, M. 2007. Using an agile software process with offshore development. *Electronic) Avalible:< http://www. martinfowler. com/articles/agileOffshore. html*: 12-07.

Friedman, T. 2005. The world is flat: A brief history of the twentieth century. *Farrar, Straus and Giroux, New York*.

Garnier, J. 2008. Rebuilding the R&D engine in big pharma. *Harvard Business Review, 86(5)*: 68.

George, R., & Labedz, C. 2006. Lean Enterprise Change Case Study - Rockwell Collins. *LAI Working Paper Series*: Massachusetts Institute of Technology.

Gibbs, W. 1994. Software's chronic crisis. *Scientific American, 271(3)*: 72-81.

Glaser, B. 1978. *Theoretical sensitivity: Advances in the methodology of grounded theory*: Sociology Press.

Glaser, B., & Strauss, A. 1967. The discovery of grounded theory: strategies for qualitative research. 1967. *London: Wiedenfeld and Nicholson*.

Goulding, C. 2002. *Grounded theory: a practical guide for management, business and market researchers*: Sage.

Grabowski, H., & Baxter, N. 1973. Rivalry in industrial research and development: An empirical study. *The Journal of Industrial Economics*: 209-235.

Granstrand, O. 1999. Internationalization of corporate R&D: a study of Japanese and Swedish corporations. *Research Policy, 28(2-3)*: 275-302.

Granstrand, O., Bohlin, E., Oskarsson, C., & Sjoberg, N. 1992. External technology acquisition in large multi-technology corporations. *R&D Management, 22(2)*: 111-134.

Grantham, C. E., & Williamson, C. 2007. *Corporate Agility: A Revolutionary New Model for Competing in a Flat World*: AMACOM Div American Mgmt Assn.

Griffin, A. 1997. PDMA research on new product development practices: updating trends and benchmarking best practices. *Journal of Product Innovation Management, 14(6)*: 429-458.

Griffith, T., Sawyer, J., & Neale, M. 2003. Virtualness and knowledge in teams: Managing the love triangle of organizations, individuals, and information technology. *MIS quarterly*: 265-287.

Gumm, D. 2006. Distribution Dimensions in Software Development Projects: A Taxonomy. *IEEE software, 23(5)*: 45-51.

Haley, T., Syst, R., & Marlborough, M. 1996. Software process improvement at Raytheon. *IEEE software, 13(6)*: 33-41.

Hansen, B., & Kautz, K. 2005. Grounded Theory Applied-Studying Information Systems Development Methodologies in Practice. IEEE Computer Society Washington, DC, USA.

Hanssen, G. K., & Fægri, T. E. 2008. Process fusion: An industrial case study on agile software product line engineering. *The Journal of Systems & Software, 81(6)*: 843-854.

Hauser, J., & Zettelmeyer, F. 1996. Metrics to evaluate R, D&E. *MIT Sloan Working Paper Series*.

Henderson, R., & Clark, K. 1990. Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Administrative science quarterly, 35(1)*.

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., & Zubrow, D. 1994. Benefits of CMM-based software process improvement: Initial results. *Software Engineering Institute, CMU/SEI-94-TR-13*.

Herbsleb, J., & Goldenson, D. 1996. A systematic survey of CMM experience and results. 25-30.

Herbsleb, J., & Moitra, D. 2001. Global software development. *IEEE software, 18(2)*: 16-20.

Herrmann, A., & Daneva, M. 2008. Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research. *16th IEEE International Requirements Engineering, 2008. RE'08*: 125-134.

Hines, E., & Srinivasan, J. 2005. IT Enabled Enterprise Transformation: Perspectives Using Product Data Management. *Information Systems Development, ISD 2005*.

Hira, R., & Rosa, P. 2007. The R&D 100. *IEEE Spectrum, 44(12)*: 38-41.

Hoogervorst, J. A. N. 2004. ENTERPRISE ARCHITECTURE:: ENABLING INTEGRATION, AGILITY AND CHANGE. *International Journal of Cooperative Information Systems, 13(3)*: 213-233.

Hoskisson, R., Hitt, M., & Hill, C. 1993. Managerial incentives and investment in R&D in large multiproduct firms. *Organization Science*: 325-341.

Huber, G. 1991. Organizational learning: The contributing processes and the literatures. *Organization Science*: 88-115.

Humphrey, W., Snyder, T., Willis, R., Curtis, B., Bollinger, T., McGowan, C., Leveson, N., Cha, S., Shimeall, T., & Kelly, J. 1997. Software process improvement at Hughes Aircraft. *Applying Software Metrics*: 204.

Huston, L., & Sakkab, N. 2006. Connect and develop. *Harvard Business Review, 84(3)*: 58-66.

Ionesco, D. 2009. T-Mobile myTouch 3G: Too Little, Too Late? : PCWorld.

Jaffe, A. 1986. Technological opportunity and spillovers of R & D: evidence from firms' patents, profits, and market value. *The American Economic Review*: 984-1001.

Jakobsen, A. B. 1998. Bottom up process improvement tricks. *Software, IEEE, 15(1)*: 64-68.

Jones, C. 1996. The economics of software process improvement. *Computer, 29(1)*: 95-97.

Jones, C. 2006. LAI Plenary Conference Key Note Talk.

Jones, C. 2008. Letter to the Shareowners. *Rockwell Collins Annual Report*. Cedar Rapids: Rockwell Collins.

Jones, C. 2009. Innovation makes us stronger. *Horizons*. Cedar Rapids: Rockwell Collins.

Katz, R., & Allen, T. 1982. Investigating the Not Invented Here (NIH) syndrome: A look at the performance, tenure, and communication patterns of 50 R & D Project Groups. *R&D Management, 12(1)*: 7-20.

Kettunen, P. 2007. Extending Software Project Agility with New Product Development Enterprise Agility. *Software Process: Improvement and Practice, 12(6)*.

Kettunen, P. 2009. Adopting key lessons from agile manufacturing to agile software product development—A comparative study. *Technovation, 29(6-7)*: 408-422.

Kivenko, K. 1995. Leading your organization to reduced cycle time and increased agility. *CMA Magazine, 69(5)*: 4.

Krotov, V., & Junglas, I. 2006. Mobile Technology as an Enabler of Organizational Agility. IEEE Computer Society Washington, DC, USA.

Kuhn, T. 1970. The structure of scientific revolutions. 1962. *Chicago: U of Chicago P*.

Lafley, A. G. 2009. How Procter & Gamble Plans to Clean Up. *Business Week*.

Lee, J., Huynh, M., & Hirscheim, R. 2008. An integrative model of trust on IT outsourcing: Examining a bilateral perspecitve. *Information Systems Frontiers, 10(2)*.

Leonard-Barton, D. 1992. Core capabilities and core rigidities: A paradox in managing new product development. *Strategic management journal*: 111-125.

Levinthal, D., & March, J. 1993. The myopia of learning. *Strategic management journal*: 95-112.

Maciag, G. A. 2008. Building Agility. *National Underwriter / Life & Health Financial Services, 112(15)*: A3-A3.

Macvittie, L. 2006. CONTROL AND AGILITY. *Network Computing, 17(15)*: 26-36.

Mafakheri, F., Nasiri, F., & Mousavi, M. 2008. Project agility assessment: an integrated decision analysis approach. *Production Planning & Control, 19(6)*: 567-576.

March, J. 1991. Exploration and exploitation in organizational learning. *Organization Science*: 71-87.

Marchenko, A., & Abrahamsson, P. 2008. Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges. *Agile 2008*: 15-26.

Massey, A., Montoya-Weiss, M., & Tony, M. 2002. Knowledge management in pursuit of performance: Insights from Nortel Networks. *MIS quarterly*: 269-289.

Milis, K., Viaene, S., & Ribbers, P. 2006. On How the Feasibility Study Is Influenced by an ICT Project's Main Trigger. IEEE Computer Society Washington, DC, USA.

Mindrum, C. 2008. Agility Training for the Learning Organization. *Chief Learning Officer, 7(12)*: 36-87.

Nambisan, S., & Sawhney, M. 2008. The Global Brain, Roadmap for Innovating faster and smarter in a networked world. Pearson Education, New Jersey.

Naur, P., & Randell, B. 1968. *Software Engineering: Report on a conference sponsored by the NATO SCIENCE COMMITTEE, Garmisch, Germany, 7th to 11th October 1968*: Scientific Affairs Division.

Nelson, R., & Winter, S. 1982. *An evolutionary theory of economic change*: Belknap Press.

Ngwenyama, O., & Nielsen, P. 2003. Competing values in software process improvement: an assumption analysis of CMM from an organizational culture perspective. *IEEE Transactions on Engineering Management, 50(1)*: 100-112.

Nussbaum, B. 2005. Getting schooled in innovation. *Business Week Online*.

Osborn, C. S. 1998. SYSTEMS FOR SUSTAINABLE ORGANIZATIONS: EMERGENT STRATEGIES, INTERACTIVE CONTROLS AND SEMI-FORMAL INFORMATION. *Journal of Management Studies, 35(4)*: 481-509.

Oskarsson, Ö., & Glass, R. 1995. *An ISO 9000 approach to building quality software*: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Overby, E., Bharadwaj, A., & Sambamurthy, V. 2006. Enterprise agility and the enabling role of information technology. *European Journal of Information Systems, 15(2)*: 120-131.

Pauleen, D., & Harmer, B. 2008. Technology's Transformative Effects on When, Where and How Individuals Work: A Grounded-Narrative Approach. IEEE Computer Society Washington, DC, USA.

Paulk, M., Curtis, B., Chrissis, M., & Weber, C. 1993. Capability maturity model, version 1.1. *IEEE software, 10(4)*: 18-27.

Paulk, M., Weber, C., Curtis, B., & Chrissis, M. 1995. *The capability maturity model: Guidelines for improving the software process*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Perrow, C. 1967. A framework for the comparative analysis of organizations. *American sociological review*: 194-208.

Pfeffer, J. 1977. The ambiguity of leadership. *Academy of Management Review*: 104-112.

Quinn, R., & Spreitzer, G. 1997. The road to empowerment: Seven questions every leader should consider. *Organizational Dynamics, 26(2)*: 37-49.

Qureshi, S., Liu, M., & Vogel, D. 2005. A Grounded Theory Analysis of E-Collaboration Effects for Distributed Project Management. IEEE Computer Society Washington, DC, USA.

Qureshi, S., & Noteboom, C. 2005. An Investigation of Episodes of Global Interactivity: What Collaborative Processes affect the Success of Distributed Projects? *AMCIS 2005 Proceedings*: 8.

Ragin, C. 2007. Comparative Methods. *The Sage Handbook of Social Science Methodology*: 67.

Randolph, W. 2000. Re-thinking empowerment: Why is it so hard to achieve? *Organizational Dynamics, 29(2)*: 94-107.

Rittel, H., & Webber, M. 1973. Dilemmas in a general theory of planning. *Policy sciences, 4(2)*: 155-169.

Rockwell Collins. 2007. Rockwell Collins Technology Search.

Rockwell Collins. 2009a. Renewing the Vision Roadmap.

Rockwell Collins. 2009b. Rewards and Recognition.

Rockwell Collins. 2009c. Top Engineers. *Horizons*. Cedar Rapids: Rockwell Collins.

Roth, A. V. 1996. Achieving strategic agility through economies of knowledge. *Strategy & Leadership, 24(2)*: 30.

Rus, I., & Lindvall, M. 2002. Knowledge management in software engineering. *IEEE software*: 26-38.

Sakson, J. 1997. Managing technical collaboration: methods and mechanisms.

Salo, O., & Abrahamsson, P. 2005. Integrating agile software development and software process improvement: a longitudinal case study. *International Symposium on Empirical Software Engineering*: 10.

Sanchez, R., & Mahoney, J. 1996. Modularity, flexibility, and knowledge management in product and organization design. *Strategic management journal*: 63-76.

Sanjek, R. 1990. *Fieldnotes: The makings of anthropology*: Cornell Univ Pr.

Saran, C. 2005. systems: Simplify systems and speed up IT function to boost business agility. *Computer Weekly*: 40-40.

Scherer, F. 1982. Inter-industry technology flows and productivity growth. *The Review of Economics and Statistics*: 627-634.

Schumpeter, J. 1927. The explanation of the business cycle. *Economica*: 286-311.

Schwaber, K. 1996. Scrum development process. *OOPSLA*.

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

Shafer, R. A., Dyer, L., Kilty, J., Amos, J., & Ericksen, J. 2001. Crafting a human resource strategy to foster organizational agility: a case study. *Human Resource Management, 3*: 197-211.

Sherehiy, B., Karwowski, W., & Layer, J. K. 2007. A review of enterprise agility: Concepts, frameworks, and attributes. *International Journal of Industrial Ergonomics, 37(5)*: 445-460.

Sim, S., Singer, J., & Storey, M. 2001. Beg, borrow, or steal: Using multidisciplinary approaches in empirical software engineering research. *Empirical Software Engineering, 6(1)*: 85-93.

Spradley, J. 1980. *Participant observation*: Holt, Rinehart and Winston New York.

Srinivasan, J. 2008a. Architecting a Lean Software Enterprise: The TODC Story. *Innovation in a Flat World, Special India Conference*. Hyderabad, India: Strategic Management Society.

Srinivasan, J. 2008b. Driving Open Innovation: Learning from Three Cases. *Workshop on Open Innovation in Services*. Cambridge, UK.

Srinivasan, J. 2008c. Studying Customer-Supplier Relationships in Global Software Development. *Workshop on Studying Work Practices in Global Software Development, Colocated with ICGSE 2008*.

Srinivasan, J., Dobrin, R., & Lundqvist, K. 2009. 'State of the Art'in Using Agile Methods for Embedded Systems Development. *2nd IEESD Workshop*. Seattle: IEEE.

Srinivasan, J., Lofgren, A., Norstrom, C., & Lundqvist, K. 2009. Lessons Learned from a Workshop on Relationship Building. *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*: 115-120.

Srinivasan, J., & Lundqvist, K. 2005. Why is Aerospace Software Development and Sustainment Hard? , *LAI Working Paper Series*: Massachusetts Institute of Technology.

Srinivasan, J., & Lundqvist, K. 2007. A Constructivist Approach to Teaching Software Processes. *International Conference on Software Engineering*: 664-672.

Srinivasan, J., & Lundqvist, K. 2009a. Organizational Enablers for Agile Adoption: Learning from GameDevCo. *XP 2009*: 63. Sardinia, Italy: Springer.

Srinivasan, J., & Lundqvist, K. 2009b. Using Agile Methods in Software Product Development: A Case Study. *ITNG 2009*: 1415-1420: IEEE Computer Society.

Srinivasan, J., & Lundqvist, K. 2009c. Using Agile Methods in Software Product Development: A Case Study. *ITNG 2009*.

Srinivasan, J., & Lundqvist, K. 2010. Agile adoption in India: Success Factors and Barriers

In D. J. Smite, P. Agerfalk and B. M. Nils (Eds.), *Agile and Distributed*: Springer.

Srinivasan, J., Norstrom, C., & Lundqvist, K. 2009. Exploring the Sources of Enterprise Agility in Software Organizations. *2nd International Engineering Systems Symposium*. Cambridge, MA.

Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., & Murphy, R. 2007. An exploratory study of why organizations do not adopt CMMI. *The Journal of Systems & Software, 80(6)*: 883-895.

Stevens, G., & Swogger, K. 2009a. Creating a Winning R&D Culture - II. *Research-Technology Management, 52(2)*: 22-28.

Stevens, G., & Swogger, K. 2009b. Creating a Winning RD Culture I. *Research-Technology Management, 52(1)*: 35-50.

Stoker, J., Looise, J., Fisscher, O., & De Jong, R. 2001. Leadership and innovation: relations between leadership, individual characteristics and the functioning of R&D teams. *The International Journal of Human Resource Management, 12(7)*: 1141-1151.

Stopper, W. G. 1998. Motorola: Agility for the Whole Organization. *Human Resource Planning, 21(3)*: 13-13.

Suits, C. G. 1953. Seventy-Five Years of Research in General Electric. *Science, 118(3069)*: 451-456.

Summers, M. 2008. Insights into an Agile Adventure with Offshore Partners. *Agile 2008*: 333-338.

Swafford, P. M., Ghosh, S., & Murthy, N. 2006. The antecedents of supply chain agility of a firm: Scale development and model testing. *Journal of Operations Management, 24(2)*: 170-188.

Swafford, P. M., Ghosh, S., & Murthy, N. 2008. Achieving supply chain agility through IT integration and flexibility. *International Journal of Production Economics, 116(2)*: 288-297.

Szakonyi, R. 1994. Measuring R&D Effectiveness-II. *Research Technology Management, 37(3)*: 44-55.

Takeuchi, H., & Nonaka, I. 1986. The new product development game. *Harvard Business Review, 64(1)*: 137-146.

Teece, D., Pisano, G., & Shuen, A. 1997. Dynamic capabilities and strategic management. *Strategic management journal*: 509-533.

Theunissen, E., Rademaker, R., & Etherington, T. 2001. Synthetic vision-the display concept of the Collins team. *Digital Avionics Systems Conference*.

Theunissen, E., Rademaker, R., Vogl, T., & Postnikov, A. 2001. Synthetic vision: from simulation to flight testing. *Proceedings of the AIAA 2001 MST Conference*.

Thomas, M., & McGarry, F. 1994. Top-down vs. bottom-up process improvement. *Software, IEEE, 11(4)*: 12-13.

Thompson, C. 2005. Role of the CIO. *Computer Weekly*: 3-3.

Tushman, M., & Nadler, D. 1986. Communication and technical roles in R&D laboratories: an information processing approach. *Management of Research and Innovation, 15*.

Tushman, M., & O'Reilly III, C. 2006. Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change. *Managing innovation and change, 1955*: 170.

Van de Ven, A. 1986. Central problems in the management of innovation. *Management Science*: 590-607.

Van de Ven, A. 1992. Suggestions for studying strategy process: A research note. *Strategic management journal*: 169-191.

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

Van de Ven, A., & Johnson, P. 2006. Knowledge for theory and practice. *Academy of Management Review, 31(4)*: 802.

Van Wyk, R. 1997. Strategic technology scanning. *Technological Forecasting and Social Change, 55(1)*: 21-38.

Velocci, A. J. 2009. Top Performing Companies. *Aviation Week & Space Technology*: McGraw Hill.

Von Zedtwitz, M., & Gassmann, O. 2002. Market versus technology drive in R&D internationalization: four different patterns of managing research and development. *Research Policy, 31(4)*: 569-588.

Wall, S. J. 2005. The Protean Organization: Learning to Love Change. *Organizational Dynamics, 34(1)*: 37-46.

Weick, K. E. 1995. *Sensemaking in organizations*: Sage Publications Inc.

Wernerfelt, B. 1984. A resource-based view of the firm. *Strategic management journal*: 171-180.

White, A., Daniel, E. M., & Mohdzain, M. 2005. The role of emergent information technologies and systems in enabling supply chain agility. *International Journal of Information Management, 25(5)*: 396-410.

Williams, L., & Cockburn, A. 2003. Agile software development: It's about feedback and change. *Computer*: 39-43.

Winter, S. 1995. Four Rs of profitability: rents, resources, routines and replication. *Resource-based and evolutionary theories of the firm: Towards a synthesis*: 147-178.

Wojciechowski, J. 2007. Open to innovative ideas. *Horizons*. Cedar Rapids: Rockwell Collins.

Wolfe, R. M. 2008a. NSF Announces New U.S. Business R&D and Innovation Survey National Science Foundation.

Wolfe, R. M. 2008b. U.S. Business R&D Expenditures Increase in 2006; Companies' Own and Federal Contributions Rise National Science Foundation.

Xu, L., & Brinkkemper, S. 2007. Concepts of product software. *European Journal of Information Systems, 16(5)*: 531-541.

Zmud, R. 1984. An examination of'push-pull'theory applied to process innovation in knowledge work. *Management Science*: 727-738.