Regis University

ePublications at Regis University

Regis University Student Publications (comprehensive collection)

Regis University Student Publications

Spring 2010

Capability Maturity Model Integration (Cmmi) for Small Organizations

Kevin Scott
Regis University

Follow this and additional works at: https://epublications.regis.edu/theses

 Part of the Computer Sciences Commons

Recommended Citation

Scott, Kevin, "Capability Maturity Model Integration (Cmmi) for Small Organizations" (2010). *Regis University Student Publications (comprehensive collection)*. 75.
https://epublications.regis.edu/theses/75

This Thesis - Open Access is brought to you for free and open access by the Regis University Student Publications at ePublications at Regis University. It has been accepted for inclusion in Regis University Student Publications (comprehensive collection) by an authorized administrator of ePublications at Regis University. For more information, please contact epublications@regis.edu.

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the "fair use" standards of the U.S. copyright laws and regulations.

Abstract

Software Process Improvement (SPI) is a large area of expertise that deals with software development standard processes and is a progression of proven methods of process improvement from many different methodologies. Personal Software Process (PSP) and Team Software Process (TSP) complement the implementation of Capability Maturity Model Implementation (CMMI) and can be applied gradually from the individual, to the team, and then to the organization. These solutions from Carnegie Melon's Software Engineering Institute (SEI) are leading edge for the field of process improvement. Solutions like this consume many resources, are very complex, require years to implement, and can be costly. The SEI solutions offer an industry standard for SPI. Three case studies were analyzed to provide insight into the benefits of CMMI for small organizations. Decisions that steer these solutions generally involve scheduling, quality, and cost. Depending on the individual needs of an organization, the CMMI technology can fulfill what is required. The example case studies were examined and concluded that given favorable conditions, implementing CMMI is feasible for small organizations.

Acknowledgements

This is for my wife Michelle who will always be my pillar of strength, my rudder, my faith.

Table of Contents

Abstract ii
Acknowledgements iii
Table of Contents iv
List of Figures vii
List of Tables viii
Chapter 1 – Introduction 1
Problem Investigation 1
Niche in the Market 4
Research Questions 4
Goals and Challenge 4
Alternative Proposed SPI Solutions to CMMI 6
Focus SPI for Small Organizations with CMMI 6
Definition of terms 7
Discipline versus Creativity 9
Chapter 2 – Review of Literature and Research 10
Review of All Literature on the Project 10
Software Engineering solutions for SPI 10
Literature on tailoring CMMI with Alternative Methodologies 11
Perception 12
General CMMI and Agile Perception 12
Perceptions of the Different Methodologies 13
Literature and Research Specific and Relevant to this Project 14
Personal Software Process (PSP) and Team Software Process (TSP) 14
CMM 15
CMMI 16
Relevance of CMMI 18
Disadvantages of CMMI 20
Advantages of CMMI 20
Best of Methodologies Concept 21
SPI Standard Commonalities and Similar Processes 21

Different Methodologies Working Together with Microsoft Solutions Framework (MSF) .. 22
Importance and Impact This Study Will Make on Body of Knowledge 23
Chapter Three – Methodology 25
General Approach 25
Research tradition followed 25
Perspective in conducting the research and writing the thesis 25
Example of support for opponents of CMMI 26
Course of the project 26
Determination of whether proposed solution would be beneficial 27
Definition of end product of project: 28
When repeating themes present themselves 28
Determination of whether CMMI is viable for small organizations 28
Chapter Four – Project Analysis and Results 29
SPI in Ireland Based on Grounded Theory Case Study 29
Background of Ireland and projects used 29
Ireland case study synopsis 30
Reason for using this case study 32
Results and analysis of Ireland case study 33
Ireland case study summary 37
NAVAIR Team Software Process (TSP) and CMMI - Case Study 38
Background of NAVAIR case study 38
NAVAIR case study synopsis 38
Reason for using this case study 38
Results and analysis of NAVAIR case study 39
Summary of NAVAIR Case study 45
Systematic – Case Study 45
Background of Systematic 45
Systematic project synopsis 46
Reason for using this case study 46
Results and analysis of Systematic case study 47
Summary of Systematic 49
Case Study Summary 51

Case study answers 51
Chapter Five – Lessons Learned and Next Evolution of the project 56
CMMI for small organizations 56
Alternative methodologies to CMMI opening opportunities 56
Lessons learned 57
Recent Trends 58
Corbis and MSF 58
Next Evolution of the Project 59
Conclusion 60
References 62
Appendix A 66
Annotated Bibliography 67

List of Figures

Figure 1: Used with permission from SEI (Wall, et al., 2007, p. 6). 15

Figure 2: Class A appraisals worldwide documented by the SEI, (2005a, 2005b, 2006a, 2006b, 2006c, 2007a, 2007b, 2008a, 2008b, 2009a) 19

Figure 3: Process evolution and cost of process network with theoretical framework (Coleman & O'Connor, 2008, pp. 777,779,780) 35

Figure 4: Progression of SEI methodologies. 39

Figure 5: Progression of software development life cycle. 39

Figure 6: TSP Coverage of the CMMI Framework (p. 7). Used with permission from SEI. 40

Figure 7: CMMI progression in maturity levels per months (Wall, et al., 2007, p. 20). 44

List of Tables

Table 1: Standish Group's 1994 and 2006 Results (Kannenberg & Saiedian, 2009, p. 2). Used with permission from Kannenberg and Saiedian. 3

Table 2: Glossary of Terms. 8

Table 3: Acronyms. 8

Table 4: CMM Key Process Areas (Paulk, 2001, p. 2). Used with permission from SEI 16

Table 5: CMMI Process Areas (Jones & Soule, 2002, p. 10). Used with permission from SEI. .. 17

Table 6: Themes, core categories, and categories of processes of Coleman and O'Connor (2008, p. 777). Used with permission from Elsevier. 34

Chapter 1 – Introduction

Carnegie Mellon University devotes its Software Engineering Institute (SEI) to discover better methodologies for software development. General principles are meant to benchmark, enhance, and then continuously improve products produced through these methodologies. The Capability Maturity Model Integration (CMMI) is a well established, effective, and proven model for Software Process Improvement (SPI) (Wall, McHale, & Pomeroy-Huff, 2007, p. 22). Its beginnings in the Capability Maturity Model (CMM) and later Humphrey's Personal Software Process (PSP) plus the Team Software Process (TSP) demonstrate a defined history (Wall, et al., 2007, p. 4). Like other industries, the software industry needs standards and guidance to establish a basis for analysis, Measure progress, and provide direction. There are a number of solutions for software development available. However, few solutions that are based on continuous improvement deliver proven results in the marketplace.

Problem Investigation

It is widely thought that CMMI is intended for settings that are big budget, and long lasting, with a command and control emphasis on process and procedures (Anderson, 2005, p. 12). This misperception has kept small software organizations from SPI opportunities (Glazer, 2001, p. 27). The perception is that CMM has not been proven to be cheaper and faster in small settings, which has led to lower adoption rates for CMMI and lost prospects for small organizations (p. 30). Small software organizations do not benefit from economies of scale where large organizations do (Staples, Niazi, Jeffery, Abrahams, & Murphy, 2007, p. 890). Large organizations also have more of a means to develop internal process standard models or implement a standard like CMMI. There is a

myth that CMMI is not intended for small organizations (Glazer, 2001, p. 30; Glazer, Dalton, Anderson, Konrad, & Shrum, 2008, p. 11). Without support of stakeholders and members of the development team, any SPI initiative will have a difficult time getting off the ground. Resources and means are more limited in small software organizations than in large ones. Potential users of CMMI need to be educated regarding the truth that CMMI can be effective in small environments. Guidance and structure of this methodology can aid small organizations with sound development practices.

Development can be more than just routine and complex work (Humphrey, 2005, p. 17). Developers can be engaged in the work and be more empowered to deliver great results through team collaboration. SPI can begin with the individual and then, through the synergy of teams, become more important and impact the organization overall. This impact can be a determining factor to the success or failure of a software organization. By using proven models, refined processes, and practices, methodologies can be used to define, measure, and analyze the development of software (Humphrey, 1995, p. 27). With CMMI, developers should be able to proactively validate SPI initiatives.

Integration problems, consistency, and not enough standards contribute to issues that need to be solved in software development. The Standish Group CHAOS report, a landmark statistical study in the technology industry and otherwise known to describe the "software crisis," has reported findings since the well known 1994 CHAOS report (Jørgensen & Moløkken, 2006, p. 1). A noticeable improvement has occurred in the software industry many years after the original report was published (Abernethy & Piegari, 2007, p. 199). Table 1 shows the documented progress in technology project success since 1994.

Year	Failed Projects	Challenged Projects	Successful Projects
1994	53%	31%	16%
2006	19%	46%	35%

Table 1: Standish Group's 1994 and 2006 Results (Kannenberg & Saiedian, 2009, p. 2). Used with permission from Kannenberg and Saiedian.

A Kannenberg and Saiedian (2009) indicated in a study about traceability that there are still issues and a need for improvement. Implementing project management techniques and improving discipline are noted as top contributors to the improvement since the original 1994 report (p. 2). Business processes also can contribute to the improvement of SPI and lead to advancements.

A request for proposal (RFP) is used in business to detail, formalize, and solicit proposals, and to start procurement procedures for sealed-bid contracts ("Request for Proposal (RFP)," 2009). The beginnings of CMM and what helped spawn the CMM movement originated from an RFP that the Department of Defense (DoD) developed to address the excess money spent on projects (Glazer, et al., 2008, p. 5). This RFP was drafted to identify a solution to the problem with government software projects that were rampant with schedule overruns, failed delivery, and limited functionality. Since that time many advances in SPI occurred, yet the perception of software development needs to be improved. According to Neville, Hoffman, Linde, Elm, and Fowlkes (2008) “50 percent of software projects fail to meet CEO expectations and 42 percent of corporate information technology projects are discontinued before completion” (p. 73). There are similar perceptions and problems that persist today. These problems are the results of software projects not completing on schedule and within budget (Ryan & O'Connor, 2008, p. 236; Verner, Evanco, & Cerpa, 2007, p. 193).

Niche in the Market

A stable foundation for building consistent, cost effective, and timely software is needed for small organizations. Not all developers work in organizations where mature practices are followed (SEI, 2006c, p. 36). The software development life cycle can be difficult to navigate even with SPI initiatives in place. Configuration Management, test cycles, Process Management, and Change Management bring the individual coders, designer, and architects together as a team. Having to reinvent the wheel in this very complex environment can mean making costly mistakes (Cesare, Patel, Iacovelli, Merico, & Lycett, 2008, p. 157). Learning from these mistakes is the key to a successful implementation of SPI. Every software developer and organization is entitled to a stable foundation that can make products predictable, cost effective, and timely. There is a lack of focus on CMMI for small organizations. The purpose of this study is to determine whether small organizations are able to implement CMMI solutions.

Research Questions

In this paper the following research questions will be explored to solve the thesis statement:

1.) What merit do opponents of CMMI have?

2.) What SPI principles promote strong SPI for small organizations?

3.) Can CMMI be implemented for small organizations considering the often limited schedule, budget, and general resource constraints that they have?

Goals and Challenge

There are a variety of different possible CMMI implementations. Projects are on track based on whether they are within budget and on time and whether they produce

strong customer satisfaction. Clark in Galin and Avrahami (2006, p. 82) reported a CMM Level improvement decreased development effort 15-20% and increased productivity 18-26% . The goal of good SPI initiatives is to decrease development costs and increase productivity. There are three objectives in this paper. The primary goal is to empower developers to be their best and achieve more mature practices at an organizational level. The second goal is to reiterate the need for disciplined teamwork in SPI and change the perception that disciplined SPI is exclusive to large projects. The third goal is to promote the use of CMMI within smaller organizations. As a result, more organizations will gain from these improvements. Beginning with developers and teams, organizations can be enlightened to the availability of this CMMI technology and be open to the possibility to benefit from it.

The problem is that not all organizations adopt proven methodologies like CMMI. There is a perception that CMMI is for large organizations (Glazer, et al., 2008, p. 3). Similarly, some Agile methodologies are perceived to be intended for only small organizations (Glazer, 2001, p. 27). Solutions for small organizations using CMMI are the focus of this thesis. In order not to reiterate what is already known about CMMI tailoring, this is not a list on specifically what is entailed in an effective tailoring of CMMI for a small organization. CMMI is open to using other methodologies with its own practices and processes, so these opportunities for small organizations will be explored. Examining the thesis questions will best answer what is needed for small organizations.

Alternative Proposed SPI Solutions to CMMI

There is a trend to better apply and standardize implementations of SPI (Boehm, 2003, p. 5). Jacobson, Ng, and Spence (2007, p. 53) recommend change that is spelled out in simple terms and implemented gradually with minimal disruption to work practices. The goal is to create ways of working more efficiently and effectively. Other software development process alternatives like the waterfall model, Rational Unified Process (RUP), ISO 2001, Agile, and other methodologies all provide SPI (2003, p. 6). CMMI and its use in small settings is the focus of this study.

Focus SPI for Small Organizations with CMMI

CMM, SW-CMM and CMMI are referred to in this paper synonymously because CMMI stems from CMM (Anderson, 2005, p. 2). CMMI is implemented successfully throughout the world in many types of organizations including commercial and government projects (SEI, 2009b, p. 13). Reasons why it is perceived to not be feasible in small settings will be addressed. This paper is about the options available through CMMI for small business and addresses recommendations that are available for projects (Glazer, et al., 2008, p. 20). Different possible solutions are available for any one application of CMMI. This flexibility enables CMMI to fit the needs of a variety of implementations.

To better analyze and narrow the topic of SPI for small business this study, concentrates on three case studies using the CMMI methodology. The first, Coleman & O'Connor (2008) identifies process reasons why small companies are not adopting SPI for small organizations in Ireland. Second, the Naval Air Systems Command (NAVAIR) applies PSP and TSP to realize the growth potential with CMMI (Wall, et al., 2007). Third is Systematic, an organization that implemented Lean Software Development,

Scrum along with CMMI to find a powerful solution for their enterprise (Sutherland, Jakobsen, & Johnson, 2007). This complementary use of SPI focuses on what principles are important in small organizations.

Definition of terms

Since terms used in CMMI may have a different meaning elsewhere, some terms are defined here. Also it is important to note that the definition of a small organization is as small as one or as many as 100 in a development team. The word "predictable" is a good example since the Agilest has a different view of how it is used (Glazer, et al., 2008, p. 10). When referring to the different methodologies some terms are not agreed upon. When comparing the term "process" between CMMI and Agile methodologies, it is not a standard term because the "…agile camp does not use it as such (Jacobson, et al., 2007, p. 41)." When clarity is needed a definition is provided in table 2. Also, table 2 is a summary of acronyms throughout the paper.

Glossary

Terms	**Definition**
Agile	Flexible, iterative processes, with peer reviews which emphasize creativity and collaboration.
Developers	Members of a software organization the produce the work.
Lean Software Development	An Agile methodology where process is the main concern.
Minimum process	The result of process erosion where the organization uses the least amount of methods, activities, practices, and documentation to meet business needs.
Predictable	Defining processes in such a way that they are repeatable and measurable through iteration cycles.
Process	An established and evolving practice contributing to the organization, flow, and improvement of an organization.

Process erosion	The reduction of methods, activities, practices and documentation that due to the organizations adoption of less process over time.
Process inertia	An organizational state where members are apathetic towards process changes.
Product Performance Baseline (PPB)	Used as a metric for projects with Systematic using LOC divided by total hours for a project.
Request for proposal (RFP)	Document that is used in business to detail, formalize, solicit proposals and start procurement procedures for sealed-bid contracts.
Scrum	An Agile methodology where an incremental and iterative schedule is used to manage work that is very complex like software development.
Small software organization	As little as one and as many as 100 employees in a development team.

Table 2: Glossary of Terms.

Acronyms

Acronym	Definition
BPR	Business Process Reengineering
CEO	Chief Executive Officer
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
DoD	Department of Defense
KPA	Key Process Area
MSF	Microsoft Solutions Framework
MTS	Microsoft Team System
NAVAIR	Naval Air Station Command
PIG	Process Improvement Group
PPB	Product Performance Baseline
PSP	Personal Software Process
RFP	Request for Proposal
RUP	Rational Unified Process
SEI	Software Engineering Institute
SLT	Software Leadership Team
SPI	Software Process Improvement
TFS	Microsoft Team Foundation Server
TSP	Team Software Process
VBSE	Value Based Software Engineering

Table 3: **Acronyms.**

Discipline versus Creativity

Development needs to take full advantage of what CMMI has to offer. Without discipline, human error factors into the project causing unpredictability and instability. However, with freedom, people are able to thrive and develop innovative solutions effectively. This proposal is not a combined methodology intended for any organization. This is an examination to find the balance between creativity and discipline to verify that CMMI can work in a small organization. According to Guckenheimer & Perez, "The modern economics require agility with accountability" (2006, p. 3). With any implementation a balance can be found. With CMMI, accountability can be achieved. However, there is a cost. There seems to be a constant play between agility and discipline. Humphrey stated, "Once the creative tasks have been identified and bounded, the routine work often can be made more accurate and efficient" (1995, p. 15). For the developer, in the long run this could free the mind to concentrate on the important task of design, coding, and architectural concerns. Humphrey goes on to point out that repetitive tasks help define what processes are routine and can pinpoint what processes can be refined (p. 16). Discipline is what should attract organizations and developers to strive to be better and continuously improve.

Chapter 2 – Review of Literature and Research

Review of All Literature on the Project

Software Engineering solutions for SPI

Models and standards help us in our everyday activities. There is a great diversity in the SPI solutions that are available. However, a common ground needs to be sought that promotes focused SPI standardization for small organizations. Without this standard, disparate teams and team members may work apart much like some of the software and systems they may support. Barry Boehm (2003) suggests that there is a needed standard in his *Value-Based Software Engineering* publication. He proposes an overall standard software engineering framework emphasizing value to the customer. Boehm goes further to say that exploring ". . . value considerations into all of the existing and emerging software engineering principles and practices, and of developing an overall framework in which they compatibly reinforce each other (p. 1)." These existing frameworks include the Spiral Model, Agile, CMMI, and Rational Unified Process (RUP). Specifically, researchers suggest that differing methodologies like CMMI and Agile can be a good solution for small software organizations, but there are few or no studies on this topic (Glazer, et al., 2008, p. 8; Liu, Chen, Chan, & Lie, 2008, p. 2; Niazi, Wilson, & Zowghi, 2005, p. 156). Furthermore, Jacobson, Ng, and Spence (2007, p. 42) reiterate that there is no current standardized framework solution. With all the possible framework solutions available, CMMI has been developed and is a more widely used, proven, and standardized framework.

Literature on tailoring CMMI with Alternative Methodologies

There are alternatives to using CMMI. According to Yin (2003), alternative perspectives are important to understand the significance of any study (p. 9). Boehm (2003, p. 1) is an important contributor to computer science and offers his Value-Based Software Engineering (VBSE). VBSE is an alternative to CMMI and is also compatible with CMMI (p. 11). The ability of CMMI to be used with other methodologies and its wide acceptance is what is found to be important aside from the ability to tailor CMMI (Paulk, 2001, p. 1). As seen in the practices and implementation flexibility of EssWork (Jacobson, et al., 2007), and the VBSE, CMMI can be implemented using these mixed methodologies. Similarly, a common ground can be found tailoring CMMI (2001, p. 1). Jacobson, Ng and Spence (2007) and also Boehm are cited examples that use CMMI with their methodology. Likewise, CMMI can be bridged to fit different size organizations with varying requirements (Glazer, 2001, p. 28). Jacobson, et al.(2007) and Boehm highlighted the niche where a combined set of methodologies help fill a need with SPI in small organizations.

Tailoring of CMMI is not the focus of this thesis, although the ability to implement CMMI in small organizations is. Tailoring CMMI does offer part of the solution to implementing CMMI for small organizations. There are also obstacles that smaller organizations need to overcome to be successful aside from tailoring. Like with CMMI, Jacobson et al. and Boehm (2003) treat software engineering as a hard science (2003). This can be an obstacle because it offers a level of complexity that would require expertise not necessarily available with the current user base. Additional expertise would be required to implement and maintain these more complex aspects. The “Way of

working together" as described by Jacobson et al. (2007, p. 1) promotes practice that in contrast with Boehm's VBSE, makes the issue of a common SPI methodology more complicated. Einstein in Guckenheimer & Perez stated that a theory ". . . should be as simple as possible, but no simpler (2006, p. 2)." Boehm (2003, p. 1) points out that there is a majority of different practices for "Value-Neutral" environments that are common to software research and practice. With VBSE, Boehm promotes a paradigm shift where widely used frameworks complement one another. He includes a synthesis of data that breaks down the economics and science to a basic level. Software engineering is comprised of different solutions for SPI of which the "Value-Neutral" and VBSE are just two. Most value-neutral approaches track earned value of schedule and cost, while a VBSE project scheduling and cost is tracked to evaluate where the project is at any point in time (2003, p. 3). The experts typically are divisive on the future direction of SPI technologies (Jacobson, et al., 2007, p. 43).

Perception

General CMMI and Agile Perception

It is a common misperception that implementing CMMI for SPI targets large firms and is not feasible for most small size businesses (Glazer, et al., 2008, p. 7). From both a management and worker perspective, CMMI has its own overhead before it produces any returns. The Hawthorne studies noted in Humphrey (1995, p. 15), state that workers produce more using creativity or the ability to work as a free thinking beings rather than when in a regimented routine, treated like machines. Creativity and free thinking of the developer is important to the organization to promote productivity and

innovation (Humphrey, 1995, p. 15). Its availability for smaller organizations should not be limited because of erroneous perceptions and a lack of knowledge.

Perceptions of the Different Methodologies

Agile methodology and CMMI are perceived as incompatible, yet there is debate over the topic. “We assumed that the CMMI world and the Agile world were like oil and water (Anderson, 2005, p. 2).” There are typically two camps, one being Agile and the other being CMMI, when considering SPI (Glazer, et al., 2008, p. 8). The perception is that the two methodologies do not mix. Anderson concludes that this is incorrect, and that the Agile and CMMI methodologies are compatible. In a study by Elshafey and Galal-Edeen (2008) the two methodologies are compared and many of the CMMI key process areas (KPA) are supported by the Agile methodology. This combination is possible and the two methodologies do complement one another. CMMI as a management methodology and Agile as a development methodology approach fit well together (Glazer, 2001, p. 28). Many small organizations do not adopt process standards because the well established standards are thought to be only for large organizations that can afford the cost and have the infrastructure to support the process standards. CMM is perceived to be for large organizations and Agile methodologies are perceived to be undisciplined (p. 27). Once these perceptions are overcome and the truth is revealed, there are multiple options available that are more appropriate to answer whether CMMI is a viable solution for small organizations.

Literature and Research Specific and Relevant to this Project

Personal Software Process (PSP) and Team Software Process (TSP)

As a good framework CMM and CMMI include many resources from which to pull. With the work of Watts Humphrey, the SEI also developed Personal Software Process (PSP) and Team Software Process (TSP). It is important to see how the dynamics of PSP and TSP can add to the CMM and CMMI technology while still being able to work independently.

PSP was developed with the individual developer in mind. Humphrey likens processes to habits (1995, p. 5). Good habits promote good code while bad habits provide unpredictable results. In an article by Gary Gack and Kyle Robinson using Six Sigma, it is emphasized that "At a minimum, the PSP and TSP provide an excellent starting place with respect to definition of a mature software process that can effectively leverage the potential of Six Sigma" (2003, p. 10). Additionally Jim McHale from the SEI stated, "TSP provide an efficient, effective vehicle for implementing CMM-based improvements" (2003, p. 6). These methodologies promote mature SPI through measure, feedback, process management, and identify weaknesses according to Humphrey (1995, pp. 7,9). At the individual level, defect detection or bugs per line of code (LOC) become important when tracking progress (1995, p. 12). Work items are units used to break down a project into more manageable pieces. With these pieces, time estimates can be used to help meet commitments and create an orderly plan and ongoing status (1995, p. 11). Overall, when using PSP, TSP, and CMMI the same principles are used and are apparent. The methodologies can build upon one another.

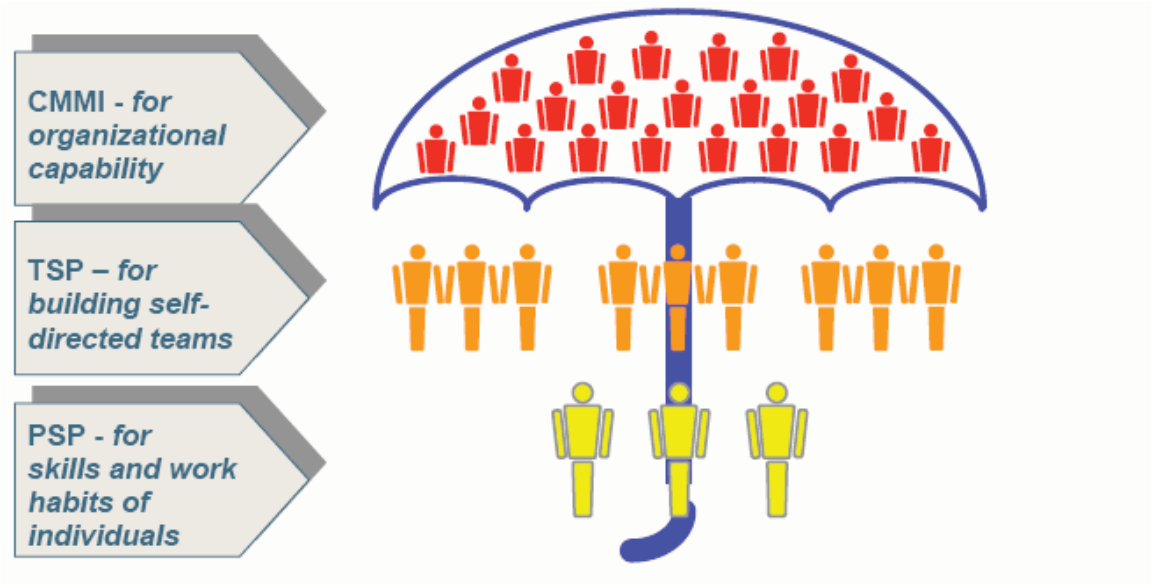


Figure 1: Used with permission from SEI (Wall, et al., 2007, p. 6).

CMM

Staged and continuous model representations are available through CMM and CMMI (SEI, 2006c, p. 10). With the staged model representation, KPA levels are strictly defined for each maturity level. With a staged approach, an appraisal can be approved only if all KPA meet a particular maturity level. Thus, if an organization is rated at Level 3, then it meets Level 3 requirements for all of the KPA. The KPA levels are separated into Process Management, Project Management, Engineering, and Support and can vary in level maturity. Appendix A details a breakdown of the KPA levels and the corresponding maturity levels along with the abbreviations. With the continuous approach an organization can choose which areas to concentrate on. An article in an SEI news bulletin entitled *CMMI Adoption Trends* stated:

> An update and preliminary results, describes the experiences of organizations that have decided to implement CMMI. The 12 case studies in the report, covering organizations such as Accenture, the Boeing Company, General Motors, and Bosch, demonstrate the impact that CMMI-based process improvement has on

> each organization's performance. The case studies feature initial evidence that adoption of CMMI can result in decreased project costs, increased schedule predictability, improved product quality, increased customer satisfaction, and a positive return on investment (Heinz, 2003a, p. 3).

Differences between CMM to CMMI are improvements made to the processes. Analysis and measurement was added to maturity level 2. CMM had a "deferral of measurement issues until higher maturity goals come into sight" (McHale, 2003, p. 11). Standard CMM is a staged approa

Level	Focus	Key Process Area
5 Optimizing	Continuous Process Improvement	Defect Prevention Technology Change Management Process Change Management
4 Managed	Predictable Process	Quantitative Process Management Software Quality Management
3 Defined	Standard, Consistent Process	Organization Process Focus Organization Process Definition Training Program Integrated Software Management Software Product Engineering Intergroup Coordination Peer Reviews
2 Repeatable	Disciplined Process	Requirements Management Software Project Planning Software Project Tracking and Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management
1 Initial	Ad hoc, chaotic	

Table 4: CMM Key Process Areas (Paulk, 2001, p. 2). Used with permission from SEI

CMMI

CMMI supports both staged and continuous approaches for model representations. According to SEI research, in 2001 approximately one third of all CMMI implementations used the continuous approach (Heinz, 2003b, p. 1). "The CMMI Product Suite with its options and flexibility should reach a broader audience and help create a

global community of process improvement for those involved in the development, maintenance, and acquisition of software-intensive systems." (Heinz, 2003a, p. 3) Table 5 shows the CMMI levels with corresponding KPA levels. There is a logical structure and progression from each maturity level. Note the changes from the CMM KPAs and maturity levels.

Level	Focus	Process Area
5 Optimizing	Continuous Process Improvement	Organizational Innovation and Deployment Causal Analysis and Resolution
4 Quantitatively Managed	Quantitative Management	Organizational Process Performance Quantitative Project Management
3 Defined	Standard, Consistent Process	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management for IPPD Risk Management Integrated Teaming Integrated Supplier Management Decision Analysis Resolution Organizational Environment for Integration
2 Managed	Basic Project Management	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
1 Initial		

Table 5: CMMI Process Areas (Jones & Soule, 2002, p. 10). Used with permission from SEI.

Table 5 illustrates five levels that represent the CMMI maturity structure along with the KPA levels. The levels start in ascending order to optimization or level 5. An

organization taking steps to achieve Level 5 makes the processes and practices more proactive than reactive.

To simplify the differences between Agile and CMMI, Glazer uses a metaphor where CMMI is the menu at a restaurant and the Agile methodologies are represented as the recipes (2001, p. 28). He calls this "The pre scriptive vs. the de scriptive." (2001, p. 2). A menu instead of a recipe is how CMMI provides a roadmap on how work can be done. The recipe represents how a project needs to be done with the use of the Agile methodology, while for CMMI, the menu is on a higher level and answers what needs to be done (Glazer, et al., 2008, p. 20). Lewis does not agree with Glazer. Lewis describes CMMI as more of a restaurant critic that critiques the implementation of the model. The analogy of the menu holds true when considering that CMMI is a management methodology and Agile is a development methodology (2001, p. 3). A clear implementation of CMMI with Agile methodologies is with the company Systematic in Sutherland, et al. (2007) where CMMI is used with Lean Software Development and SCRUM (both Agile methodologies). In realizing the complementary nature of CMMI with an Agile methodology, the use of the two together make a synergistic solution and circumvent many of the obstacles and misperceptions of using CMMI in a small setting.

Relevance of CMMI

Every software developer and organization needs a process improvement methodology that can make products predictable, cost effective, timely, and more valuable to the customer. CMMI roots are from CMM that was originally created to fulfill the RFP from the Department of Defense (DoD) (Glazer, et al., 2008, p. 5). CMMI has grown domestically, internationally, commercially, and in the government sectors.

Class A appraisals are used for maturity level ratings ("Capability Maturity Model Integration (CMMI)," 2009, p. 2). Documented by SEI, since March 2005 CMMI Class A appraisals in the United States have increased over four times, while all other countries have had increases in excess of 12 times worldwide (SEI, 2005a, 2005b, 2006a, 2006b, 2006c, 2007a, 2007b, 2008a, 2008b). See figure below:

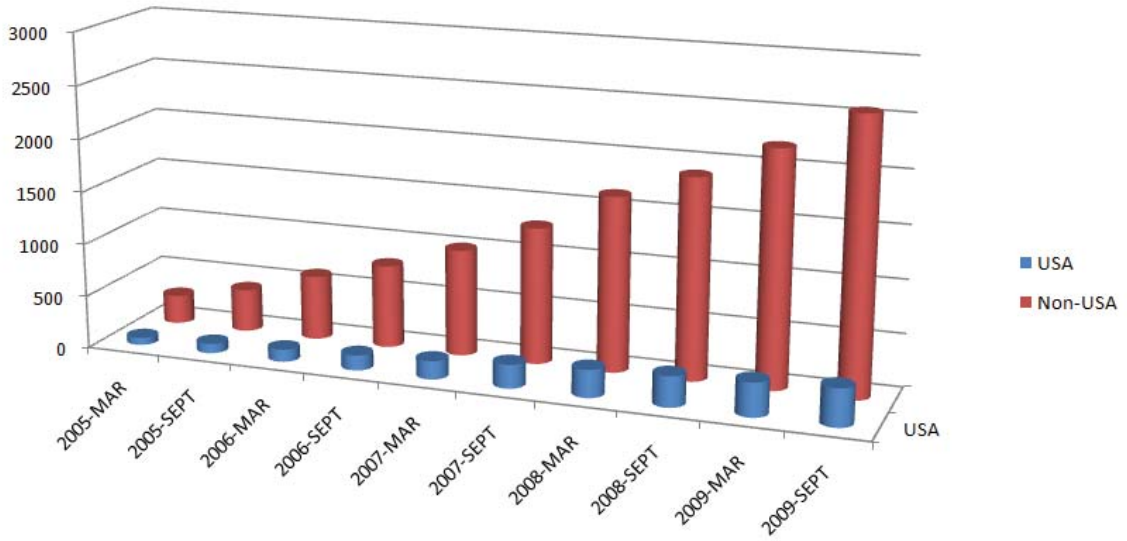


Figure 2: Class A appraisals worldwide documented by the SEI, (2005a, 2005b, 2006a, 2006b, 2006c, 2007a, 2007b, 2008a, 2008b, 2009a)

Countries outside the United States are clearly taking advantage of the CMMI advancements and no evidence was found contrary to this. Off shoring projects is possible with foreign countries like India with their lower cost labor force (Guckenheimer & Perez, 2006, p. 3). Increasingly, CMMI level 5 is a requirement in getting contracts for medical and government organizations (Sutherland, et al., 2007, p. 1). This motivated foreign countries to invest in CMMI. Since CMMI has become such a widely used model worldwide, this paper focuses on SPI from a CMMI perspective.

Disadvantages of CMMI

Large government contracting projects that are mission critical, long lasting, and require auditing and traceability are what is typically thought of when referring to CMMI (Anderson, 2005, p. 2). According to Guckenheimer & Perez (2006, p. 3) disciplined processes like CMMI add complexity that is not necessary and impedes progress on projects. Government Projects with large command and control frameworks typically place a low amount of trust in developers (Anderson, 2005, p. 2; Glazer, et al., 2008, p. 5; Humphrey, 2005, p. 4). Knowing this, developers may consider the traceability and auditing features of disciplined processes to be a threat to personal freedom and creativity. Generally, developers are not open to process, perceiving that CMMI is difficult to adopt and in addition ". . . slows the pace of software development to a frustrating level" (p. 2). Beyond the perceptions at the developer level organizations may require a consultant to implement CMMI along additional integration with disparate systems.

Advantages of CMMI

CMMI certification allows investors to gauge for the status or functional level of the software organization or project. This paper concentrates on focusing on smaller organizations to adopt CMMI in the commercial and government market. CMMI rating appraisal can be utilized where there is a need for improved competitive advantage. Even though CMMI is associated with large projects, it has the capacity for tailoring. This methodology has, on its own, been scalable to small organizations (Paulk, 2001). Recent developments with the Microsoft Solutions Framework (MSF) have also made some of its offerings more accessible for small organizations to adopt CMMI (Anderson, 2005, p.

2). Even though CMMI is not exclusive to government contracts, government restrictions or regulations can have an effect on projects (Ginsberg & Quinn, 1995, p. 7). Actually, in this case, having CMMI is an advantage. The drive of an organization to achieve level 5 and the steps to getting there make it more proactive and even more compliant to requirements.

Best of Methodologies Concept

SPI Standard Commonalities and Similar Processes

The Software Development Life Cycle (SDLC) utilizes process stages that reflect a progressing iterative developing cycle. Don Wells in Glazer (2001, p. 27) states that the XP programming software development process is planning, designing, coding, and testing. These same principles resound in the different life cycles of software development. As mentioned in the introduction, there is a common vision in SPI circles that CMMI and Agile can work together. It is a good precedence toward making processes repeatable and predictable. Is there a better approach to establishing better SPI practices for small organizations? These same principles resound in the CMMI methodology. Humphrey (1995, p. 14) talks about habituation that causes developers to keep a blind eye toward some of what is in the code because they look at the code all of the time and are exposed to it every day. Another person looking at the projects or a different perspective to analyze the code helps resolve this phenomenon. With CMMI, code reviews and peer reviews have been established and are seen to be very valuable in coding practices (Humphrey, 1995, p. 236; Paulk, 2001, p. 3). The concept of pair programming in the XP methodology allows verification of code by two developers

(Paulk, 2001, p. 6; Phongpaibul & Boehm, 2006, p. 2). Both practices are invaluable to developing code even with the extra time and effort involved.

The "undisciplined" methodologies do not take full advantage of learning from experience, while the PSP builds a framework to learn from mistakes and take full advantage of what can be known about software projects. Mark Paulk, who is the primary author of the CMM, stipulates that "XP has disciplined processes, and the XP process is well defined. (Paulk, 2001, p. 8)" These different methodologies are closer together when looking past the misperceptions. With PSP team members as well as individuals within the group can be more consistent and effective overall (Humphrey, 1995, p. 14). After researching this topic, the author finds that a more insightful thesis question is: What combined methodology scenarios (i.e. CMM with Agile) promote the use of CMMI with small organizations?

Different Methodologies Working Together with Microsoft Solutions Framework (MSF)

Early literature like Glacier's paper on dispelling the process myth focuses on the misconceptions of CMMI and how Agile and CMMI can be used together. (2001, p. 1). Similarly, Anderson (2005, p. 1) helped develop the Microsoft Solutions Framework (MSF) that now incorporates a synthesis of Agile and CMMI methodologies. Early publications for both Glazer (Glazer, 2001) and Anderson (Anderson, 2005) come full circle in a publication with the SEI on the combination of the Agile and CMMI methodologies (Glazer, et al., 2008). A publication by Dangle, Larsen, Shaw, and Zelkowitz (2005) indicates that a small software organization can implement CMMI SPI standards even with limited resources.

Glazer argues there is divisiveness in SPI circles where CMM and Agile methodologies are at odds (2001, p. 27). In his article, he points out that the CMM methodology can be tailored and be more like Agile methodologies. Disciplined teams may not realize their potential if they are not innovative and Agile. The teams may not benefit from the flexibility. The literature is highly suggestive that CMMI and Agile methodologies are compatible. Glazer (2001, p. 3) views XP as a development methodology and CMMI as a management methodology and allows both to exist in a scenario, complementing one another, and working together. Paulk in Glazer (2001, p. 29) concludes that the XP methodology rules dictate the same requirements of CMM Level 2 KPA levels and if quantitative statistics and more controls are used even Level 4 could be achieved. XP and CMM ". . . can generate a mutually supportive environment, profitable company, and reliable product" (2001, p. 4). These supportive environments provide a good solution for small organizations to adopt CMMI. Many more organizations would be open to using CMMI if this was known widely.

Importance and Impact This Study Will Make on Body of Knowledge

Like any industry, the organizations in the software industry strive to build quality products, quickly, at minimal cost. SPI like CMMI and Agile are not meant to cure all the problems found in software development organizations (Glazer, et al., 2008, p. 7). Performance and providing a better quality is important to the industry as a whole. From the individual perspective, SPI techniques including the PSP promote individual motivation and organizational practices (Humphrey, 1995, p. 8). Worldwide, the international and domestic interests can be realized with a more available and flexible SPI technologies. Domestic standardization and a larger abundance of appraisal ratings in

the United States help keep projects from being outsourced overseas. Overall, competitive advantage motivates organizations to adopt such models and practices to improve the company performance, reputation and long term success. India's and southeast Asia's ability of acquiring contracts is a result of their adoption of CMMI (Guckenheimer & Perez, 2006, p. 3). Small organizations can have a great impact on the economy of any nation. The Organization for Economic Co-Operation and Development found that ". . . enterprises with fewer than 10 employees represent 93 percent of all companies in Europe and 56 percent in the US—66 percent of total employment" (Laporte, Alexandre, & Renault, 2008, p. 82). Considering the potential number of small organizations, it would be significant for small organizations to improve performance and sustainability. It is not apparent in the literature what are the best available solutions are for small organizations.

Chapter Three – Methodology

General Approach

Research tradition followed

Comparing different views on the subject material provided essential data in determining objective answers. Qualitative research was used in the design of this paper, while engaging in reflexivity for a thorough review of the literature (Leedy & Ormrod, 2005, p. 285). Research done on this thesis was found to be subjective and required the qualitative approach. With an interpretive ontology and epistemology, the information and knowledge needed to be assimilated. Motivation behind the project was due to a positivist methodology to provide benefits to small organizations adopting SPI strategies. Values along with the author's view contributed to the content. Subject material gathered during the research process greatly influenced the outcome of the study.

Perspective in conducting the research and writing the thesis

Sampling of the body of knowledge was done in exploring the research question of there being CMMI solutions available for small software development organizations. In order to develop a comprehensive study, the research material was exhausted. The research first concentrated on SPI standards to focus on what process a small software organization might follow to improve software development. CMMI stood out because of its influence on software development worldwide (SEI, 2009b). The first case study was chosen because of its focus was on the same research question of SPI for small organizations. Other case studies were searched for examples and practical application of CMMI solutions possibly showing that CMMI is viable for small organizations. Conclusions were derived from an iterative process, analysis, study, and sampling of data

found in scholarly literature on the Internet and publications. Ongoing sampling produced case studies on determining whether CMMI solutions do exist that answer the thesis question.

Example of support for opponents of CMMI

Data supporting opponents of CMMI can be found. Staples, et al. (2007, p. 893) studied 40 organizations evaluating CMMI appraisals that decided not to use CMMI. Primary reasons for declining included: CMMI is too costly, not enough time and resources, they are already using another form of SPI, and the organization is too small (2007, p. 893). Continuing on these studies found that it is not feasible for small organizations to use CMMI (2007, p. 893). Opposition to using CMMI in a small organization does exist. While this does not prove the thesis question wrong, it does support it that there is opposition against it. The thesis statement can be proven if case studies demonstrate that CMMI is viable for small development organizations.

Course of the project

Original research was done on topics about process automation, standards, SPI, and CMM/CMMI. It took some time to find the subjects that would answer the thesis questions best. Narrowing the topics became necessary because CMMI and SPI are complex subjects. More insightful questions were explored with the study based on data found after the beginning round of research. In order to identify topics that would solve the question, “What solutions would help small organizations reach a higher maturity?” CMMI was explored. This question was revealing to take a closer look at the root problem. CMM/CMMI focuses on SPI as a software management methodology (Glazer, 2001, p. 3) and is a widely used tool for reaching higher maturity for organizations (SEI,

2009b). The next question, “Does CMMI work for small organizations?” was used to focus further. As stated early in this paper, small organizations are not as capable due to fewer resources, and they do not benefit from economies of scale to implement CMMI like large organizations do (Staples, et al., 2007, p. 8). A good foundation in SPI and CMMI is important first. Automation and better standards are not in the scope of this project.

Determination of whether proposed solution would be beneficial.

CMMI framework was originally built for large organizations. The perception that CMMI is heavy, too disciplined, and intended for large organizations has created a myth that CMMI would not fit with small organizations (Glazer, 2001, p. 1; Glazer, et al., 2008, p. 7). A continuing movement is working to change that perception and the case studies chosen for this thesis were selected to demonstrate the practical application and examples that small organizations not only can but do implement viable solutions of CMMI. The motivation for this project is to provide a path in SPI that can help smaller organizations establish and attain higher maturity levels with CMMI that is repeatable and predictable. CMMI helps solidify the development group as a team and make them work more effectively as a whole (Wall, et al., 2007, p. 5).

Key high level outputs are conclusions to the case studies and literature on the topic. This was done in the following chapters on project analysis and results, and on the next evolution of the project.

Small organizations can improve, while benefitting from the proactive and continued improvement provided by the CMMI technology.

The set of deliverables that this thesis provides are:

1.) Find that adaptability, agility, and creativity can exist with CMMI.

2.) Identify that software creativity and discipline requires balance.

3.) Explore the benefit of using different methodologies with CMMI.

Definition of end product of project:

When repeating themes present themselves

To measure that the project is successful, the research was conducted to answer whether a CMMI is a viable solution and would benefit smaller development organizations. In doing this, as described by Leedy & Ormrod (2005, p. 76), the research at a point became redundant and a sense of finding familiar concepts and repetitive patterns occurred. In researching all aspects of the possible solutions to the questions, an objective conclusion was drawn.

Determination of whether CMMI is viable for small organizations

CMMI is not intended for, nor is it designed for all organizations. The question is whether CMMI will fit within small organizations. The conclusion found after the case studies will answer the thesis statement. Summary and recommendations were determined by the case study analysis and literature. The motivation of this thesis to improve small development organizations SPI options was satisfied.

Chapter Four – Project Analysis and Results

Technologies are tools for us to use on an ongoing basis and stay ahead of the ever changing field of computer science. CMMI is a technology that has potential to grow and provide valuable SPI processes to organizations. In a small organization with a need for as many tools as possible to resolve SPI problems, a combined method would be the best common standard providing flexibility and usefulness. CMMI is not exclusive to large corporations, regulated business, and government contracts. For small organizations to successfully adopt CMMI more often would result in improved productivity and more exposure to the potential of CMMI. Disparate small teams within large organizations can also benefit. Improved performance and effectiveness of teams impact an organization as a whole (Sutherland, et al., 2007, p. 6; Wall, et al., 2007, p. 22). A synergy of CMMI with other methodologies can complement one another and the way SPI can perform. Sampling the body of knowledge to find whether there are viable CMMI solutions for small software development organizations resulted in a positive conclusion. Evidence supports the proposition that CMMI can be used with other methodologies for use within small software organizations.

SPI in Ireland Based on Grounded Theory Case Study

Background of Ireland and projects used

The indigenous Irish software industry is an ideal testing ground for small organizations due to the average size of organizations there. No multinational companies are used that may have influence outside of Ireland (Coleman & O'Connor, 2008, p. 773). Therefore, the projects in the study all fall under the same regulatory and economic rules.

Ireland case study synopsis

Based on grounded theory, Coleman and O'Connor (2008, p. 772) provided a perspective on how the Irish software industry interprets and uses Agile, ISO 9000 and CMMI. In the study, 21 companies are included using 25 interviews in three stages. Of the interviews, 34 questions were asked dealing with human issues, software development strategy, and company background. They conceded that of the data provided on commercial SPI the published data is only a small sampling. Additionally, the Irish software industry is not taking advantage of the highly-publicized SPI models like Agile, ISO 9000, and CMMI. Two research questions are posed to be answered by this study (p. 773):

Why are software companies not using "best practice" SPI models?

What software processes are software companies using?

This Irish study used grounded theory to conduct interviews and to categorize the working processes and the interrelationship between the processes in a software development environment. Once in place, a framework was drafted and helped conceptualize the potential issues to answer to the above questions. Coleman and O'Connor (2008, p. 777) first broke processes into two different categories, essential and non-essential. Essential processes involve requirements, design, and testing. The non-essential processes describe how some managers grouped and prioritized SPI items like planning, estimating, quality documentation, and measurement (p. 780). Over a short term intangible gains in employee empowerment and more organizational intellectual property were improved. In the long term, more tangible gains were quantified by cuts in project costs, productivity, and time to market. Repeatable processes affect future

projects in streamlining projects and also in reducing costs (p. 776). According to this study, rigid SPI measures have a negative impact on process allowing less flexibility and less creativity (p. 783). Short term gains, organizational involvement, team environment, and thorough process were primarily the motivation factors in the SPI decisions. The needs for specific processes were the priority over the value over mid to long term goals.

The Irish study does not praise CMMI too highly, and aside from some simple statistics, does not include a representation of CMMI in the study. CMMI appraisal data 2002-2006 in Coleman and O'Connor (2008, p. 772) suggests that there are relatively few implementations of CMMI in Ireland. This may be due to a lack of small organization reporting and are therefore not accounted for. Small organizations within large companies could be omitted. Also, this study was not set up to focus on CMMI specifically.

CMMI does provide additional processes that go beyond quality management compared to OSI 9000. Also, Coleman and O'Connor (p. 774) report that there are very little published findings on the use of ISO 9000 with software development. Overall, the conclusion of using CMMI was that it was too much for what most small organizations would need. Some organizations do not fit the CMMI methodology where other methodologies like XP may work better.

The perspective of Coleman and O'Connor gives a depth to the research in this thesis. The Coleman and O'Connor (2008, p. 782) findings with the CMMI appraisal data 2002-2006 indicate a different view than what is reported in Chapter 2 of this thesis. In this thesis, the increased use of CMMI in the United States and other countries is reported, whereas, the Coleman O'Connor study suggests that the numbers for CMMI are

very small compared to the total number of software development organizations. In Ireland, the statistics match this perspective where 1.9%, only 10 software companies out of 630, use CMMI. Coleman and O'Connor stipulate that small organizations do not adopt CMMI. The results of strong CMMI use are not conclusive. They do point out that part of this is due to CMMI being a relatively new SPI methodology. Additionally, Wilkie, et al., and Staples et al. in Coleman and O'Connor (p. 774) indicate that the sales, marketing, and SPI approach need improvement. While there is an indication that even though CMMI results are weak in this study, there is still some potential in its use for small organizations.

The reason for standardizing on a SPI methodology like CMMI is to be able to make the processes repeatable and predictable (Coleman & O'Connor, 2008, p. 782). In a large multinational corporation, there are often a number of small software organizations. These units could benefit from processes that are repeatable and predictable among other software organizations within the same corporation. Additionally, employees can reuse these processes. One finding from the Ireland study was that two of the primary influences on current SPI practices were the previous knowledge of the development manager and founder of the organization. The stage that is set for the study does not use any multinational companies and the authors are clear that flexibility and creativity are more in demand than repeatability and predictability in these environments (p. 11). This study is a perfect setting for analyzing the needs of independent small organizations.

Reason for using this case study

The questions posed in the Irish case study are the same as the questions posed in this thesis. Also, this study does develop a framework for processes in small software

development environments. By using grounded theory, the authors of the Ireland study establish theories and conclusions that are fresh and practical. In analyzing the framework closely it becomes very useful to determine why small organizations may not adopt CMMI.

Results and analysis of Ireland case study

The Ireland study concluded to favor less process mainly because process tends not to promote creativity, innovation, and flexibility (Coleman & O'Connor, 2008, p. 780). Additionally, a primary contributor to less process was less documentation which was more cost effective. ISO models are being used more than CMMI according to the study. However, ISO was not favored because of its need for documentation. There is a good argument made for documentation that if an organization has documented its quality systems, then most of the ISO 9000 requirements would be met. This argument does not seem to make any difference for these organizations since cost of process and documentation is the main reason for not adopting the strict controls of ISO 9000 or CMMI. Many managers interviewed in the study believe that documentation is the largest part of the cost of process (p. 780). Best practice SPI methodologies like CMMI were considered overkill and a proactive measure that includes unnecessary costs. In the Irish organizations, the theme of less process was considered over more process. XP promotes less documentation (p. 781). When adopting XP a partial adoption is more common than full adoption of the methodology (p. 774). The culture of the organization has something to do with the SPI adoption that is taken on. ISO 9000 and CMMI are perceived to require too much documentation and to decrease the promotion of innovation and creativity. Admittedly, Coleman and O'Connor explain that the ability to trust

development staff is a major factor in using XP over OSI 9000 or CMMI because it allows for less documentation and oversight (p. 778). Market conditions were also a primary factor on which SPI methodology is used. There was one case where an organization within a regulated industry indicated that they adopted the XP methodology to initiate the project but would use other SPI models like ISO 9000 or CMMI for FDA approvals needed in future implementations of the project (Coleman & O'Connor, 2008, p. 778).

This grounded theory study with its interviews builds a framework for describing the processes of how SPI initiatives work within the Irish organizations. The results of the study start with the concepts, themes, and attributes or categories that are derived through the following table:

Theme	**Categories**
Process Formation	*Background of Software Development Manager*
	Background of Founder
	Management Style
	Process Tailoring
	Market Requirements
Process Evolution	*Process Erosion*
	Minimum Process
	Business event
	SPI Trigger
	Employee Buy-in to Process
	Hiring Expertise
	Process Inertia
Core Category	**Category**
Cost of Process	*Bureaucracy*
	Documentation
	Communication
	Tacit Knowledge
	Creativity Flexibility

Table 6: Themes, core categories, and categories of processes of Coleman and O'Connor (2008, p. 777). Used with permission from Elsevier.

Through a series of questions and interviews, conclusions were drawn with grounded theory. The categories were then depicted in the framework and visual diagrams of how the Irish organizations operate. This is listed in Figure 3.

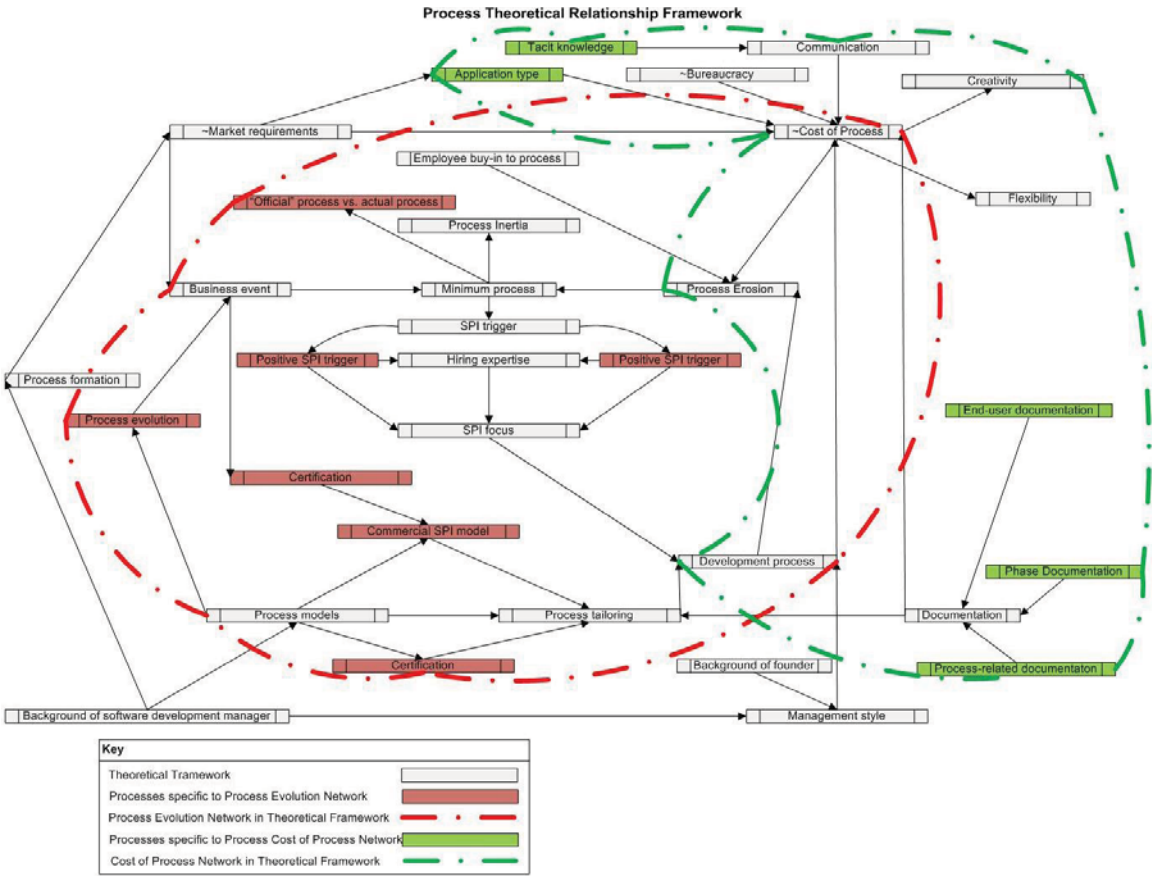


Figure 3: Process evolution and cost of process network with theoretical framework (Coleman & O'Connor, 2008, pp. 777,779,780).

Figure 3 is a testament to the complexity of the processes within a software organization. This is confusing, costly, and difficult to implement due to the complexity. As with the diagrams depicted in the study, all nodes or processes are connected with a precedence operator indicated by an arrow. The precedence operator indicates the parent and successor node where the successor is the node that the arrow indicates. Combinations of categories that were merged to a common name are nodes depicted with a tilde '~' (Coleman & O'Connor, 2008, p. 777). Starting with the theoretical framework from right

to left with nodes in gray, the foundation is established with the different networks included within the colored dotted lines. Every category that is red is specific to the process evolution network. Every category that is green is specific to the cost of process network. The processes within the red dotted line are part of the process evolution network and every process within the green dotted line is part of the cost of process network. Every category that is not inside one of the colored networks is part of process formation. Process inertia occurs when process is ignored and where the organization does not enforce SPI process. After a SPI initiative has been put in place, possible apathy takes place. An interpretation of the diagram is that process inertia stems from the practice of minimum process that then is caused by process erosion. Also, minimum process can cause a SPI trigger. Likewise, a business event could be the cause of minimum process. Due partly to process inertia, most organizations in the study resort to XP because it uses the least amount of process of the three methodologies. Employee buy-in is also important for the SPI methodology used.

The only process category out of place is process tailoring. Coleman and O'Connor (2008) did not place this process in process evolution where it should be. It is interesting why the wide use of the ability to tailor CMMI and the use of CMMI with other methodologies like the Agile methodologies is missing. They do indicate in the study that the technology must be too new.

What is also interesting is that the results do not account for or mention stakeholder or management buy-in. This could be due to the type of culture or nationalism in Irish companies (Coleman & O'Connor, 2008, p. 778). The employees' input may be highly valued in Ireland specifically and used differently in other countries

like the United States. Additionally, the background of the founder in companies that have been bought and sold with new management in place would not fit the process formation.

Ireland case study summary

Questions of why it is difficult for small organizations to adopt CMMI are answered in the Coleman and O'Connor report because of its findings with Irish software development organizations. The conclusion that Coleman and O'Connor draws on CMMI is that "one size fits all" does not work with small organizations (2008, p. 782). As pointed out with other case studies in this paper, it is possible to have CMMI work with small organizations. Since the study includes only data from Ireland which is isolated economically, politically, and geographically, the conclusions that are found are inconclusive. The framework of process themes and categories is the primary output of the result with this study. With this framework, this case study does reveal what small organizations find least appealing about the disciplined approaches. Documentation is the primary cost of process and why managers in the Irish companies generally commented that the cost of additional resources and time was too much. A minimum of documentation is required and less process because it impairs the creativity and flexibility within the organization (2008, p. 780). This might be in part because of cultural differences and influenced geographical limitations of the study. Including this study in this thesis provides a representation of the sampling of data in the body of knowledge. Subsequent uses of reused and continuously improved processes in future processes were not considered due to cost and priority in Ireland.

NAVAIR Team Software Process (TSP) and CMMI - Case Study

Background of NAVAIR case study

The Naval Air Systems Command (NAVAIR) does development, acquisitions, and support of related weapons systems and aircraft for the U.S. Navy and Marine Corps. The case study presented here focuses on two aircraft organizations, including the P3-C Orion aircraft and the AV-8B Harrier aircraft. The original intent of these two projects was for a quick implementation (Wall, et al., 2007, p. 9). The NAVAIR organization is considered a large organization. However, the individual projects that implement the CMMI solution are composed of smaller team organizations.

NAVAIR case study synopsis

A key decision maker in the Software Leadership Team (SLT) for these projects was part of a Business Process Reengineering (BPR) group driving recommendations for NAVAIR. He was a PSP instructor and arranged for Watts Humphrey to brief the SLT on how a quick SW-CMM implementation can be attaining using PSP and TSP because they complement one another (Wall, et al., 2007, p. 9). This was the beginning of the project with a great deal of work for the team members and managers to do in order to attain their goals successfully.

Reason for using this case study

The gradual implementation approach that NAVAIR used to build the SPI process from the ground up using PSP, TSP and then CMMI solved many problems that small organizations typically encounter using CMMI. Implementations of PSP, TSP and CMMI were intended to impact schedule, cost, and quality.

Results and analysis of NAVAIR case study

NAVAIR is an example of bridging the gap between project teams with an organization wide implementation of CMMI. The benefits of CMMI are not just found within the individual teams but across more than one team. The use of PSP and TSP is a prime example of how the CMMI technology can benefit the smaller organization . Many complaints regarding SPI programs is that they take too long to implement, do not reap the benefit originally intended, and lose momentum over time (Wall, et al., 2007, p. 1). Through gradual improvement from PSP, to TSP, and then to CMMI maturity levels the organization can improve with a progressive growth.

PSP ➡ TSP ➡ CMMI

Figure 4: Progression of SEI methodologies.

Also, starting with requirements to delivery and then to the maintenance of the individual, projects follow the natural course of the software development life cycle. The development of software follows a natural progression.

Requirements ➡ Delivery ➡ Maintenance

Figure 5: Progression of software development life cycle.

This gradual progression of CMMI is described by Wall et al. (2007, p. 3) as being a ". . . systemic approach to the problems that most organizations face, such models tend to perpetuate the barriers to improvement that exist in most organizations." Developed first in the CMM, CMMI treats the organization as a whole rather than just as a development effort (p. 3). By keeping the barriers down, the improvement process is more likely to be successful. Management can also be a driver for change and eliminate barriers. CMMI data is not conclusive yet, due to a lack of data since its recent release. Wall, et al. (p. 3) suggest that initial data does suggest that the progression to higher maturity levels is similar to SW-CMM. As outlined by Wall et al. (p. 5) the CMMI framework presents the

"what" for process improvements that can be used through a top-down approach, while TSP and PSP answers the "how" for most of the process areas.

The TSP addresses key goals of the SW-CMM and CMMI methodologies. According to Wall, et al., (p. 6) by using TSP organizations can improve the quality of software, within budget, and on time.

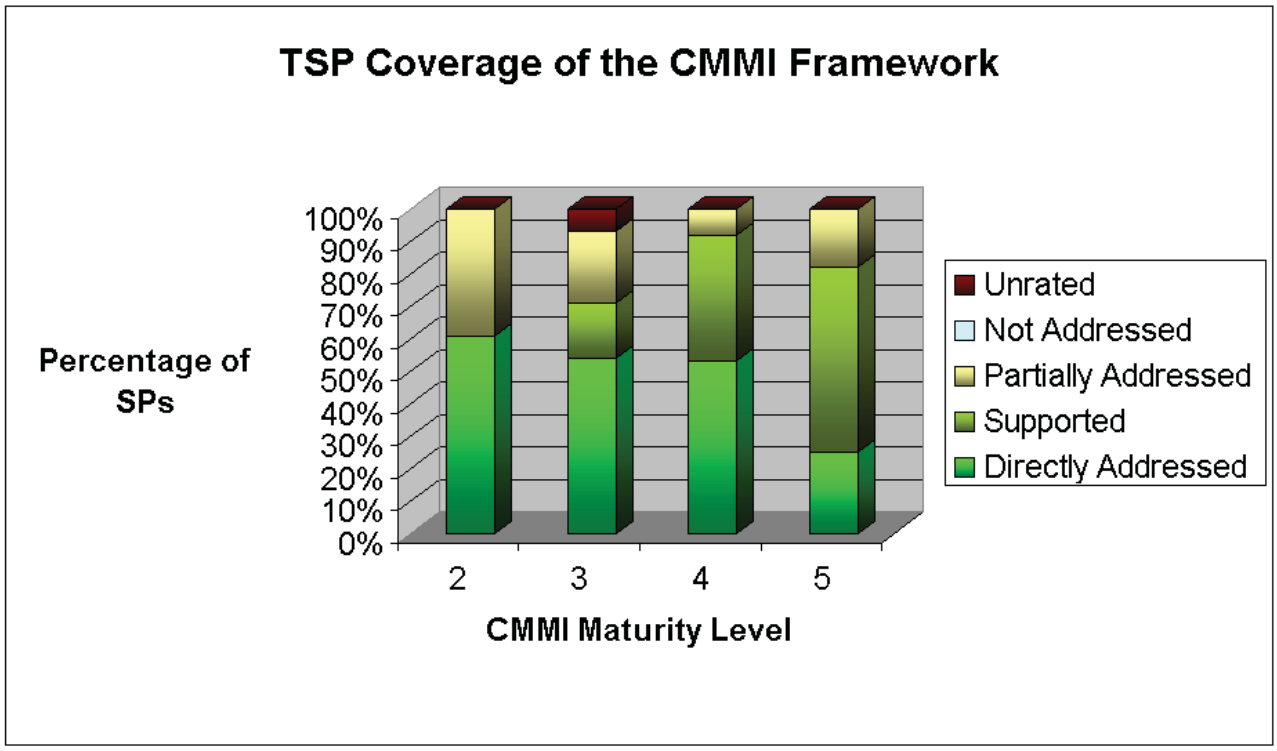


Figure 6: TSP Coverage of the CMMI Framework (p. 7). Used with permission from SEI.

Partially addressed, supported, and directly addressed CMMI software processes were covered through each maturity level. TSP includes a significant portion of the CMMI maturity levels.

NAVAIR committed to SPI initiatives and formed an Integrated Program Leadership Team (IPLT) to help with the transition (Wall, et al., 2007, p. 10). The IPLT was involved in High-Performance Organization (HPO) workshops to focus on conducting a strategic customer-value analysis to ensure values, vision, and leadership

philosophy. Early in the P-3C rollout, the IPLT recognized that the organization was already at maturity Level 2 (Wall, et al., 2007, p. 11). Beyond Level 2, the project was on its way to attaining higher levels of CMMI maturity. To facilitate improvement a Process Improvement Group (PIG) was established to developed and oversee the individual Process Action Teams (PATs) (Wall, et al., 2007, p. 11). With this structure, the organization can better utilize team members while focusing in on the individual CMMI process areas.

There were setback with PATs due to training and a lack of CMMI understanding (Wall, et al., 2007, p. 11). The PIG realized this problem and after reevaluating the situation, it was better able to identify exactly what process areas with TSP overlapped with SW-CMM (Wall, et al., 2007, p. 11). The strong structure and adherence to processes that NAVAIR used required understanding and acceptance at all levels of the organization. This strong involvement of the stakeholders along with results and outcomes of entry and exit criteria added to the success (Wall, et al., 2007, p. 11). An additional factor to the success was "The principal motivator for the development of the TSP was the conviction that engineering teams can do extraordinary work, but only if such teams are properly formed, suitably trained, staffed with skilled members, and effectively led (Wall, et al., 2007, p. 4)." Overall, it was a team effort heavily involving developers and management in the process.

NAVAIR scheduled an executive strategy seminar with stakeholders and a transition planning sessions. An introduction strategy was developed. The planning session and guidelines are as follows (Wall, et al., 2007, pp. 4,5):

1.) To pilot the process 2 to 5 projects were identified.

2.) Projects involve 3 to 15 people.

3.) Adhere to 4-18 month schedule.

4.) Train affected managers, engineers, and support personnel.

5.) Conduct pilot projects and evaluate the results.

6.) Train and authorize an internal PSP/TSP transition team.

7.) Define the introduction goals and responsibilities.

8.) Designate a team to plan and initiate a broad rollout.

9.) Work project by project and launch each one by using TSP.

10.) Build an experience base and train managers, engineers, and other support personnel as needed.

11.) Repeat the introduction steps across the organization.

The foundation of the NAVAIR process plan was in place and linking TSP and CMMI within the organization and institutionalizing the processes were to follow.

Similar to what was posed in chapter 1 of this thesis, Humphrey in Wall et al. posed a question about SW-CMM for module-size software programs: “Can software development teams and individuals apply similar principles to improve their work?” (2007, p. 4). With this question came the PSP and TSP that later NAVAIR used CMMI to catapult its implementation in less than half the time most organizations would have taken. Wall, et al. (p. 4) point out that Humphrey incorporated all of the practices from levels 0 through 5 of the SW-CMM. With the three technologies–PSP, TSP, and CMMI—an organization can incrementally improve, optimizing the effect on projects schedules, cost, and quality.

PSP provides forms, scripts, methods, measures, and standards to better manage, plan, and measure from an individual level (Wall, et al., 2007, p. 4). TSP and CMMI use these same base principles to build a continuous improvement process. “Know thyself,” a concept coined by the ancient Greek philosopher Aristotle is explored with the idea of Knowledge Management (KM): how can employees provide a greater contribution when they have the skills and ability to use and pass on knowledge ("'Know Thyself' is the First Step to Successful Knowledge Management," 2005). This same concept applies in working from an individual level first in the PSP, and TSP, to then improve organizational success with CMMI. The use of PSP and TSP at NAVAIR is a prime example of how the CMMI technology can benefit the smaller organization.

By using TSP with CMMI, NAVAIR expedited the release of projects in the organization. Each individual group, the P3-C, and AV-8 development groups overseen by the SLT of NAVAIR, organized the use of PSP, TSP, and CMMI to accelerate the SPI of their projects. Each group benefitted individually due to the improved cost, quality, and time to market with their software implementation. This was a primary set of goals that were met ahead of SEI norms (See figure 5).

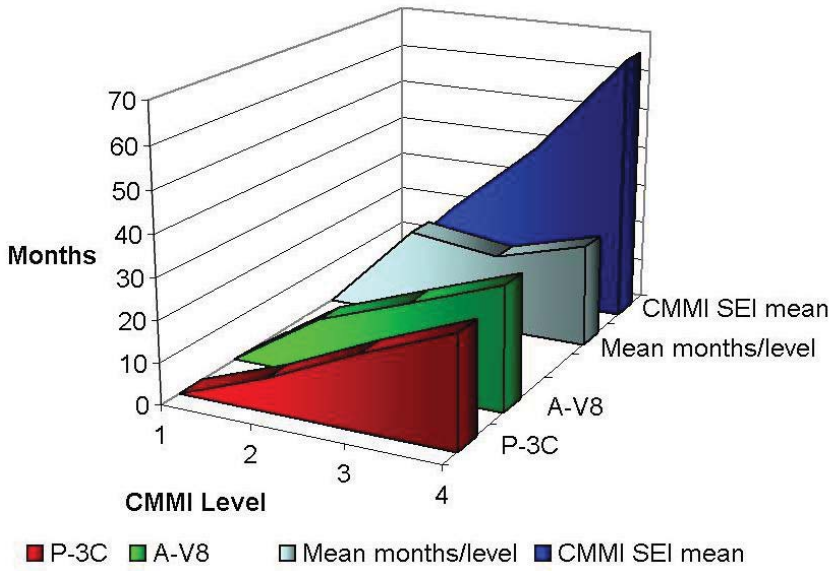


Figure 7: CMMI progression in maturity levels per months (Wall, et al., 2007, p. 20).

Data from the NAVAIR study is conclusive. However, details on the progression from month to month for the individual projects needed to be factored due to there not being data for every level progression. What is important to note is the quick progression to CMMI level 4 in 27 months in the P-3C project (Wall, et al., 2007, p. 10) and 30 months in the A-V3 project (p. 16). The typical mean SEI CMMI implementation is nearly six years.

Contributing reasons for the success of NAVAIR and its processes using TSP with CMMI are abundant. A member in the leadership group had CMMI experience in being a PSP authorized instructor (Wall, et al., 2007, p. 9). There was a quick implementation of TSP methods to progress to CMMI maturity level 4. They built processes intending to reuse them in other parts of the organization. Existing processes used were recognized as already meeting some of the CMMI criteria. Also, buy-in from management and participation with the project teams during planning and implementation was vital.

Summary of NAVAIR Case study

Consultation of CMMI sources is helpful to gain insight that is not apparent in the CMMI documentation. NAVAIR was able to gain SEI resources through a PSP instructor that was already working for the organization. Tracking progress of the SPI methodology is important to show development and management the reason for all the effort. Repeatable processes are essential to facilitate consideration of different factors that may affect the varied outcomes of projects. In large companies, it is essential to reproduce results to subsequent projects and to deliver these same processes to other parts of the organization and to completely different or unrelated projects. NAVAIR concluded that because of existing processes used throughout the organization, many of the requirements of maturity level 2 were already being met (Wall, et al., 2007, p. 11). Appropriate stakeholders participated in the process improvement efforts to ensure the most beneficial input and output. Management was advised to champion these methodologies and the process implementation (p. 16). Otherwise, the SPI solution was in jeopardy of failure (p. 19). Buy-in from both management and team members enabled the projects to operate together with faith in the organization as a whole.

Systematic – Case Study

Background of Systematic

The software company Systematic was established in 1985 focusing on the healthcare, defense, and manufacturing markets (Sutherland, et al., 2007, p. 1). The Agile process methodology Lean Software Development was adopted to drive optimization of processes. Late in 2005, the organization reached CMMI Level 5 maturity. Following this achievement, the company adopted another Agile methodology, Scrum, to assist the

organization to become more adaptable and flexible. Since the adoption of Scrum the organization continued to efficiently deliver software that the customer wants and needs.

Systematic project synopsis

Optimizing value to the customer—rather than developing the largest and most complicated system or program—is the key to successful software development. Scrum supports this concept (p.4). Not surprisingly, the primary goal of Systematic was customer satisfaction. Two different small teams and two different large teams were incorporated into pilots to shift Systematic toward being more nimble and adaptable (Sutherland, et al., 2007, p. 3). The pilots were driven by Lean Software Development, CMMI, and Scrum. Results of these pilots represented the hard work and commitment of Systematic to its customers. Adoption of all methodologies lent to the success of Systematic.

Reason for using this case study

With Level 5 maturity, rework fell 42 percent and 92 percent of all project milestones were on schedule (Sutherland, et al., 2007, p. 1). After Systematic attained CMMI Level 5, implementers decided to use early testing and Scrum for the rest of the project. The case study demonstrates the ability of CMMI and Agile methodologies to complement one another while maintaining the maturity of the organization through continuous process improvement. With these methodologies, years of experience within Systematic was used to develop a set of unified software processes throughout the organization (Sutherland, et al., 2007, p. 1). Cross-functional teams were used and helped in leveraging the SPI experience to other parts of the organization (Sutherland, et al., 2007, p. 4). The organization was able to utilize SPI processes across multiple projects to

keep costs down and better understand the policies, practices, and processes of each project.

Results and analysis of Systematic case study

Frequent delivery of functionality drove the projects forward and focused all stakeholders on the progress of individual deliverables (Sutherland, et al., 2007, p. 3). Delivering functionality on a bi-weekly schedule enabled Systematic to outbid other vendors and deliver on the promise with its resources and know how. The proposal set an expectation of high customer involvement. With this delivery schedule and close workings with the customer, the project was transparent to the customer (Sutherland, et al., 2007, p. 3). The customer was aware of changes and progress throughout the course of the projects. Defects could be found and resolved early on, compared to traditional CMMI implementations. This use of Scrum saved money and cut rework by 50% (Sutherland, et al., 2007, p. 2). The same issues not using Scrum in this project would have caused defects to surface later in the project causing more rework, costing money, and valuable resources. Inspections were also used to examine the coding and determine when developers could progress to work on other parts of the project (Sutherland, et al., 2007, p. 3). The inspections lasted only a few minutes, so they would not impede the project's progress. Story-based software development worked well where features were subdivided into stories that were composed of a smaller numbers of hours. The strategy was to break up task features into smaller parts in order to better manage the processes (p.3). Most important for these subdivided features would be early testing before any coding was done. This process was incremental and feature driven with inspections. Story-based development better defined when the individual tasks were complete,

compared to other methods. Additionally, early testing indicated the progress of individual tasks and overall project progress to all stakeholders (p.3).

Planning was a key element to how Systematic conducted business. Mary Poppendieck, a leading authority in Lean Software Development was consulted. Lean Software Development was used with the idea to "Build Integrity In" (Sutherland, et al., 2007, p. 2) and accelerate learning and ensure fast delivery. Additional planning went into the decision to mix CMMI with Scrum. According to Sutherland et al. (2007, p. 1), "Management of complexity requires process discipline while management of change requires adaptability. CMMI provides process discipline and Scrum enhances adaptability." A delicate balance between complexity, functionality and schedule was essential for this to be successful.

Systematic evaluated requirements specified by the customer and reevaluated what was actually needed. If the customer did not like limiting the scope, Systematic simply did not do the project. The XP2002 Standish Group Study in Sutherland (Sutherland, et al., 2007, p. 4) stipulates that 64% of features in fixed price contracts are rarely utilized by customers. Openness and ongoing communication are important to the scope, cost, and satisfaction of any software development project. The scope of the project should be focused on the specific needs of the customer. One case resulted in a 50% reduction of requirements and overall cost. The methodologies used at Systematic ensured satisfaction of the customer and a high quality of the product (Sutherland, et al., 2007, p. 3).

A Productivity Performance Baseline (PPB) was used as a metric for projects with Systematic using LOC divided by total hours for a project (Sutherland, et al., 2007, p. 3).

The larger projects (more than 4000 hours) showed a 201% increase in productivity while small projects (less than 4000 hours) exhibited significantly more productivity in comparison to the large projects (Sutherland, et al., 2007, p. 3). There were additional improvements overall. Also, there was significant improvements shown with the small projects compared to the large projects. However, members within the organization agreed that the improvement was attributable to Scrum. Additional explanation for this phenomenon is that the projects had already used processes and management techniques similar to Scrum. With Scrum, smaller projects outperformed the large projects when comparing with the PPB. Results also were favorable with the large projects. One large project with 10 team members reported a 38% reduction in defects. Another large project with 19 team members also used early testing. This team showed a 42% reduction in defects (p.4). Close customer ties were important. Weekly goals were set. Close teamwork was promoted. Collocation of test, domain specialist, and development personnel were used to help promote communication and team work. Documentation, along with entry and exit criteria, was required for the projects.

Summary of Systematic

This is an ideal representation of organizations studied in the body of knowledge. CMMI can be implemented for small organizations while considering schedule, budget, and customer satisfaction. The Systematic case study is an example of an organization already at CMMI Level 5 implementing Scrum to complement its ability to be disciplined and mature while also being flexible and nimble for organizations. They did this by using "Small [incremental] adjustments to existing processes" (Sutherland, et al., 2007, p. 4). These projects may be examples of ideal circumstances. However, the results show a

strong improvement while strong customer satisfaction is provided. A vast amount of experience was behind the mastery of how Systematic was successful (Sutherland, et al., 2007, pp. 1,4). The ability of the organization to use Lean Software Development, CMMI, and then Scrum enabled them to optimize their processes, policies, and practices. Systematic's recipe of using Lean Software Development and Scrum produced successful results. Processes were institutionalized through CMMI to benefit the organization (2007, p. 2) while Scrum enabled the organization to be more nimble and adaptable.

In order to manage requirements, innovation, and complexity, both Agile and CMMI methodologies can be used together (Sutherland, et al., 2007, p. 1). The CMMI methodology can be mixed with Agile methodologies to suit unique situations where CMMI otherwise would not fit (Sutherland, et al., 2007, p. 6). Benefits from different methodologies are combined with Systemantic's implementation to prove that CMMI can be utilized for small organizations.

The clear intent of the organization to institutionalize these policies is vital. Stakeholders and particularly team members were clear on the organization's direction and purpose. This focus of the teams resulted in the notable results. This methodology may be used on some projects or all projects depending on the organization. It is recommended that use of the methodology companywide will provide the added benefits and economies of scale. Sutherland et al. (Sutherland, et al., 2007, p. 4) suggests "face-to-face" meetings such as with the "all hands meetings". Management buy-in and involvement is a key to driving the institutionalization.

Case Study Summary

All of the answers to the thesis questions are presented with the thesis case studies. This paper describes organizations in broad terms where small organizations can be part of larger organizations or companies. It is difficult for small organizations to build a SPI solution with few or limited means. Some perceptions of opponents of CMMI are valid. However, this does not mean that small organizations are unable to adopt CMMI. Many CMMI principles founded and improved upon by SEI have helped lead small organizations to successful and impressive SPI results like NAVAIR and Systematic.

Case study answers

Regarding the Irish case study, its research question is the same as that presented in this thesis. The answer is difficult to perceive. However, the result of the Irish study provides a series of frameworks and categories that illustrate how SPI is adopted for small organizations. Specifically, the case study presents the cost of process network and process evolution network that forms the overall framework mapped out by Coleman and O'Connor (2008, pp. 777,779,780) elaborating the process and defining what is most difficult when implementing SPI strategies for small organizations.

Opponents of CMMI primarily note that organizational bureaucracy and documentation hinder any progress in small organizations. The results of the Irish study indicate that the outcome of these practices is what Coleman and O'Connor (2008, p. 779) refer to as minimum process. Developers are typically not motivated by additional bureaucracy and documentation and not all software organizations can sustain an overburdened bureaucracy and technical writing staff. What was found is that less documentation correlated only with organizations that had a high amount of trust with

their developers. Not all organizations fit this profile. An organization with a low amount of trust toward developers typically required more documentation. Small organizations also typically had less bureaucracy, a detail that was compensated by having more communication, also allowing for less documentation (p. 780). A balance with any individual organization must be found.

There is a common myth that CMMI is for large companies that have driving resources and the means that can support a full implementation of CMMI. Consequences of this myth have caused small organizations to not adopt CMMI. Options can be explored to customize or tailor CMMI to fit an organization. However, many company cultures are not open to change. Other SPI models may fit better for a particular company. There is no one size fits all solution, but a combined CMMI methodology with an Agile methodology can fit the needs of most any organization. Implementing CMMI is not easy, but exploring what it can offer is important for finding the best opportunity for an organization. The cost of this improvement can be justified by its savings alone if applied properly. Organizations can benefit from embracing this technology and move on to continuous software improvement.

Incremental change improves an implementation because it provides stability and a transparent awareness to stakeholders. High transparency adds visibility to a project for all stakeholders to know where the project stands. In Sutherland et al. (Sutherland, et al., 2007, p. 3) this demonstrates a steady implementation that can be improved upon. Cyclical improvement building on previous cycle iterations leads to continuous improvement. This scaled down approach helps in envisioning the possibilities for the

organization as a whole. Internal training and knowhow of CMMI has been a key factor to implementation success.

Expertise in CMMI and other SPI methodologies is used to leverage progress and is a major contributor to the success and momentum of the projects in the case studies presented. Each one of the case study examples centralizes on experience and training with SPI solutions. Hiring experts was one of the primary process categories identified by Coleman and O'Connor (Coleman & O'Connor, 2008, p. 779). Having an experienced team also is a factor that contributes to success implementing SPI initiatives. A member of the SLT was a PSP instructor and helped NAVAIR establish its use of PSP, TSP, and CMMI. Additionally, Watts Humphrey conducted a briefing with the NAVAIR SLT. This led to many other organization members to become TSP coaches and PSP instructors. In Sutherland, et al. (Sutherland, et al., 2007, p. 2), a management seminar session was held with Mary Poppendieck who is a leading author with Lean Software Development. Sutherland himself is a well known authority on Scrum. Overall, there was a significant presence and influence of highly trained and specialized individuals for the projects represented in the case studies. Instituting SPI methodologies requires a level of proficiency to be successful and consultants would be recommended if there was a lack of internal expertise.

Motivation, confidence, and desire of developers to achieve success are what drove the individuals of the organization to victory. In the Irish case study, employee buy-in had an adverse effect on the organizations and caused what they called minimum process. Buy-in from employees was a primary factor that contributed to the adoption and realization of the methodology.

Buy-in from management is paramount to maintaining direction, motivation, and support. Also management makes the project possible. Common success factors involve the empowerment of management on driving the organization and using CMMI. Organizations that do not have buy-in from management are doomed to fail. While this may seem obvious, the data from the study does indicate that management's adoption and oversight of SPI solutions does highly influence the outcome. With any change, there is resistance identifying, and addressing this problem must be a primary concern. In the NAVAIR case study, management was a champion to change and influenced every step of the process. It was recognized that the employees also must be involved and understand why the changes were taking place.

PSP, TSP, and CMMI synergy was very effective in the NAVAIR study. The commonly approved KPA with the three methodologies truly did accelerate the projects with NAVAIR. Companies developing this ground up approach in hope of growing into a larger organization benefit from the process and its evolution. The more a company puts into the process, the more it gets out of it. Starting slowly and building the process along with the organization is a natural progression.

The intent of this paper is to demonstrate that small organizations do not need to "reinvent the wheel" with SPI. The technology is available with CMMI and it is maturing to better maintain process improvement into the future and adjust with the ever changing industry. With a mixed methodology, adaptable, agile, and creative process can be realized with CMMI. The creativity of Agile combined with the discipline from both CMMI and Agile offer a better SPI methodology that can fit within large and small organizations. The discipline is there to actually assist the creative process. The

combination of CMMI and Agile technologies enables the benefits of both flexibility and performance. It is true that small organizations can adopt CMMI based on what is presented in this thesis. If CMMI could be implemented properly for more software organizations large and small, fewer projects would fail. Software projects will more likely be completed on schedule and within budget while also being more innovative in fulfilling customer needs.

Chapter Five – Lessons Learned and Next Evolution of the project

CMMI can help software organizations develop and improve. However, not all organizations benefit from CMMI or SPI. Some developers follow practices for the sake of following practices, and not for individual improvement and the benefit of the organization. The greatest stumbling block is the ability to affect change (Wall, et al., 2007, p. 19). Enabling developers individually is one hurdle. Management buy-in institutionalizes change and compels the organization to follow in the same direction.

CMMI for small organizations

In most situations, only large organizations are known to adopt CMMI (Anderson, 2005, p. 12). The author focused the research to produce conclusions regarding whether the thesis holds true. The reason for this research question focus is because of the myth about CMMI. The perception that the methodology is an impediment to quickly develop software that is unnecessary and cumbersome (Glazer, 2001, p. 1). Even with this perception, CMMI is based on software engineering experience that goes back decades (2005, p. 12). In order to circumvent this stigma, there had to be examples of a traditional or non-traditional CMMI implementation successfully implemented for small organizations. Is it possible to find a scaled down solution that satisfies requirements of CMMI while working for the advantage of a small organization? This solution would be both for the benefit of the small organizations that would have this powerful technology available and also in the interest of enlightening people that CMMI can be implemented based on the need that it would fill.

Alternative methodologies to CMMI opening opportunities

Alternative methodologies such as EssWork (Jacobson, et al., 2007, p. 65), VBSE (Boehm, 2003, p. 11), and Agile methodologies (Anderson, 2005, p. 13; Glazer, et al.,

2008, p. 24) all can be used with CMMI. While it is important to acknowledge these alternatives, the concentration of this thesis was on a traditional CMMI solution that can be implemented in small organizations. As a result of finding the use of mixed methodologies with CMMI, the author looked further at the possibilities of combining other methodologies used with small organizations closely. Research conducted after considering alternatives used along with CMMI led to opportunity with other findings. Many ideas were inspired by reading and researching new ideas in the field of SPI and software development. These additional ideas contributed to the outcome of the study.

Solid SPI models set out processes and practices that are intended to be proven methods for continuous improvement. SEI considers CMMI a model and not a standard (Glazer, et al., 2008, p. 7). CMMI has had an increasing Adoption Rates in the United States and overseas. Today organizations still do not adopt these models of SPI due to the myth that CMMI is designed and intended for large organizations. Without sound SPI, integration of disparate software development systems generally are not mature and lack automation. CMMI has taken hold in industry, and there is opportunity for growth with small organizations. It is difficult to state that there is a "one size fits all" solution (Coleman & O'Connor, 2008, p. 782). However, with a combined solution, an organization can utilize a SPI solution with CMMI that produces results with organizations of any size. With the recognition that CMMI can be utilized in small organizations the opportunity for an improved awareness of the solutions can advance industry and SPI technologies overall.

Lessons learned

The case studies presented in this thesis demonstrate a need and ability of small organizations to adopt CMMI. The next evolution of this thesis project would be to analyze the appraisal process and determine how a CMMI implementation would work: how best to implement CMMI solutions with less of a budget and resources. SEI in Anderson (2005, p. 7)

suggests that payoffs for CMMI are delivered at the higher levels. Therefore, there is a market for additional prepackaged enhancements and improvements. The current MSF is one potential solution. With this in mind, the MSF intends to extend the ability to achieve CMMI to Level 5.

Recent Trends

Today, the project reveals that there are many players in the software industry that maintain good SPI standards. People tend to differ in opinion when the subject is as important as software development. Early in 2008 Microsoft released its latest Visual Studio Team System (VSTS) including the MSF with the idea of collaboration and integration. The MSF is an interpretation of the Agile methodology that fulfills the requirements of CMMI Level 3 (Anderson, 2005, p.2). With this solution, there are more opportunities for small organizations to adopt the SPI models of CMMI.

Corbis and MSF

The Agile community can benefit from the lessons and predictability of CMMI. Likewise, organizations that use CMMI can benefit from useful techniques and practices of Agile methodologies. David Anderson, with his company Corbis, has "been able to meet CMMI rigor for predictability while simultaneously using an Agile approach that adapts to the unpredictability of the work and the market" (Glazer, et al., 2008, p. 10). This example of applying Agile concepts to CMMI in the market place is still a test bed and prototype of similar projects. Like the case studies presented in this paper there is a niche for CMMI for small business. The capabilities and use of CMMI are expanded with its combined use with Agile methodologies. There is current support of CMMI and Agile combined methodologies by SEI (Glazer, et al., 2008). Interestingly, MSF is referenced many times. However, there is little academic research on MSF, as mentioned early in this paper.

Next Evolution of the Project

Future research that is applicable to SPI is automation to eliminate the human variable and provide as close to faultless predictability where possible. However, the topics of automation and standards would be better discussed with a separate research project. EssWork is geared to balance process and enable practice to promote collaboration. This product offering integrates with Microsoft Visual Studio, Team Foundation Server, and uses the MSF (2007, p. 64). A fully integrated solution, prepackaged for a general audience, is needed. Additional marketing and detail is needed on the part of the originators of CMM/CMMI (Coleman & O'Connor, 2008, pp. 773-774). Different methodologies, practices, and rules have common ideals and origins (Anderson, 2005, p. 13), so a common ground can be found. With these commonalities and relationships, it is possible for combined methodologies to complement one another. Glazer (2001, p. 2) describes XP as a software development methodology and CMMI as a management methodology that complement one another. For the purpose of realizing a SPI solution for small business, Agile, PSP, and TSP can be explored more in depth.

The next insightful question would be: How can small organizations find a cost effective SPI solution based on CMMI? Cost is not the only problem. Commitment and follow through by users for small software organizations is important. Buy-in by management in these situations is also difficult and costly. Without buy-in, the project is not possible. There have to be solutions available that small organizations can more readily use.

Conclusion

CMMI is a solution to at least part of the problem of projects not completing on schedule and within budget. Case studies presented in this paper demonstrate the need and ability of using CMMI. The practical application of CMMI implementation in small organizational settings is possible. When considering the history of CMMI verses Agile, Hillel Glazer's conclusion that XP could be appraised at Level 2 by following its Agile rules and practices was astounding. Today the MSF takes the concept much further, which is idyllic. Glazer saw the value in this approach. Stakeholders can now have additional confidence in a software organization that is certified under CMMI guidelines, which shows a balanced organization that is focused on SPI. This study reveals what is available in the software development world today. The analysis and results of this study show that small software development organizations can adopt a CMMI implementation using several different methods. Divisive practices and non-standard trends in the industry have separated SPI solutions in the past and have diminished a synergy that would otherwise benefit the software development world. There are software experts from Agile and CMMI communities trying to bridge the gap to take advantage of benefits from both software methodologies. Creativity can coexist with discipline in SPI. Whether we are Agile or set in our ways, disciplined or inspired, every individual has a way of producing results. The organization needs to fit for the individual just as the individual needs to fit the organization or methodology. Otherwise, results are unpredictable. As with the menu versus the recipe analogy in the introduction of this paper, the ad hoc software development environment can be likened to a game of people playing with different rules, different processes, and different goals. With a standard process model,

software development organizations have the ability to deliver predictable, reusable, and sustainable software that can be continually improved upon. Once sustainability is achieved, it can be continued and developed further to produce a better product. Users of the software produced will benefit from the improvement it affords.

References

Abernethy, K., & Piegari, G. (2007). Teaching Project Management: An Experiential Approach. Retrieved September 1, 2009, from website:http://delivery.acm.org.dml.regis.edu/10.1145/1190000/1181888/p198-abernethy.pdf?key1=1181888&key2=5118491521&coll=Portal&dl=GUIDE&CFID=49983700&CFTOKEN=14450108

Anderson, D. J. (2005). Stretching Agile to fit CMMI Level 3. Retrieved December 31, 2007, from website:http://www.serena.com/docs/agile/papers/agile-2005-paper.pdf

Boehm, B. (2003). Value-based software engineering. Retrieved December 31, 2007, from website:http://delivery.acm.org.dml.regis.edu/10.1145/640000/638775/p3-boehm1.pdf?key1=638775&key2=3090109421&coll=ACM&dl=ACM&CFID=27457292&CFTOKEN=86701357

Capability Maturity Model Integration (CMMI). (2009). Retrieved September 7, 2009, from website:http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

Cesare, S. d., Patel, C., Iacovelli, N., Merico, A., & Lycett, M. (2008). Tailoring Software Development Methodologies in Practice: A Case Study. Retrieved June 6, 2009, from website:http://cit.zesoi.fer.hr/browseIssue.php?issue=35

Coleman-Dangle, K., Larsen, P., Shaw, M., & Zelkowitz, M. V. (2005). Software Process Improvement in Small Organizations: A Case Study. Retrieved December 31, 2007, from website:http://csdl.computer.org.dml.regis.edu/dl/mags/so/2005/06/s6068.pdf

Coleman, G., & O'Connor, R. (2008). Investigating software process in practice: A grounded theory perspective. Retrieved March 9, 2008, from website:http://www.sciencedirect.com.dml.regis.edu/science?_ob=MImg&_imagekey=B6V0N-4PBDPV7-1-7&_cdi=5651&_user=1922016&_orig=search&_coverDate=05%2F31%2F2008&_sk=999189994&view=c&wchp=dGLbVzz-zSkWb&md5=6446a66667c21978d9f0c63bd6c250c9&ie=/sdarticle.pdf

Elshafey, L. A., & Galal-Edeen, G. H. (2008). Combining CMMI and Agile Methods. Retrieved May 25, 2009, from website:http://www.fci-cu.edu.eg/infos2008_old/infos/SE_04_P027-039.pdf

Gack, G. A., & Robinson, K. (2003). Integrating Improvement Initiatives: Connecting Six Sigma for Software, CMMI, Personal Software Process (PSP) , and Team Software Process (TSP). Retrieved May 20, 2004, from website:http://www.asq.org/pub/sqp/past/vol5_issue4/sqpv5i4gack.pdf

Galin, D., & Avrahami, M. (2006). Are CMM Program Investments Beneficial? Analyzing Past Studies. Retrieved May 20, 2008, from website:http://csdl.computer.org.dml.regis.edu/dl/mags/so/2006/06/s6081.pdf

Ginsberg, M. P., & Quinn, L. H. (1995). Process Tailoring and the the Software Capability Maturity Model. Retrieved February 14, 2009, from website:http://www.sei.cmu.edu/reports/94tr024.pdf

Glazer, H. (2001). Dispelling the Process Myth: Having a Process Does Not Mean Sacrificing Agility or Creativity. Retrieved March 1, 2008, from website:http://www.entinex.com/CrossTalkNov2001.pdf

Glazer, H., Dalton, J., Anderson, D., Konrad, M., & Shrum, S. (2008). CMMI or Agile: Why Not Embrace Both! Retrieved November 28, 2008, from website:http://www.sei.cmu.edu/pub/documents/08.reports/08tn003.pdf

Guckenheimer, S., & Perez, J. J. (2006). Software Engineering with Microsoft Visual Studio Team System, Adobe Reader. Retrieved May 8, 2006, from website:https://www.informit.com/content/images/0321278720/samplechapter/Guckenheimer_ch01.pdf

Heinz, L. (2003a). CMMI Adoption Trends. Retrieved May 18, 2004, from website:http://www.sei.cmu.edu/news-at-sei/features/2003/4q03/feature-1-4q03.htm

Heinz, L. (2003b). Taking the Road Less Traveled: The CMMI Contunuous Approach. Retrieved May 20, 2004, from website:http://www.sei.cmu.edu/news-at-sei/features/2003/1q03/pdf/feature-4-1q02.pdf

Humphrey, W. S. (1995). *A Discipline for Software Engineering*. Boston, MA: Addison Wesley Longman Publishing Co., Inc.

Humphrey, W. S. (2005). Why Big Software Projects Fail: The 12 Key Questions. Retrieved December 28, 2008, from website:http://la-acm.org/Archives/laacm0512-Why-Big-SW-Projects-Fail0503Humphrey.pdf

Jacobson, I., Ng, P. W., & Spence, I. (2007). Enough of Processes - Lets do Practices. Retrieved May 7, 2008, from website:http://www.jot.fm/issues/issue_2007_07/column5.pdf

Jones, L. G., & Soule, A. L. (2002). Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice. Retrieved July 11, 2009, from website:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.7278&rep=rep1&type=pdf

Jørgensen, M., & Moløkken, K. (2006). How Large Are Software Cost Overruns? A Review of the 1994 CHAOS Report. Retrieved September 1, 2009, from website:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.2578&rep=rep1&type=pdf

Kannenberg, A., & Saiedian, H. (2009). Why Software Requirements Traceability Remains a Challenge. Retrieved September 1, 2009, from website:http://www.stsc.hill.af.mil/crosstalk/2009/07/0907KannenbergSaiedian.html

'Know Thyself' is the First Step to Successful Knowledge Management. (2005). Retrieved August 8, 2009, from website:http://knowledge.wpcarey.asu.edu/article.cfm?articleid=1122

Laporte, C. Y., Alexandre, S., & Renault, A. (2008). Developing International Standards for Very Small Enterprises. Retrieved May 21, 2008, from website:http://profs.logti.etsmtl.ca/claporte/English/VSE/Publications/Developing%20International%20Standards_IEEE_Comp_March_08.pdf

Leedy, P. D., & Ormrod, J. E. (2005). *Practical Research Planning and Design* (8th ed.). Upper Saddle River: Kevin M. Davis.

Liu, J. Y.-C., Chen, V. J., Chan, C.-L., & Lie, T. (2008). The impact of software process standardization on software flexibility and project management performance: Control theory perspective. Retrieved March 9, 2008, from website:http://www.sciencedirect.com.dml.regis.edu/science?_ob=MImg&_imagekey=B6V0B-4RP0MJ7-3-3&_cdi=5642&_user=1922016&_orig=search&_coverDate=01%2F26%2F2008&_sk=999999999&view=c&wchp=dGLbVzb-zSkWz&md5=4183f853689046169419b859be798269&ie=/sdarticle.pdf

McHale, J. (2003). The Case for Using TSP with CMM/CMMI. Retrieved June 20, 2009, from website:http://www.sei.cmu.edu/library/assets/tsp.pdf

Neville, K., Hoffman, R. R., Linde, C., Elm, W. C., & Fowlkes, J. (2008). The Procurement Woes Revisited. Retrieved December 28, 2008, from

website:http://www.computer.org.dml.regis.edu/portal/web/csdl/abs/mags/ex/2008/01/mex2008010072abs.htm

Niazi, M., Wilson, D., & Zowghi, D. (2005). A Maturity Model for the Implementation of Software Process Improvement: an Emperical Study: An Emperical Study. Retrieved December 31, 2007, from website:http://www.sciencedirect.com.dml.regis.edu/science?_ob=MImg&_imagekey=B6V0N-4BBMWD5-5-1&_cdi=5651&_user=1922016&_orig=search&_coverDate=01%2F15%2F2005&_sk=999259997&view=c&wchp=dGLzVlz-zSkWA&md5=a960c881fcf987b2fca03f75bad00068&ie=/sdarticle.pdf

Paulk, M. C. (2001). Extreme Programming from a CMM Perspective. Retrieved December 31, 2007, from website:ftp://ftp.sei.cmu.edu/pub/documents/articles/pdf/xp-from-a-cmm-perspective.pdf

Phongpaibul, M., & Boehm, B. (2006). An Empirical Comparison Between Pair Development and Software Inspection in Thailand. Retrieved September 7, 2009, from website:http://delivery.acm.org.dml.regis.edu/10.1145/1160000/1159749/p85-phongpaibul.pdf?key1=1159749&key2=4653732521&coll=Portal&dl=GUIDE&CFID=51826041&CFTOKEN=99987352

Request for Proposal (RFP). (2009). Retrieved September 1, 2009, from website:http://www.google.com/search?hl=en&lr=&defl=en&q=define:Request+for+proposal&ei=MQUTS6HsBNaOtge0r8ysCQ&sa=X&oi=glossary_definition&ct=title&ved=0CAkQkAE

Ryan, S., & O'Connor, R. V. (2008). Development of a team measure for tacit knowledge in software development teams. Retrieved December 28, 2008, from website:http://www.sciencedirect.com.dml.regis.edu/science?_ob=MImg&_imagekey=B6V0N-4SPYKC9-1-7&_cdi=5651&_user=1922016&_orig=search&_coverDate=06%2F10%2F2008&_sk=999999999&view=c&wchp=dGLbVzb-zSkWb&md5=d3e74f4a36be5af134ca22411040d893&ie=/sdarticle.pdf

SEI. (2005a). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2004 Year End Update. Retrieved December 28, 2008, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2005marCMMI.pdf

SEI. (2005b). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2005 Mid-Year Update. Retrieved January 14, 2009, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2005sepCMMI.pdf

SEI. (2006a). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2005 Year End Update. Retrieved December 28, 2008, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2006marCMMI.pdf

SEI. (2006b). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2006 Mid-Year Update. Retrieved December 28, 2008, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2006sepCMMI.pdf

SEI. (2006c). CMMI for Development, Version 1.2. Retrieved November 28, 2008, from website:http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf

SEI. (2007a). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2006 Year End Update. Retrieved December 28, 2008, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2007marCMMI.pdf

SEI. (2007b). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2007 Mid-Year Update. Retrieved December 2008, 2008, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2007sepCMMI.pdf

SEI. (2008a). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2007 Year End Update. Retrieved December 28, 2008, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2008MarCMMI.pdf

SEI. (2008b). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2008 Mid-Year Update. Retrieved December 28, 2008, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2008SepCMMI.pdf

SEI. (2009a). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2008 Year End Update. Retrieved April 28, 2009, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2009MarCMMI.pdf

SEI. (2009b). Process Maturity Profile CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2009 Mid-Year Update. Retrieved November 27, 2009, from website:http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2009SepCMMI.pdf

Staples, M., Niazi, M., Jeffery, R., Abrahams, A., & Murphy, P. B. (2007). An Exploratory Study of Why Organizations do not Adopt CMMI. Retrieved February 9, 2008, from website:http://www.sciencedirect.com.dml.regis.edu/science?_ob=MImg&_imagekey=B6V0N-4M64516-1-C&_cdi=5651&_user=1922016&_orig=search&_coverDate=06%2F30%2F2007&_sk=999199993&view=c&wchp=dGLzVzz-zSkzV&md5=43af496904754f51f6b4510d30445ca9&ie=/sdarticle.pdf

Sutherland, J., Jakobsen, C. R., & Johnson, K. (2007). Scrum and CMMI Level 5: The Magic Potion for Code Warriors. Retrieved February 8, 2009, from website:http://jeffsutherland.com/scrum/Sutherland-ScrumCMMI6pages.pdf

Verner, J. M., Evanco, W. M., & Cerpa, N. (2007). State of the practice: An exploratory analysis of schedule estimation and software project success prediction. Retrieved December 28, 2008, from website:http://www.sciencedirect.com.dml.regis.edu/science?_ob=MImg&_imagekey=B6V0B-4KDBM1K-1-3&_cdi=5642&_user=1922016&_orig=search&_coverDate=02%2F28%2F2007&_sk=999509997&view=c&wchp=dGLzVtb-zSkzS&md5=d74ddfb6f6e4bdc551bfd6d79eebc6c3&ie=/sdarticle.pdf

Wall, D. S., McHale, J., & Pomeroy-Huff, M. (2007). Case Study: Accelerating Process Improvement by Integrating the TSP and CMMI. Retrieved January 14, 2009, from website:ftp://ftp.sei.cmu.edu/pub/documents/07.reports/07tr013.pdf

Yin, R. K. (2003). *Case Study Research Design and Methods* (3rd ed.). Thousand Oaks: Sage Publications, Inc.

Appendix A

Abbreviation	Name	Area	Maturity Level
REQM	Requirements Management	Engineering	2
PMC	Project Monitoring and Control	Project Management	2
PP	Project Planning	Project Management	2
SAM	Supplier Agreement Management	Project Management	2
CM	Configuration Management	Support	2
MA	Measurement and Analysis	Support	2
PPQA	Process and Product Quality Assurance	Support	2
PI	Product Integration	Engineering	3
RD	Requirements Development	Engineering	3
TS	Technical Solution	Engineering	3
VAL	Validation	Engineering	3
VER	Verification	Engineering	3
OPD	Organizational Process Definition	Process Management	3
OPF	Organizational Process Focus	Process Management	3
OT	Organizational Training	Process Management	3
IPM	Integrated Project Management	Project Management	3
RSKM	Risk Management	Project Management	3
DAR	Decision Analysis and Resolution	Support	3
OPP	Organizational Process Performance	Process Management	4
QPM	Quantitative Project Management	Project Management	4
OID	Organizational Innovation and Deployment	Process Management	5
CAR	Causal Analysis and Resolution	Support	5

CMMI Key Process Areas (KPA) ("Capability Maturity Model Integration (CMMI)," 2009, p. 3).

Annotated Bibliography

Abernethy, K. and Piegari, G. (2007). "Teaching project management: An experiential approach." *Journal of the Consortium for Computing Sciences in Colleges*. 22(3), 198-205. Retrieved September 1, 2009 from http://delivery.acm.org.dml.regis.edu/10.1145/1190000/1181888/p198-abernethy.pdf?key1=1181888&key2=5118491521&coll=Portal&dl=GUIDE&CFID=49983700&CFTOKEN=14450108

The Standish report offers data about the progress of information technology. Pertinent statistics are presented in this project management reference. Technology is partly the cause of recent improvements in the field of information technology, although project management is listed as a primary contributor. The data from this article is significant to show that there is an improvement in the field of information technology.

Anderson, D. J. (2005). "Stretching Agile to fit CMMI level 3: .The story of creating MSF for CMMI® process improvement at Microsoft Corporation." Paper presented at Agile Conference, Denver, CO, July 2005. Retrieved December 31, 2007 from http://www.serena.com/docs/agile/papers/agile-2005-paper.pdf.

The teachings of W. Edwards Deming help implement a new solution to achieve software development standards through an adaptive, iterative, and affective way of working. It is adopted to fill a niche in the market. Microsoft Solutions Framework (MSF) offers a solution for small to medium size businesses with a path to achieving the requirements for the Capability Maturity Model Integration (CMMI) maturity level 3. The progression of Team Foundation Server (TFS) to incorporate the two schools of thought of the CMMI and Agile software development models is described in this publication. This balance between Agile and CMMI process and application is from a culmination of years of work. Agile methodologies concentrate on productive, highly iterative, and responsive techniques while producing highly valuable software products quickly. CMMI methodologies are grounded, stable, and disciplined. Both the Agile methods and CMMI have basis in Deming's work. This publication reviews some of Deming's concepts such as completion velocity, special cause variation, common cause variation, and tampering. Utilizing a loose project plan and not one with command and control is emphasized, while providing iterative work process.

Boehm, B. (2003). "Value-based software engineering." *Software Engineering Notes*, 28(2), 1-12. Retrieved December 31, 2007, from http://delivery.acm.org.dml.regis.edu/10.1145/640000/638775/p3- boehm1.pdf?key1=638775&key2=3090109421&coll=ACM&dl=ACM&CFID=27457292&CFTOKEN=86701357.

Value-based software engineering (VBSE) is proposed as a replacement for the old way of Value-neutral software engineering in this scholarly publication. A paradigm shift is promoted for ". . . developing an overall framework in which they compatibly reinforce each other." This shift considers many methodologies, including RUP and CMMI. Boehm includes a synthesis of data that breaks down the economic and mathematic factors for the reader using the principles outlined early in the paper. The

statistical analysis offers valuable tools for evaluating cost and schedule of projects. Software engineering is presented as a hard science and indicates that the old value-neutral approach does not work. With VBSE, project schedule and cost are tracked to evaluate where the project is at any one point in time. Software engineers are now more responsible for the code produced in being able to produce results that are measurable and should feel empowered to produce results because of the greater responsibility and trust.

Boehm, B. (2008). "Making a difference in the software century." *Computer*. 41(30, 32-28. Retrieved May 21, 2008, from http://www.computer.org/portal/site/computer/menuitem.5d61c1d591162e4b0ef1bd108b cd45f3/index.jsp?&pName=computer_level1_article&TheCat=1005&path=computer/homepage/0308&file=cove r.xml&xsl=article.xsl&

The decades to come will define the 21st century as the Software Century. Software engineers are in high demand and the challenge is there for them to accept and succeed with this challenge. There are several obstacles to overcome in the industry, though, to be successful. Many acronyms are coined, such as: THWADI (That's How We've Always Done It); BITAR (Buy Information To Avoid Risk); IKIWISI (I'll Know It When I See It); DAVAS (Dependability As Value Assured to Stakeholders); OSUFA (One Size Uniformly fits All); SISOS (Software Intensive Systems of Systems); TANIA (There Are No Islands Anymore) by this well known author. These acronyms are defined and used to communicate to software engineer's current state of the industry and how change, rapid change, uncertainty and emergence, dependability, interdependence, and diversity are challenges. This is a very positive publication which sounds like a pep encouraging one to affect a change. Boehm makes a convincing argument that technology is leading to even more rapid change, and that the software industry needs to update old methods. Evidence of CMM not working in Thailand is presented as an argument not to use OSUFA practices.

Borjesson, A. and Mathiassen, L. (2004). "Successful process implementation." *Software*, 21(4), 36-44. doi: 10.1109/MS.2004.27.

At the telecommunications company Ericsson, 18 different process initiatives were studies over the course of five years. Defined in this study is the concept of implementation success. Borjesson points out that 1.) SPI is not possible without implementation success; 2.) assessing implementation success is easy; 3.) the affects of SPI initiatives can been seen only when implementation succeeds; 4.) implementation success is integral to successful SPI. The use of the IDEAL (Initiating, Diagnosing, Establishing, Acting, Learning) model is used throughout this study. This model is from the SEI and is a highly iterative cycle for SPI. Iterations exposed failures, and in turn, led to the concentration of learning that added to the development of the process and product overall. The results of the study showed that the number of iterations improved the implementation success of projects. Overall, this was the case but there were some minor exceptions. The progression from old practices improved performance and led to implementation success. The authors of this publication recommend using practice pull and process push projects. When SPI initiatives were dedicated to individual projects,

resistance to change was overcome and learning was achieved, resulting in was implementation success.

Capability maturity model integration (CMMI). In *Wikipedia*. Retrieved September 7, 2009, from http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration.
This is a reference on the web of the CMMI definition. Key process areas correlating to what maturity level is mapped out for the different constellations.

Cesare, S. d., Patel, C., et al. (2008). "Tailoring software development methodologies in practice: A case study." *Journal of Computing and Information Technology* - CIT 16, 2008, 3, 157–168. doi:10.2498/cit.1000898
It has been assumed that software development principles and processes were founded on methodologies that are applicable to all sizes of organizations. The ability to tailor CMMI is fundamental in its ability to affect all different types and sizes of organizations. This is a well thought out pointed article on method tailoring, not just with CMMI, but with other SPI methodologies. Tailoring allows these practices and processes to work with different scenarios.

Christie, A. M. (1995). *Software process automation: The technology and ITS automation*. Pittsburgh, Carnegie Mellon University.
Initial research done in order to focus on computer automation and the beginnings of CMM was necessary for a better understanding of process automation. Published around the same time of many CMM references, it lent insight to what was important in SPI. This publication is focused on the early years of CMM and lends insight to what it prescribes. Detailed data on CMM and process automation tools are found to be very valuable as core research.

CMMI adoption trends. (2003, Dec. 1, 2003). *News at SEI*. Retrieved May 18, 2004, from http://www.sei.cmu.edu/news-at-sei/features/2003/4q03/feature-1-4q03.htm.
This article describes trends of CMMI as of the publication. Many statistics are provided to suggest what direction CMMI was going and an upward trend of current adoption. The collection of information is directed towards possible adopters of the methodology.

Coleman, G., & O'Connor, R. (2007). Using grounded theory to understand software process improvement: A study of Irish software product companies. *Information & Software Technology*, *49*(6), 654-667. doi:10.1016/j.infsof.2007.02.011.
Grounded theory is applied in development settings. Stages of grounding theory, including coding and memoing are described in detail through the process of conducting research. Categories are determined and used to logically analyze the research data. This methodology matches the technique needed for this study in that previous experience can help focus the research. Information viewed as the "cultural insider" fits with the author's personal experience and adds value to the topic. Research motivation is briefly described and can be on the focus of concentration. The theory can be revised based on the collected data. Irish software companies are used in the study of 21 companies. The results of the study in part suggest that XP was the most widely used and accepted

process model. The study found that both CMMI and ISO 9000 were perceived as requiring resources and money with less proportional benefit.

Coleman, G., & O'Connor, R. (2008). Investigating software process in practice: A grounded theory perspective. *Journal of Systems & Software*, *81*(5), 772-784. doi:10.1016/j.jss.2007.07.027.

Detailed case study of Irish software development organizations are presented and analyzed in this publication. This study focuses on small organizations, but mentions little about CMMI. XP is the methodology found to be used most with the organizations in Ireland. CMMI is discussed favorably. However, there is no representation of CMMI in the study. The importance of this paper is the questions that are asked about why SPI is difficult for small organizations. Several analyses and a process categorization based on grounded theory is invaluable to the insight of why small organizations use SPI.

Coleman-Dangle, K.,Larsen, P.,Shaw, M., & Zelchowitz (2008). "Software Process improvement in small organizations: A case study." *Software*, 22(6), 68-75. doi: 10.1109/MS.2005.162

The Capability Maturity Model contains 18 key process areas (KPA) that promote SPI. A motivation for an organization to implement the CMM is for requirements of contracts. This study of a small organization implementing the CMM is an ideal example of the material that is relevant and what is needed for SPI standards for smaller organizations. The benefits of having more development resources available for SPI in a larger organization are clearly pointed out. Some weaknesses of the CMM are highlighted, including how the levels are divided up by the individual KPA that may need to be considered at lower levels. This is later improved for metrics and analysis in CMMI level 2. This case study on a growing small business called DataStream Content Solutions (DSCS) is a prime example of a smaller organization attempting to benefit from methodologies like CMM. With DSCS a grant through the University of Maryland and the Fraunhofer Center for Experimental Software Engineering allowed for consultants to work with DSDS to start implementing CMM. The goal was to be at level 3 at the end of the year. However, they found that it takes time to achieve such a goal.

Elshafey, L. A. and Galal-Edeen, G. H.. (2008). "Combining CMMI and Agilemethods." Cairo: Cairo University. Retrieved May 25, 2009, from http://www.fci-cu.edu.eg/infos2008_old/infos/SE_04_P027-039.pdf.

Overcoming the stigma of combining CMMI and Agile methodologies is achieved in this paper. This different perspective uses data that compares the CMMI and Agile process areas. This approach to better SPI is new and innovative. This supports the thesis in that it proposes a combination of methodologies to fulfill the many diverse needs of software development. There is a synthesis of data backed up with other sources. This paper points out the pros and cons of both software development methodologies.

Gack, G. A. and Robinson, K. (2003). "Integrating improvement initiatives: connecting Six Sigma for software, CMMI, Personal Software Process (PSP), and Team Software Process (TSP)." *Software Quality Professional*, 5(4), 1-13. doi: 10.1109/MS.2006.

As advocate for using PSP and TSP, this paper corroborates the evidence that PSP and TSP can enhance CMMI implementations. Many process improvement methodologies are reviewed in this article. Six Sigma, CMMI, PSP, and TSP. PSP and TSP as a solution to fulfill CMMI KPAs are discussed. A conclusion the authors derive is that the primary goal of process improvement is to enable people to become more efficient and effective.

Galin, D. and Avrahami, M.. (2006). "Are CMM program investments beneficial? Analyzing past studies." *Software*, 23(12), 81-87. Retrieved May 20, 2008, fromhttp://csdl.computer.org.dml.regis.edu/dl/mags/so/2006/06/s6081.pdf.

This study of over 400 projects reported in 19 papers, analyses whether the CMM program investments are beneficial. Benefits listed include: 1.) milestone completion on time improvement; 2.) fewer defects; 3.) fewer correction cycles and regression tests; 4.) reduced error density. Economic gains include: 1.) reduced cost of testing and maintenance; 2.) better design review methods, improved development methods overall and reduced development cost due to productivity; 3.) reduced compensation paid to customers for slipped release dates or faulty software. This study goes to great lengths to utilize a wide sampling of the target population. One key element is identifying the different projects in the study over time with CMM level transition (CMMLT) to show at what level in CMM the project progressed. Other variations in projects are performance metrics, development tasks, project size, use of different development environments, and use of different coding practices. This study concludes that the CMM does lead to improvement in software maintenance and development.

Ginsberg, M. P. and Quinn, L. H. (1995). "Process tailoring and the Software Capability Maturity Model." (Report No. CMU/SEI-94-TR-24). Pittsburgh, PA.: Carnegie Mellon University. Retrieved February 14, 2009, fromhttp://www.sei.cmu.edu/reports/94tr024.pdf.

Process tailoring for the Capability Maturity Model (CMM) is detailed in this paper that promotes the model to all types of organizations. The myth that CMM and CMMI is a model that is a one size all is dispelled by this publication which suggests customization or tailoring of each implementation to individual organizations. The promotion of the tailoring practices benefits the methodologies in allowing its promotion to more than just large organization and government contracts.

Glazer, H. (2001). "Dispelling the process myth: Having a process does not mean sacrificing agility or creativity." *Crosstalk: The Journal of Defense Software Engineering*, 14(12), 27-30. Retrieved March 1, 2008, from http://www.entinex.com/CrossTalkNov2001.pdf.

There are two camps expressing opposing views in the software industry. XP programming is compared with CMM methodologies. Glazer simplifies the issue by dispelling some myths on both sides of the issue and creatively presenting some solutions. Early on, it is pointed out that CMM originated from large contracts and the Department of Defense (DOD). Also, XP programmers in general may be considered undisciplined from a CMM perspective. The author describes XP as a software development methodology and CMM as a management methodology that complement

one another. Size of projects are important in this analysis because large projects typically would not fit under an Agile or XP type methodology, where small projects would be more apt to adopt CMM baring any resource or financial limitations. Even though an organization is small, CMM is meant to be tailored. Glazer defines a small project as no more than 20 people.

Glazer, H., Dalton, J., et al. (2008). "CMMI or Agile: Why not embrace both!" (Report No. CMU/SEI-2008-TN-003). Hanscom AFB, MA.: Carnegie Mellon University. Retrieved November 28, 2008, from http://www.sei.cmu.edu/pub/documents/08.reports/08tn003.pdf.

Beginning with the article in 2001 on dispelling the process myth and the 2005 publication on stretching Agile to fit CMMI Level 3 by David Anderson, Agile and CMMI can be used together and complement each other. There is a clear and detailed history of Agile and CMMI methodologies and the authors make a compelling argument that Agile and CMMI work well together. This perspective from the SEI is compelling. The point made clearly is that there is a misperception that needs to change in using a mixed implementation of Agile and CMMI. The authors go further to state that there is a synergy in using these methodologies together, and that the best of both camps can be utilized. A primary goal is automation to eliminate human error.

Guckenheimer, S. and Perez, J.J. (2006). *Software engineering with Microsoft Visual Studio Team System, Adobe Reader*. Boston, MA: Addison-Wesley Professional.

"The Value-up Paradigm" is the title of the first chapter in the book Software Engineering with Microsoft Visual Studio Team System. This presentation of an Agile perspective through Microsoft's Visual Studio Team System (VSTS) that is an integration of many integrated development tools that are extensible. Competitive advantage becomes an issue with counties like India that take advantage of technologies like CMMI. Builds of the system are incremental and integrate with all parts of the system including testing, metrics, reporting and defect tracking. Bringing all these development systems into a cohesive extensible system that is automated offers a strong solution. Consistency through and across projects for reporting and metrics offers a sound foundation for judging the status of the project. Manual generation of software process is time consuming and in many cases will not be done. An automated solution like this makes development organizations realize the potential gain of tracking and integrating software processes. This publication is a culmination of what David J. Anderson presented in his 2005 paper on Team system.

Howard, G. A. (2007). *Software assembly line*. (Master's Thesis) Retrieved from Dayton Memorial Library, Regis University. (165.236.235.140/lib/GHoward2007).

The software assembly line is a practical application of software and hardware automation. This study to prove 21CFR820.75 process validation is a good example of agile methodology principles. This is relevant to process good standards and software process improvement as well as format for the current thesis format requirements.

Humphrey, W. S. (1995). A *discipline for software engineering*. Boston, MA, Addison Wesley Longman Publishing Co., Inc.

This book is part of the foundation for the CMM. The personal software process (PSP) and team software process (TSP) has greatly contributed to SPI overall. Disciplined software development is defined and explored in detail for both academics and businesses. Previous work from many fields including statistical analysis is used in this book. Procedures outlined in the book are proven methods in the text and have been used in the classroom. As a seminal work in software engineering, this book outlines a solid opportunity for established developers to define, measure, and analyze the PSP to lead to more predictable, cheaper and mature software.

Humphrey, W. S. (2005). "Why big software projects fail: The 12 key questions." *Crosstalk: The Journal of Defense Software Engineering.* 18(3), 25-31. Retrieved December 28, 2008, from http://la-acm.org/Archives/laacm0512-Why-Big- SW-Projects-Fail0503Humphrey.pdf.

Beyond the CHAOS report, there remain findings that software organizations still experience a high rate of failure. The intent of this article is to point out weaknesses in order to pinpoint possible improvements for any organization large or small. A primary contributor to the failure in software is planning. This synthesis of data can demonstrate some problems and possible solutions with using SPI in software organizations.

Humphrey, W. S. (2007). "Software process improvement – A personal view: How it started and where it is going." *Software Process: Improvement and Practice*, 12(3), 223-227. Retrieved December 31, 2007, fromhttp://www3.interscience.wiley.com.dml.regis.edu/cgi-bin/fulltext/114121497/PDFSTART.

Watts Humphrey started nearly 54 years ago and later led much of the original CMM. This paper is a history, general overview, and projection as to what may happen with SPI. His interesting history starting at MIT and then working at IBM complements his already spectacular career as an educator and innovator. In working with a level 5 organization, Humphrey came upon a situation where the management was at level 5, but the development group was at more an initial or level 1 maturity. He concluded that this was partly due to the fact that CMMI defined what “should” be done, but not “how” to do it. From this, he developed what he is probably best known for: the Personal Software Process (PSP) and Team Software Process (TSP). In addition to this, several practices came out of the work that he did with the SEI and have been published and taught as a concept called self-directed teams and requires cultural change and buy in by management. Resistance to change is great and a number of other factors keep SPI initiatives from being adopted even though they are proven to help. It is inevitable according to Humphrey that these initiatives be adopted.

Jacobson, I., Ng, P. W., et al. (2007). "Enough of processes – Let’s do practices." *Journal of Object Technology*, 6(6), 41-67. Retrieved May 7, 2008, from http://www.jot.fm/issues/issue_2007_07/column5.pdf.

There is a trend to better apply and standardize implementations of processes designed up until now. There are the current processes, commonalities, and the future era of practices fostering change. Processes all have basic problems that stem from there not being a standard are often based on conflicting expert opinion. EssWork is geared to

balance process and enable practice to promote collaboration. From the *Journal of Object Technology* this article promotes an agile methodology where it is extensible, adaptable, and capable. Jacobson recommends change that is spelled out in simple terms based on a history of overemphasizing process in the past for software firms. The recommendations are to integrate to this practice model and implement automation to eliminate mundane tasks and promote reuse. The goal is to create ways of working more efficiently and effectively. This method offers a way to mix and match ideas and best practices. These different ways of practicing software development will address the individual organization, project preference, or risk. Essential Unified Process (EssUP) is a freeware that integrates with Eclipse, Microsoft Team Foundation Server (TFS), and JIRA. This is an example of a solution that provides a practical way to achieve software development standards.

Jones, L. G. and Soule, A. L. (2002). "Software process improvement and product line practice: CMMI and the framework for software product line practice." (Report No. CMU/SEI-2002-TN-012). Hanscom AFB, MA. Carnegie Mellon University. Retrieved July 11, 2009, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.7278&rep=rep1&type=pdf.

Product line practices in relation to SPI and CMMI are discussed in this article. This paper promotes CMMI and the processes and practices that are products of it. Configuration management is a central focus of this article. Also, a framework aside from CMMI is proposed, and a comparison of the two frameworks is analyzed.

Jørgensen, M. and Moløkken, K.. (2006). "How large are software cost overruns? A review of the 1994 CHAOS report." Information and Software Technology, 48(4), 297-301. Retrieved September 1, 2009, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.2578&rep=rep1&type=pdf.

Research presented about the CHAOS report is discussed in depth in this article. It casts doubt on the 1994 CHAOS report and tries to explain those findings. The conclusion of the study indicates that the original CHAOS report on cost overrun of 189% was too high. Reproducing the CHAOS report study today is difficult since there was an incomplete description of how the study was done. Perceived improvements in the software industry may be due to the original reports of cost overruns being too high.

Kannenberg, A. and Saiedian, H.. (2009). "Why software requirements traceability remains a challenge." *Crosstalk: The Journal of Defense Software Engineering, 22(5), 14-21.* Retrieved September 1, 2009, from http://www.stsc.hill.af.mil/crosstalk/2009/07/0907KannenbergSaiedian.html.

According to this article, traceability methods and tools fall short based on today's needs. Complex environments like weapons systems, medical devices, and aircraft demand a better standard in software traceability tools. Traceability is defined as the ability to follow and describe requirements through the software development life cycle. The research question proposes that there be better traceability solutions established for the software industry.

Know thyself is the first step to successful knowledge management. (2005, Oct. 26) Knowledge@ W. P. Carey. Retrieved from **know**ledge.wpcarey.asu.edu/article.cfm?articleid=1122.

This is a knowledge management article that explores the need for employees to understand themselves in order to be more successful. Based on the importance of wisdom and each individual's knowhow in a company, the author's point is to use this knowledge to add value.

Laporte, C. Y., Alexandre, S., et al. (2008). "Developing international standards for very small enterprises." *Computer*, 41(3), 98-101. Retrieved May 21, 2008, from http://profs.logti.etsmtl.ca/claporte/English/VSE/Publications/Developing%20International%20Standards_IEEE_Comp_March_08.pdf.

Small organizations are in need of solutions that are fulfilled for large organizations. Statistics for small organizations are used to prove the point that there is a valid need for SPI standards for small enterprises. Compared to large organizations, it is more difficult to implement software standards and solutions in small organizations. The 2004 International Organization for Standardization/International Electrotechnical Commission Joint Technical Committee 1/Sub Committee 7 (SC7) aids in supporting this hypothesis and works to produce results. From this commitment Working Group 24 was formed to solve many of the problems presented in the paper.

Leedy, P. D. and Ormrod, J. E. (2005). *Practical research planning and design*. Upper Saddle River, NJ Kevin M. Davis.

Developing skills to write and design research studies is outlined in this book. Different research types and methods are explored. This is mainly for reference purposes and is intended for thesis or dissertation writing. Many good examples are presented in the text.

Lewis, B. (2009). InfoWorld (2008, February 23) Re: Models, methodologies, menus and recipes [Web log message]. Retrieved February 24, 2009, from http://weblog.infoworld.com/lewis/archives/2008/02/models_methodol.html.

This article is based on an interview of Hillel Glazer about his publication on dispelling the process myth in 2001. The conversation likens CMMI to the "what" in process improvement or a menu, whereas, Agile is likened to the recipe or the "how". This goes back to the original Glazer article and the Bob Lewis expands on this idea by likening CMMI to a restaurant reviewer. Ultimately, the conclusion is that CMMI and its goal is to reduce variability.

Liu, J., Chen, V., Chan, C., & Lie, T. (2008). The impact of software process standardization on software flexibility and project management performance: Control theory perspective. *Information & Software Technology*, *50*(9/10), 889-896. doi:10.1016/j.infsof.2008.01.002.

The control theory perspective is used to analyze the impact of process standardization on flexibility and performance. The triangle of schedule, cost, and requirements in software development is referenced. Standardization is defined as the

consistent use of techniques, tools, methodologies, and procedures. There is a clear contention between controls and flexibility. This study advocates flexibility. As a side note, the authors argue that there is little empirical evidence on CMM popularity. In addition, three SPI areas in the literature are identified: 1.) descriptive; 2.) prescriptive 3.) and reflective. The hypothesis of this paper is that standardization has a relationship with performance and flexibility with projects. The results are positive and the question of how software development is going to enable flexibility is left for further research.

McHale, J. (Feb. 26.2003). "The case for using TSP with CMM/CMMI." Paper presented at a meeting of the U. S. Department of Defense, Boston, MA. Retrieved June 20, 2009, from http://www.sei.cmu.edu/library/assets/tsp.pdf.

Presentation used for establishing the use of TSP with CMMI. This publication provides a good overview of both TSP and CMMI methodologies. Useful graphics and statistics pertaining to the use of TSP with CMMI are available. Conclusions provided suggest the ultimate use of CMMI and not TSP alone.

Neville, K., Hoffman, R. R., et al. (2008). "The procurement woes revisited." *Computer Society*, 23(1), 72-75. Retrieved December 28, 2008, fromhttp://www.computer.org.dml.regis.edu/portal/web/csdl/abs/mags/ex/2008/01/mex200 8010072abs.htm.

This article on human-centered computing and engineering is about how technology design and development falls short. There are many good quotes and is a reference that illuminates the challenges that we face with software development.

Niazi, M., Wilson, D., & Zowghi, D. (2005). A maturity model for the implementation of software process improvement: an empirical study. *Journal of Systems & Software*, *74*(2), 155-172. doi:10.1016/j.jss.2003.10.017.

In introducing SPI practices, Niazi points out that there is a lack of strategy in SPI initiatives today. He believes there is always a method “what” needs to be done. However, there is not a “how” the implementation is to be done. Prior studies identify four factors that can be considered having an impact on SPI implementations: 1.) reviews; 2.) standards and processes; 3.) staff experience; 4.) training and mentoring. This study carries on this research and extends it. Critical success factors (CSF) are used in this study using 20 companies and 24 interviews were conducted. Also, four research questions, positive and negative, about both the literature and empirical study are asked for in this study. According to this paper, little empirical evidence so far has been done on SPI implementations. This research methodology in this study uses content analysis where common themes are identified and categories are determined for what are the positive and negative influences on SPI. The adaptation of CMMI is used to create a maturity model that is tailored to what the authors of this study need.

Paulk, M. C. (2001). "Extreme programming from a CMM perspective." *Software*, 18(6), 1-8. Retrieved December 31, 2007, from ftp://ftp.sei.cmu.edu/pub/documents/articles/pdf/xp- from-a-cmm-perspective.pdf.

Mark C. Paulk is one of the original writers of the CMM process documentation, seminal work to the CMM methodology. Extreme programming is defined in this

publication and is compared in depth to the CMM. According to Paulk, the CMM common sense and with minor tailoring can be implemented in radically different environments large and small. Even though the CMM documentation is lengthy (500 pages) this publication describes it as easily broken down into 52 statements or goals describing the 18 key process areas (KPAs). These are listed and in a table indicating what levels of the CMM they correspond with. In comparison, extreme programming (XP) is described as originating from Beck, Jefferies, and Cunningham and is typically for smaller teams. Shorter iterative cycles are used with four activities: 1.) coding; 2.) testing; 3.) designing; 4.) listening. Some other aspects are continuous communication with the customer, a minimalist solution with functional and unit testing. Paired programming is a controversial topic, but this is dispelled noting that research has shown that it decreases defects and cycle time. Project velocity is defined as the number of stories considering size that can be implemented in a cycle. CMM is clearly a management methodology, while the XP methodology is for teams.

Paulk, M. C., Curtis, B., et al. (1993). *Capability maturity model*.(Report No. CMU/SEI-93-TR-24/ESC-TR-93-177). Pittsburgh, PA. Carnegie Mellon University. Retrieved December 31, 2007, from http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA263403.

This is the original publication of the CMM from the SEI that defines the maturity levels and key process areas. This is well organized, defined and detailed with the processes and principles of CMM. There is a synthesis of what should be done for mature practices in software process improvement. The direction of CMM is mapped out and a future direction of the project is included.

Phongpaibul, M. and Boehm, B. (2006). "An empirical comparison between pair development and software inspection in Thailand." International Symposium on Empirical Software Engineering, 2006, 85-94, ACM: 1-59593-218-6/06/0009. Retrieved September 7, 2009, from http://delivery.acm.org.dml.regis.edu/10.1145/1160000/1159749/p85-phongpaibul.pdf?key1=1159749&key2=4653732521&coll=Portal&dl=GUIDE&CFID=51826041&CFTOKEN=99987352 .

Publication with Barry Boehm centered on XP paired programming and software inspections. This has detailed information on both concepts as well as a comparison of them. The comparison is based on benefits versus costs. The ocus on Thailand allows the paper to detail the pros and cons. The effect on quality is the primary deliverable of this project and is the basis of the papers research question.

Request for proposal (RFP). On *www.Google.com*. Retrieved September 1, 2009, fromhttp://www.google.com/search?hl=en&lr=&defl=en&q=define:Request+for+proposal&ei=MQUTS6HsBNaOtge0r8ysCQ&sa=X&o i=glossary_definition&ct=title&ved=0CAkQkAE.

This is a collection of definitions from various sources of RFP: A document that is used in business to detail, formalize, solicit proposals and start procurement procedures for sealed-bid contracts is called a request for proposal (RFP).

Reifer, D. (2000). The CMMI: it's formidable. *Journal of Systems & Software*, *50*(2), 97. Retrieved from Business Source Premier database. Retrieved December 31, 2007, from http://www.sciencedirect.com.dml.regis.edu/science?_ob=MImg&_imagekey=B6V0N-3YN9408-1-1&_cdi=5651&_user=1922016&_orig=search&_coverDate=02%2F15%2F2000&_sk=999499997&view=c&wchp=dGLbVlb-zSkWz&md5=cb94900b420d0a19d102d77b57fd58c6&ie=/sdarticle.pdf.

In the years when CMM was still relatively new it had many critics. This publication is a critique of the methodology and encourages colleagues to critique the methodology as well. The writer states that the consensus to come up with a cohesive representative document for CMM is one of the major problems. Another problem emerged in the large government contractors who all developed integration and the solution models separately, and different architectures were being used. The CMM standard was not matured itself. This ushered in the CMMI that offered a different challenge where its documentation was in excess. The documentation for the staged representation was over 700 pages. It is difficult to reproduce from person to person, because the process is so formidable.

Ryan, S., & O'Connor, R. (2009). Development of a team measure for tacit knowledge in software development teams. *Journal of Systems & Software*, *82*(2), 229-240. doi:10.1016/j.jss.2008.05.037.

Three studies were conducted to compile this analysis. Measurement is used to evaluate team tacit knowledge measure (TTKM). This software development study explores the relationship of explicit job knowledge, social interaction, and tacit knowledge. There is good material on how technology and software process improvement is challenged based on budget and schedule.

Software Engineering Institute. (2005a). *Process maturity profile*. (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2004 Year End Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved December 28, 2008, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2005marCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2004 year end.

Software Engineering Institute. (2005b). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2005 Mid-Year Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved January 14, 2009, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2005sepCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2005 mid-year.

Software Engineering Institute. (2006). *CMMI for development, Version 1.2* (Report No. CMMI-DEV, V1.2/CMU/SEI-2006-TR-008/ES-TR-2006-008). Pittsburgh, PA: Carnegie Mellon University.Retrieved November 28, 2008, from http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf.

This is the original publication of the CMMI from the SEI and details the models beyond what the original CMM document did in 1993. This publication is much longer then the CMM document and also has details on key process areas and its five maturity levels. This framework is intended for development and maintenance of products and services for software organizations. It is composed of models, appraisal methods, and training.

Software Engineering Institute. (2006a). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2005 Year End Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved December 28, 2008, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2006marCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2005 year end.

Software Engineering Institute. (2006b). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2006 Mid-Year Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved December 28, 2008, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2006sepCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2006 mid-year.

Software Engineering Institute. (2007a). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2006 Year End Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved December 28, 2008, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2007marCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2006 year end.

Software Engineering Institute. (2007b). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2007 Mid-Year Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved December 2008, 2008, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2007sepCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2007 mid-year.

Software Engineering Institute. (2008a). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2007 Year End Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved December 28, 2008, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2008MarCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2007 year end.

Software Engineering Institute. (2008b). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2008 Mid-Year Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved December 28, 2008, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2008SepCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2008 mid-year.

Software Engineering Institute. (2009a). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2008 Year End Update). Pittsburgh, PA: Carnegie Mellon University. Retrieved April 28, 2009, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2009MarCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2008 year end.

Software Engineering Institute. (2009b). *Process maturity profile.* (Report No. CMMI v1.1 SCAMPI v1.1 Class A Appraisal Results 2009 Mid-Year Update). Pittsburgh, PA: Carnegie Mellon University. Pittsburgh, Pa: Carnegie Mellon University. Retrieved November 27, 2009, from http://www.sei.cmu.edu/cmmi/casestudies/profiles/pdfs/upload/2009SepCMMI.pdf.

SEI report that shows detailed current statistics and adoption trends for Class A CMMI appraisals in the United States and worldwide. These statistics are current as of 2009 mid-year.

Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., & Murphy, R. (2007). An exploratory study of why organizations do not adopt CMMI. *Journal of Systems & Software*, *80*(6), 883-895. doi:10.1016/j.jss.2006.09.008.

Primarily small organizations do not adopt CMMI because it is infeasible. This study is done by analyzing the sales of an Australian appraisal company. Most research, according to the authors, is about the successes of CMMI and the experiences related to those successes. A review of around 600 papers did not find any research specific to why organizations do not adopt CMMI. A niche in determining other reasons why small

organizations do not adopt CMMI is defined. A small organization is one of 19 people or less. Larger organizations are thought to be able to take on the additional costs of CMMI due to economies of scale. The Agile methodologies are mentioned as a current solution for smaller organizations. Speculation on recasting the CMMI for small organizations would make it more popular. The finding of this paper is in much agreement with other scholarly articles related to this topic that small companies do not adopt CMM due a limited amount of resources and that many of the practices are not applicable to small organizations. Most software-development organizations are small organizations, so there is a niche and a need.

Sutherland, J., Jakobsen, C. R., et al. (2007). ScrumLog. (Sept.4, 2007). Re: Scrum and CMMI Level 5: Themagic potion for code warriors [Web log message]. Retrieved February 8, 2009, from http://jeffsutherland.com/scrum/Sutherland-ScrumCMMI6pages.pdf.

Systematic is a successful organization that is composed of small software development groups. It was initially appraised at Level 5 CMMI after years of development. The organization's situation as a software company and positive application of CMMI along with the Agile Scrum and Lean Software Development makes it an ideal case study. Incorporating Scrum enabled the organization to cater better to their clients' needs and oriented it more the small projects. Two small projects and large projects are included in this case study.

Taking the road less traveled: The CMMI continuous approach.(2003, Mar. 1, 2003). News at SEI. Retrieved May 20, 2004, from http://www.sei.cmu.edu/library/abstracts/news-at-sei/feature41q03.cfm?DCSext.abstractsource=RelatedLinks - 43k - 2003-03-01.

Continuous and staged approaches of CMMI are described and explored. This brief article offers an internal perspective on the adoption option for CMMI. Many organizations are represented by the pros and cons of using either CMMI approach.

Verner, J., Evanco, W., & Cerpa, N. (2007). State of the practice: An exploratory analysis of schedule estimation and software project success prediction. *Information & Software Technology*, *49*(2), 181-193. doi:10.1016/j.infsof.2006.05.001.

There are many contributors to the implementation of successful software projects. This qualitative analysis of software development and the factors that affect the outcome of projects draws conclusions as to why projects fall short or fail. A result of the study is that projects were more successful when project managers were involved early on.

Wall, D. S., McHale, J., et al. (2007). *Case study: Accelerating process improvement by integrating the TSP and CMMI*. (Report No. CMU/SEI-2007-TR-013). Pittsburgh, PA.: Carnegie Mellon University. Retrieved January 14, 2009, from ftp://ftp.sei.cmu.edu/pub/documents/07.reports/07tr013.pdf.

Two important naval case studies are part of a parent organization called NAVAIR. Both case studies use the same basic approach in using PSP and TSP to accelerate the schedule of CMMI implementations. These projects are prime examples of

small organizations building the process from the ground up for an SPI implementation. The results of the study indicate the effectiveness of this approach while cutting the schedules in half for the CMMI performance.

Wilkie, F. G., McFall, D., et al. (2005). "An evaluation of CMMI process areas for small- to medium-sized software development organisations." Software process Improvement and Practice, 10(2), 189-201. doi: 10.1002/spip.223.

Six small-to-medium-sized businesses are compared that used CMMI. A separate smaller framework is drawn out to draw conclusions and data from. Parts of the conclusions are that developers for smaller organizations carry a larger burden and not on the company processes. Medium-sized organizations tended towards process but did not use CMMI fully. The methodology proposed in this study is a watered down framework based on what small-to-medium-sized organizations perceived as beneficial in this study.

Yin, R. K. (2003). *Case study research design and methods*. Thousand Oaks, CA: Sage Publications, Inc.

Case study research is explored in this book focusing on different writing styles and methodologies. The text focuses on writing principles for case studies. This book is intended mainly as a reference publication on how to build a case study analysis based on research.