



UNIVERSITY OF OREGON
APPLIED INFORMATION MANAGEMENT

Presented to the Interdisciplinary
Studies Program:
Applied Information Management
and the Graduate School of the
University of Oregon
in partial fulfillment of the
requirement for the degree of
Master of Science

Contract Management Techniques: Managing the Progress, Quality and Coordination of Outsourced Software Development

CAPSTONE REPORT

**Lorena Lambert
Senior Project Manager
Oregon Department of Transportation**

University of Oregon
Applied Information
Management
Program

March 2005

722 SW Second Avenue
Suite 230
Portland, OR 97204
(800) 824-2714

Approved by

Dr. Linda F. Ettinger
Academic Director, AIM Program

ABSTRACT

Contract Management Techniques: Managing the Progress, Quality and Coordination of Outsourced Software Development

As society's reliance on technology spreads, a growing number of organizations have turned to software development outsourcing to augment their available resources. Using the Capability Maturity Models for Software Engineering and Software Acquisition as a framework, this study identifies practical contract management techniques based on recommendations in selected literature (1990-2005). These techniques are intended for application by project managers to effectively manage the progress, quality and coordination of work in outsourced software development projects.

TABLE OF CONTENTS

ABSTRACT	II
LIST OF TABLES.....	V
CHAPTER 1 - INTRODUCTION	1
BRIEF PURPOSE	1
FULL PURPOSE	3
<i>The Popularity of Outsourcing</i>	3
<i>Keeping Control of the Process</i>	3
<i>Managing the Contract</i>	4
<i>Purpose of this Study</i>	5
<i>Intended Audience</i>	8
LIMITATIONS	9
<i>Approach to the Literature</i>	9
<i>Use of Maturity Models</i>	9
<i>Scope of Research</i>	10
<i>Source of Information</i>	10
<i>Limitation of Timeframe</i>	11
<i>Application of Terminology</i>	11
PROBLEM CONTEXT	12
<i>The Outsourcing Phenomenon</i>	12
<i>The Capability Maturity Model</i>	13
DEFINITIONS.....	16
CHAPTER 2 – REVIEW OF REFERENCES.....	19
<i>Capability Maturity Model Literature</i>	19
<i>Outsourcing and Software Acquisition</i>	21
<i>Research Methods and Styles</i>	28
CHAPTER 3 -- METHOD.....	30
OVERALL METHODOLOGY	30
LITERATURE SELECTION	31
DATA ANALYSIS.....	34
DATA PRESENTATION.....	37
CHAPTER 4 – ANALYSIS OF DATA.....	39
STAGE 1 – KEY CAPABILITIES	39
STAGE 2 – CONTRACT MANAGEMENT TECHNIQUES.....	44
<i>Managing Progress through Project Planning</i>	47
<i>Managing Progress through Progress Reviews</i>	48
<i>Managing Progress through Contract Controls</i>	49
<i>Managing Quality through Process Reviews</i>	52
<i>Managing Quality through Deliverable Reviews</i>	53
<i>Managing Quality through Contract Controls</i>	54
<i>Coordinating Efforts through Relationship Management</i>	58
<i>Coordinating Efforts through Issue Management</i>	59
<i>Coordinating Efforts through Requirements Management</i>	59
CHAPTER 5 – CONCLUSION.....	61
<i>Project Progress</i>	62
<i>Deliverable Quality</i>	62
<i>Coordination of Effort</i>	63
<i>Application of Techniques</i>	63

APPENDIX A	65
APPENDIX B.....	71

LIST OF TABLES

TABLE 1 – KEY CONTRACT MANAGEMENT CAPABILITIES 41
TABLE 2 – KEY PROJECT PROGRESS CAPABILITIES & SUPPORTING TECHNIQUES 45
TABLE 3 – KEY DELIVERABLE QUALITY CAPABILITIES & SUPPORTING TECHNIQUES..... 50
TABLE 4 – KEY COORDINATION OF EFFORT CAPABILITIES & SUPPORTING TECHNIQUES.. 55

CHAPTER 1

INTRODUCTION

Brief Purpose

Due to the ever growing demand for software products and the rapid and sweeping changes in technology, an increasing number of organizations are outsourcing all or part of their software development activities (Whitten, 1995). Though outsourcing promises a number of benefits, if controls are not put in place to manage the contract and the quality of work delivered, the consequences to the project can be dire (Perry & Devinney, 1997).

Negative consequences can be compounded when an organization deploys a formal software development process. Such a process is intended to provide a predetermined series of activities that can boost software quality and lower costs (Fielden, 2001).

However, the use of a software development process requires that the organization have complete control over the development lifecycle and deliverables used. If the selected outsource vendor uses a different process than the organization, project management can become more complex.

This study is designed as a literature review (Leedy & Ormrod, 2001). Using the Capability Maturity Model literature as a starting place, the purpose of this study is to identify a practical set of techniques that can be applied by project managers to effectively manage the progress, quality and coordination of software development work being done through a vendor. Specific activities presented as part of the Capability Maturity Model for Software Engineering (CMM) (Software Engineering Institute, 1994) and the related Capability Maturity Model for Software Acquisition (SA-CMM) (Cooper

& Fisher, 2002) are used to frame the content analysis (Leedy & Ormrod, 2001) of selected resources. The study specifically addresses the following areas:

- ✚ Managing project progress
- ✚ Managing deliverable quality
- ✚ Coordinating efforts between the hiring agency and the vendor

The data analysis findings are presented as: (1) a table of key capabilities that once achieved, are intended to improve a project manager's ability to manage project progress, deliverable quality, and the coordination of work throughout the lifespan of the contract, (2) a list of techniques that can be applied to outsourced software development projects to improve the chances of contract management success in each of the areas identified, (3) and a narrative discussion of each key capability describing the value, the application, and key techniques discussed within the literature.

The study is designed to support any project manager responsible for the creation and management of software development contracts. It should be especially beneficial to those project managers working within an organization that currently employs a formal maturity model or software development process to manage in-house development.

Full Purpose

The Popularity of Outsourcing

As society's reliance on technology spreads, there is an increasing amount of pressure on organizations to produce reliable software applications faster, cheaper and with fewer resources (Stone, 2003). Because of this demand, a growing number of organizations have turned to software development outsourcing in the hopes of getting projects done sooner and at a lower cost than their resources would otherwise allow. By using a vendor to perform all or part of the development activities on a project, an organization can purchase much needed experience in new and challenging technologies, gain additional resources, start project activities sooner, and possibly reduce the overall cost of the effort (Whitten, 1995).

Keeping Control of the Process

However, outsourcing an organization's software development activities is not without risk. In today's software development arena the key to successful projects is in having consistency and specific guidelines for performing tasks (Stone, 2003). Thus, it is critical that software development be managed in a disciplined manner with clear processes and controls.

According to the Capability Maturity Model, a framework designed to provide guidelines for effectively employing a software development process, there is a defined series of abilities an organization can master that will help them to generate this consistency and control (Software Engineering Institute, 1994). As a consequence, many organizations have employed a formal software development process or methodology to help

accomplish these abilities. According to Fielden (2001), such formal software development processes, “boost software quality and lower costs by breaking the development process into manageable chunks...determining the inputs and outputs of each activity, and preserving best practices for subsequent projects” (p.63).

Once an organization has managed to implement such capabilities and improve the overall maturity of their software development process, they are taking a risk to step outside the bounds of tight control to outsource portions of their development. This risk is increased if the vendor does not use such a process. As Perry and Devinney (1997) suggest, “no organization wants to engage the services of a vendor less competent than it is or of a vendor who hasn’t already implemented disciplined processes to meet goals and objectives” (p.2). However, the prevalence of literature available regarding vendor selection suggests that this too can be controlled. Many resources exist to help an organization effectively recruit and select a competent vendor, and various capability assessment tools, such as the Capability Maturity Model, provide certification a vendor can acquire to signify their success in these areas (Perry & Devinney, 1997).

Managing the Contract

If the selected vendor uses a different methodology than the organization, management of the project can become more complex. It can be problematic to ask the vendor to follow the Software Development Process used by the agency because, according to Fayad (1997), new tools and methods introduce confusion and the vendor is likely to feel less able to do the work. On the other hand, if the vendor is allowed to use their own methodology, the organization can potentially lose control over the process and the quality of deliverables used. Consequently, relationships with vendors are often

frustrating because the organization doesn't believe they have control over the development. A belief that can be perpetuated by contracts that are poorly managed (Whitten, 1995).

Furthermore, according to McConnell (1996), "[outsourcing affords] less visibility into the project's progress...and compensating for that lack of visibility requires astute and attentive management" (p.493). Riley (2004) states that in order for outsourcing activities to be successful the organization must maintain a clear process and governance structure with a team of their own individuals who understand the work being done and the processes being used by the vendor.

Purpose of this Study

In order to identify and understand effective techniques for managing outsourced software development, this researcher turns for direction to the work of the Software Engineering Institute (SEI) and the Capability Maturity Model literature. Within the Capability Maturity Model for Software Engineering, SEI includes a number of abilities that an organization can employ that will help to effectively manage vendors doing development work (Software Engineering Institute, 1994). Additionally, SEI has created a Software Acquisition Capability Maturity Model. This model details a set of process maturity levels designed to support the implementation of software acquisition processes (Cooper & Fisher, 2002).

Although the Capability Maturity Model literature is often used as a tool to help organizations assess the quality of their software development environment, it is only intended to identify what an organization needs to do, not how it can be done (Humphrey,

1999). The purpose of this study is to provide a set of practical techniques that can begin to address the question of how these capabilities can be applied. The techniques identified specifically address only a subset of the capabilities described within the Capability Maturity Model literature and focus on:

- ✚ Managing project progress
- ✚ Managing deliverable quality
- ✚ Coordinating efforts between the hiring agency and the vendor

This study attempts to use current literature (i.e., that published near or after the release of SEI's Capability Maturity Model in 1991) to identify a set of accepted contract management techniques. These techniques are intended to support project managers in managing project progress, deliverable quality and the coordination of efforts between the vendor and the hiring agency. Accordingly, this study uses literature review to explore existing materials in order to identify and organize previous findings according to the aforementioned goals of this study (Leedy & Ormrod, 2001).

Following the process for content analysis suggest by Leedy and Ormrod (2001), an initial analysis of the SEI Capability Maturity Model literature is conducted. This analysis will be used to identify key capabilities required for successful contract management. Then, a second stage of content analysis is conducted on selected literature in the areas of project management, software development processes and software development outsourcing. This second stage identifies specific techniques to support contract management within the key capabilities identified from the Capability Maturity Model literature.

As described above, the study is accomplished using a two stage approach. The first stage explores the SEI Capability Maturity Model literature in order to identify key capabilities needed to effectively manage contracts. The resulting set of capabilities are framed as a set of a priori concepts (Stemler, 2001) that are then used as the categorization scheme for the second stage of analysis. The second stage of analysis focuses on a larger pool of selected literature for general contracting, outsourcing and software development processes. The analysis is used to identify a set of practical techniques that can address each of the key capabilities.

The first stage of data analysis provides material that can be structured into a table of key capabilities for a project manager to consider. Table 1 provides the reader with this key set of capabilities, and their relative descriptions, specifically limited to the management of progress, quality and coordination of work. This table alone is useful to bring focus to the daunting maturity framework presented in the Capability Maturity Model literature.

The second stage of analysis provides the researcher with material that can be structured into a list of practical techniques for project managers to use in achieving the capabilities identified by the first stage of analysis. These techniques are listed according to the capability they support and are augmented by a narrative description.

Not all of the identified techniques will be applicable to all projects. And other techniques, which lie outside the scope of this study, will no doubt be required. Still, the list of techniques provided should help project managers to focus their efforts with regard to the management of project progress, deliverable quality, and the coordination of efforts between the vendor and the hiring agency.

Intended Audience

According to Jim Harris (2001), it is the role of the project manager to ensure, “that the various project components are fully recognized, understood and orchestrated to blend with one another” (p.1). This necessitates that the project manager be in consistent control of project integration, scope, schedule, cost, quality, resources, communications, risks, and procurement (which includes contract management). As a rule, if the manager fails to effectively manage any of these areas, the others will be negatively affected (Project Management Institute, 2000).

Given that the outsourcing of software development activities puts the organization and management of much of the work in the hands of an outside vendor, the management of the related contract becomes critical. According to Harris (2002), the contract provides the governance for the project. Thus a project manager can only successfully manage the effort if it is based on a quality contract and supported by sufficient contract management practices. It is the intent of this study to provide project managers with a set of techniques that will help them to be successful in this area of contract management.

Limitations

The Capability Maturity Model was built on the concept that organizations need goals for their software work but that each individual organization would likely find its own way of achieving them. The idea here was that no one individual or group can prescribe a method of software development that will work for all organizations and all software development projects (Humphrey, 1999). In keeping with this belief, this study does not attempt to create a checklist of activities to be used in all outsourcing situations. Rather, by relying on the Capability Maturity Model literature as a contextual framework, it provides a set of techniques from which a skilled project manager can pick and choose based on the specifics of his or her environment and the project activities being outsourced.

Approach to the Literature

A qualitative approach to the literature which makes use of an exploratory, more open-ended style of research is used (Creswell, 2003). This method is used because the intent of the study is to identify an emergent set of techniques from existing literature. The study does not attempt to provide statistical analysis or evaluate a cause and effect relationship between contract management practices and successful projects. Therefore, this qualitative approach to the existing literature is appropriate (Creswell, 2003).

Use of Maturity Models

This study is limited to practical methods for gaining abilities relative to the Capability Maturity Model for Software Engineering (Software Engineering Institute, 1994) and the Software Acquisition Capability Maturity Model (Cooper & Fisher, 2002). Although other maturity models, such as the Capers Jones's model, ISO standard 90001 and TQM

exist, and undoubtedly have valuable ideas to offer (Saiedian & Kuzara, 1995), SEI's Capability Maturity Model is likely the most widely used as a way to assess the sophistication of an organization's software development methods (Stone, 2003).

Scope of Research

The study is further limited in that it does not attempt to address all of the guidelines provided by SEI literature with regard to contracting and software acquisition. Instead, it focuses specifically on those capabilities that are directly related to the management of project progress, deliverable quality and the coordination of efforts between the vendor and the hiring agency. Therefore it does not discuss vendor selection, the merits of outsourcing, or legal issues with regard to contracting. Additionally, topics related to the highly controversial practice of "offshore" outsourcing are eliminated in order to avoid discussions with relation to culture differences and political pressures.

Furthermore, because the Capability Maturity Model is directly related to the effective creation of software products, this study does not consider those contract relationships which involve transferring the majority of an organization's IT services, including maintenance and helpdesk activities, to a third party.

Source of Information

Material for this study is retrieved using online databases (particularly Computer Database, Business Source Premier, and Inspec), professional and academic websites, journals, magazines, professional whitepapers and technical reports. A detailing of the search strategy is located in the Method chapter, Data Collection section. During the first

phase of the study, materials are limited to SEI's Capability Maturity Model literature and sources directly related to the guidelines presented in those models.

Limitation of Timeframe

The Capability Maturity Model became available to the public in 1991, which sets the early limit for literature used in this study (Software Engineering Institute, 1994). During the second phase of the study the scope of literature reviewed is expanded to include software outsourcing, software development processes and lifecycles, methods of software quality control, and literature providing insight or opinion into the application of the Capability Maturity Model. For consistency, the second stage of the study is limited to literature produced near or after the time the Capability Maturity Model was published.

Application of Terminology

Although this study uses the term outsourcing to describe the act of hiring a third party to perform all or part of an organization's software development activities (Merchant Internet), literature using the terms contracting, subcontracting, and software-acquisition is included. Likewise, the phrase software development process is often used in the literature synonymously with software development methodology or even software development lifecycle. For this reason all such terms are used to find appropriate literature.

Problem Context

The Outsourcing Phenomenon

Over the years, outsourcing has become an increasingly popular activity (Rubin, 1997). Originally, outsourcing was most popular in areas related to the data center, networks, and telecommunications. Then the outsourcing of helpdesks and workstation support became wide-spread. Now, organizations have begun to outsource their software development and maintenance activities as well (Rubin, 1997). In essence, outsourcing is simply an answer to the age-old question of whether to buy or build (Jacobson, 2004). According to Whitten (1995), an increasing number of organizations are selecting to buy.

There are a number of ways in which an organization can make use of software development outsourcing. The majority of these methods fall within one of three types of outsourcing: product component outsourcing, process component outsourcing, and total outsourcing (Merchant Internet).

Product component outsourcing involves the employment of an outside vendor to build only a piece of the overall system. Usually the outsourced component is one that is outside of the hiring organization's ability to resource internally. This type of outsourcing is done in order to augment the skills of the organization, but does not attempt to shift overall development activities outside of the organization (Merchant Internet).

Process component outsourcing is used to contract out the functions of one or more process steps, such as system design, construction or implementation (Merchant Internet).

This is likely to be done for similar reasons as product component outsourcing but has a

larger scope and usually involves a larger contract (Gyorkos, Rozman, & Leskovar, 2000).

Total outsourcing refers to the complete shift of all software development activities from the hiring agency to the vendor. This includes design, development, programming, testing and maintenance. Total outsourcing is usually used when the hiring organization does not wish to maintain the resources and skills to accomplish software development in-house (Merchant Internet).

Regardless of the type employed, outsourcing comes with the promise of greater skills, faster resources, more predictability, and a less expensive end product. However, if not properly managed outsourcing software development can bring with it a series of problems (McConnell, 1996). Projects with poor software contract management practices often exceed cost, slip schedules, and are of dubious quality (Marciniak & Reifer, 1990).

The Capability Maturity Model

The Capability Maturity Model was originally created as a potential answer to this outsourcing problem. The United States Department of Defense partnered with the Carnegie Mellon Software Engineering Institute to develop a method for assessing the capability of potential software contractors (Software Engineering Institute, 1994). The idea was to create a software process assessment and evaluation process that would set the bar for qualified contractors (Software Engineering Institute, 1994). With this new bar, and the possibility of large government contracts on the line, contractors began seeking to improve their assessed capability. According to Schaeffer (1998), the model

has since “contributed to widespread success in assisting organizations in improving their efficiency in developing quality software products” (p. 1).

Due to the success of the Capability Maturity Model for Software Engineering, SEI has developed a series of similar models spanning a wide range of software development areas. These models use the same general framework as the original and can be used to focus on more specific functional disciplines. The additional models include the Software Acquisition Model, the Integrated Product Development Model and a number of others that can, and should, be used in conjunction with one another in order to stretch process improvement resources (Schaeffer, 1998).

Every Capability Maturity Model is presented as a hierarchy of five maturity levels, each building upon the last and progressively moving into greater levels of capability. The levels are defined according to the key processes required to achieve them. Within a level, each process is defined according to a set of activities that, when applied collectively, can achieve the related key process (Software Engineering Institute, 1994). As an organization achieves higher levels of maturity, they are said to have a more stable environment and be more likely to turn out successful, predictable software products (Software Engineering Institute, 1994).

Resultantly, as organizations move up the capability levels, many find great success (Schaeffer, 1998). The Hughes Aircraft’s Software Engineering Division, for example, was able to move their business into level 3 of the Capability Maturity Model, and was able to estimate a resulting annual saving of approximately \$2 million (Saiedian & Kuzara, 1995).

Although several such stories can be found, many organizations have had trouble implementing the model. This is because the dense presentation of material and lack of practical techniques make it difficult for them to discern how it could be implemented in their environment (Bamberger, 1997). In fact, according to Gainer (1998), 75% of software development organizations were estimated to be at the “Initial” level of capability in 1998. These organizations would benefit from research that shows the essence of the Capability Maturity Model literature and provides practical ways that it can be implemented in their various environments (Bamberger, 1997).

Although the Capability Maturity Model for Software Engineering and related models span the realm of software development activities, it can be assumed that improvement in any related process area will aide in the creation of a better product (Stone, 2003). And while SEI offers a formal application and certification program to implement the Capability Maturity Model, this all-out approach to improvement may not be for all organizations (Gainer, 1998). The original model was developed with very large organizations in mind (Gainer, 1998). Nevertheless, according to Paulk (1998), “the fundamental concepts are...useful to any size organization in any application domain and for any business context” (p.3). Thus, small and middle-sized organizations may need to focus their efforts on specific process improvement goals within the scope of the model and forego large scale implementation and formal certification (Gainer, 1998).

It is in this light that the scope of this study is decided. The goal is to identify a manageable set of key capabilities within a specific area of application, contract management, and identify techniques with which to implement them.

Definitions

Capability Maturity Model – SEI’s Capability Maturity Model for Software

Engineering (often referred to simply as the Capability Maturity Model) is a framework that provides guidelines for planning, engineering and managing software development. It is intended to improve the consistency and control with which software development projects are implemented within an organization. Thus, project scope, schedule, quality and budget can be predicted with a improved level of certainty (Software Engineering Institute, 1994).

Contract – A contract is an agreement between the seller of goods or services and the buyer in need of such goods or services. It is usually executed in one of three ways: (1) fixed price contracts which involve a predetermined price for a clearly defined product, (2) cost-reimbursable contracts, which are limited to the reimbursement of actual vendor costs and generally include incentives tied to project objectives, (3) time and material contracts, which include aspects of both fixed cost and cost-reimbursable contracts to provide the flexibility of a cost-reimbursable contract while offering more of the cost control of fixed-cost contracts (Project Management Institute, 2000).

Contract Management – “Contract management is the industry term for the development and maintenance of client/partner business agreements in a structured manner” (techlistings.net).

Outsourcing – The act of hiring a third party to perform all or part of an organization’s software development activities (Merchant Internet). Software development outsourcing can range from component outsourcing, which is contracting out only a portion of a

larger development project, to the total outsourcing of all activities associated with the development of a software product (Merchant Internet).

Project Management – According to the Project Management Book of Knowledge, project management is “The application of knowledge, skills, tools and techniques to project activities to meet the project requirements” (Project Management Institute, 2000) Project management includes the management of: project integration, scope, time, cost, quality, human resources, communications, risk, and procurement (Project Management Institute, 2000).

SA-CMM – The Software Acquisition Capability Maturity Model (SA-CMM) is a framework created by SEI, based on the original concept used in the Capability Maturity Model, which provides guidelines for the buyer in a contractual relationship to effectively manage the acquisition process (Cooper & Fisher, 2002).

SEI – The Software Engineering Institute is a research and development center funded by the U.S. Department of Defense for the purpose of helping others to, “make measured improvements in their software engineering capabilities” (Software Engineering Institute).

Software Development Lifecycle – “A conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.”(Whatis.Com, 2005)

Software Development Process – Software Development Processes provide a predetermined series of activities that, when done consistently, should result in a software “product” that can be delivered (Whitten, 1995). According to Fielden (2001), the use of a software process can improve quality and lower the cost of software development projects. This is accomplished by, “breaking the development process into manageable chunks...determining the inputs and outputs of each activity, and preserving best practices for subsequent projects.” Software Development Process is generally used synonymously with Software Development Methodology and is often confused with the term Software Development Lifecycle.

Vendor – for the purposes of this study a vendor is a third party organization hired to complete either all or a portion of a software development activity. Vendors are generally located offsite, but may reside within the hiring organization at least part time. Such vendors commonly are given control of the management of their own processes and activities (Whitten, 1995).

CHAPTER 2

REVIEW OF REFERENCES

The following annotated bibliography provides a brief description of each of the key materials used in the development of this study. The included references are organized into three categories: (1) Capability Maturity Model Literature, (2) Outsourcing and Software Acquisition and (3) Research Methods and Styles. Within each category of literature, the selected references are presented in alphabetical order.

Capability Maturity Model Literature

Bamberger, P. S. (1997). Essence of the Capability Maturity Model. 112-114.

Judy Bamberger is one of the key authors of the original Capability Maturity Model and has provided here a brief commentary on how the model is intended to be used and understood. She begins by describing the original context for which the Capability Maturity Model was intended. She goes on to identify the issues that many organizations have had with regard to implementation of the Capability Maturity Model. Then she presents some ideas on how the model can be better understood and adapted to each specific environment. The article does a good job of presenting the trouble organizations have had understanding and implementing the Capability Maturity Model. It supports the idea of research that provides practical ways in which the key processes defined in the model can be implemented. The content of this article is useful in describing the context within which this study resides.

Cooper, J., & Fisher, M. (2002). Software Acquisition Capability Maturity Model (SA-CMM) Version 1.03. Pittsburgh, PA: Software Engineering Institute.

The Software Engineering Institute expanded upon the large scale success achieved through the development of the Capability Maturity Model for Software Engineering by creating a host of supporting models to address process maturity in other areas of the software development and management process. The Software Acquisition Capability Maturity Model is one such model.

The framework presented is intended to support and assess the buyer's software acquisition process, and includes activities ranging the entire lifecycle of the acquisition effort. The scope of the model begins with pre-contract activities (such as planning, soliciting and vendor selection) and ends with contract conclusion.

As with the Capability Maturity Model for Software Engineering, the model is presented in five maturity levels, each intended to provide a framework for improving acquisition processes. Thus, when applied, the model can allow an organization to more effectively and consistently acquire software products and services.

The material found in this report, together with the Capability Maturity Model for Software Engineering, provides the foundation for stage 1 analysis in this study. Data found in all five maturity levels are considered. The resulting a priori concepts are determined based on those key process areas that relate directly to project progress, deliverable quality and coordination of efforts between the vendor and the hiring agency.

Software Engineering Institute. (1994). The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley.

The Capability Maturity Model for Software Engineering was created by the Software Engineering Institute as a set of guidelines to be used by the United States Department of Defense to assess potential software development contractors. As the government began using the model as a measuring stick, contractors began to employ process improvement activities in the hopes of improving their assessment. The original model was released to the general software community in 1991. This book presents the most recent version of the Capability Maturity Model for Software Engineering in its entirety along with a complete description and technical overview of the model and its use.

The model presented in this book is intended to support the entire software engineering process from beginning to end. It is written from the software development seller's point of view and provides a capability framework with five levels of maturity. Each level has a series of key processes that must be created and improved upon to advance the maturity rating of the software organization. This reference is used to provide general contextual information regarding the Software Engineering Institute and the Capability Maturity Model. Furthermore, the goals, abilities and activities framed within the key process area of "Software Subcontract Management" are used as foundational data for the stage 1 analysis in this study.

Outsourcing and Software Acquisition

AASHTO Highway Subcommittee on Advanced Transportation Systems. (2000). ITS Software: Effective Acquisition Practices. Washington, DC: AASHTO.

The intent of this report is to provide the Intelligent Transportation Systems (ITS) Project Manager with effective software acquisition management practices that can be employed

to improve the chances of acquisition success. Although the material is focused to support the management of acquisition with regard to ITS, the authors have researched software acquisition and engineering techniques both within ITS projects and in other governmental industries. The report was created under the guidance of the National Cooperative Highway Research Program and was conducted through the use of surveys, interviews, and round-table discussions with contractors and transportation officials in government offices across the nation. The goals of this study include: (1) identification of the types of projects being undertaken by governmental organizations, (2) exposure of common problems and issues, (3) and detection of techniques used to address these issues. The techniques described in this report are used as data for the stage 2 analysis in this study.

Air Force Software Technology Support Center. (2000). Guidelines for Successful Acquisition and Management of Software-Intensive Systems (GSAM). Utah: Air Force Software Technology Support Center.

The GSAM is a full set of acquisition guidelines provided for the Department of Defense workforce to standardize and improve the acquisition and management practices performed by defense agencies. The report was created and is maintained by the Department of the Air Force Software Technology Support Center and includes three sections. The first of these sections relates directly to software acquisition practices. Because the Department of Defense is responsible for the creation and management of very large software systems, they have put a great deal of effort into the standardization and improvement of their processes. Although this report is intended as a working guide and training manual for the Department of Defense, it has been well received by the general software development community.

The guidelines provided within this report represent processes identified through lessons-learned and best practices in both government agencies and industry. Chapter 8, Contracting for Success, is used as data for the stage 2 analysis in this study.

Binstock, A. (1999) Outside Development Partners [Web Page]. URL <http://www.informationweek.com/shared/printHTMLArticle.jhtml?article=/757/prout.htm> [2005, January 31].

Binstock provides in this article a brief but valuable treatment of the outsourcing relationship. He focuses on the setup, execution and coordination of the vendor/agency relationship. The article provides a short description of situations where outsourcing partnerships are valuable. It then goes on to discuss methods for building the required relationships. The techniques presented are intended for use with vendors that will be treated as long-term partners and will take on multiple projects with the agency. Although the discussion of each technique is a bit shallow, the overall discussion paints a picture of a tightly integrated relationship. The information presented here is valuable as data for the stage 2 analysis of this study. Specifically, the techniques presented speak directly to the coordination of efforts between the vendor and the hiring agency. The article is printed on InformationWeek.com, a publication directed to the general technology professional.

Choudhury, V., & Sabherwal, R. (2003). Portfolios of Control in Outsourced Software Development Projects. Vol. 14(3), 291-314.

This article is an academic study examining the methods used for control of outsourced projects within five separate information systems organizations. The study found that outcome, behavior, and vendor empowerment controls were used in most of the included projects. Additionally, it finds that although projects began with relatively simple

controls, by the end of most projects a significant number of additional controls were required. For the purposes of this article, the term control is used to indicate an attempt by the hiring agency to ensure that the vendor acts in a manner that is “consistent with achieving desired objectives” (p. 292). It looks at both formal controls, that use vendor performance evaluations and predefined rewards, and informal controls, that rely on social strategies to motivate the vendor. The controls identified support the stage 2 analysis in this study. Although the focus of the identified controls will lend primarily to the coordination of efforts, there is a fair amount of data to support deliverable quality and project progress as well. The authors of this article are affiliated with state universities and the work is published in the Information Systems Research academic journal.

Ito, Y., Ito, H., Obara, K., Izumi, H., & Kawasaki, R. (1994). A Practical Study on Software Quality Management for Sub-Contracted Products. The Fourth European Conference on Software Quality (pp. p. 375-384). Zurich, Switzerland: VDF Hochschulverlag AG an der ETH.

The authors of this paper work for an organization responsible for the development and maintenance of mainframe software in Japan. Software projects in this organization are completed through design by an in-house team and then outsourced for development. The focus is on improving the quality of the deliverables received from the selected vendor. The authors have focused the scope of their effort on a specific project where they have applied the suggested quality control mechanisms. The mechanisms are described and the effect on the resulting system is identified. The paper address four specific control mechanisms: (1) conduct design reviews to clarify specifications, (2) assign a key person on each component to own the quality of the outcome, (3) compare the fault estimations to actual results, (4) use a “quality diagnosis sheet” to analyze the causes of inferior

quality as it arises. The findings of this paper relate directly to this study's deliverable quality area of focus in the stage 2 analysis.

Marciniak, J. J., & Reifer, D. J. (1990). Software Acquisition Management: Managing the Acquisition of Customer Software Systems. NY: John Wiley & Sons, Inc.

Marciniak and Reifer have created what is one of the only complete books dedicated to the management of the software acquisition process. It is recommended by Steve McConnell, author of the very popular development book, *Rapid Development* (1996). This book looks at software acquisition management from both the buyer and the seller point of view, and provides a holistic approach to the process. It begins with three introductory chapters that introduce the topic of software acquisition and provide context with regard to the software engineering environment. The book goes on to describe the creation of a quality contract, including a complete statement of work. Next, it looks at the entire acquisition process describing the roles and responsibilities of the buyer, the seller, and the entire team. Finally, Marciniak and Reifer look more specifically at cost and quality management. The material presented is used as input to the purpose and frame of this study. It is then further used as foundational data for the stage 2 analysis. Data is taken primarily from: (1) the introductory chapters, (2) Chapter 6: The Buyer's Model, (3) Chapter 8: The Team Approach, (4) and Chapter 10: Quality Management.

Perry, W., & Devinney, S. (1997). Achieving Quality Outsourcing. Information Systems Management, 14(2), 23-26.

Perry and Devinney are respectively the executive director and managing director of the Quality Assurance Institute in Orlando Fl. In this article they have applied their expertise with regard to quality assurance to provide a brief discussion of the quality challenges of

outsourcing. They begin by providing a description of some of the challenges faced by outsourcing organizations and then present methods used to identify a vendor and manage the quality of the deliverables received. Once a vendor is selected, the article outlines three primary methods to improve the quality of deliverables: (1) clear requirements definition, (2) formal change management systems, (3) and the linkage of milestones to quality reviews. The discussion regarding vendor selection is not used within this study. However, the remaining material is used to support the purpose of this study and the stage 2 analysis of techniques used to provide quality deliverables.

Project Management Institute. (2000). A Guide to the Project Management Body of Knowledge. Newton Square, PA: Project Management Institute.

The Project Management Institute, a highly respected organization that provides training, publications, and certifications in the area of project management, has created this guidebook. It is intended to provide project managers with a framework for the successful management of projects. The material presented here is used to provide information and definitions with regard to the intended audience for this study.

Specifically, the guide presents project management in terms of nine key areas. One of these areas is project procurement management. This portion of the guidebook provides an overview of: (1) procurement planning, (2) solicitation planning, (3) solicitation, (4) source selection, (5) contract administration, (6) and contract closeout. The information provided with regard to contract administration and closeout is used as data for the stage 2 analysis in this study.

Reifer, D. J. (2002). Software Management. Los Alamitos, California: IEEE Computer Society.

Reifer's book, published by IEEE, is a tutorial done by means of a narrated compilation of original and reprinted papers written by leaders in the field of software engineering and management. This tutorial is in its sixth edition and includes the most current thinking at the time of publication. Although the book in its entirety is valuable as a reference for project managers in the thick of software development, this study uses only Chapter 13: Acquisition Management. This chapter includes three papers, two of which are useful for this study.

The first paper, written by John Marciniak, was written specifically for publication in this tutorial. Marciniak's paper provides a general discussion of the software acquisition process and environment, the vendor selection process, and assessment methods and metrics that can be employed to improve visibility and control. The information provided with regard to assessment methods and metrics act as data to support the stage 2 analysis in this study.

The second paper, written by Clair Brown, David Neale, and Kenneth Nidiffer, was also written specifically for this publication. This paper focuses on the work that can be done to build successful relationships with contractors, and to overcome the intrinsic relationship stresses that are a part of the contracting process. However, the authors include the management of people, processes, methods, tools and systems as part of this relationship management effort. Specifically, the paper addresses the importance of building a relationship of mutual trust, shared objectives and risks, and common understanding. It then goes on to describe five key activities required in the contract

management process: (1) monitor supplier processes, (2) monitor performance factors, (3) conduct reviews, (4) manage risks, (5) and control change. The areas discussed here that are of interest in this study are those related to building a relationship with the vendor in order to coordinate efforts, the monitoring of performance factors, and the use of reviews. This information is used as data to support the stage 2 analysis in this study.

Svennberg, D. (2001). Software Acquisition Management Guidelines. Sweden: Linköping University.

This paper is a master's thesis submitted to the Department of Computer and Information Science of Linköping University in Sweden. It is intended to provide a set of guidelines for managing software acquisition projects that include at least some level of outsourced software development. Guidelines spanning the entire lifecycle of the project are included. Specifically, the thesis addresses: (1) project steering, (2) project management and organization, (3) software acquisition training, (4) requirements management, (5) product evaluation, (6) transition to support, (7) and project close-out activities. The guidelines presented with regard to project management, requirements management, product evaluation, and project close-out activities are used as data for the stage 2 analysis in this study.

Research Methods and Styles

Palmquist, M. (2004) Content Analysis [Web Page]. URL <http://writing.colostate.edu/references/research/content/> [2005, January 23].

Palmquist provides a clear and concise treatment of content analysis practices. The website supplies brief definitions of content analysis along with a summary of the types of analysis that can be done. Specifically, the site discusses conceptual and relational analysis as the primary ways to analyze content. For each of these categories of analysis,

a set of steps for implementing the analysis are provided and examples are given to solidify the ideas. It is conceptual analysis as it is defined here that is used to perform the analysis for this study. Chapter III of this study describes how each of the steps of conceptual analysis are addressed by the researcher.

Stemler, S. (2001) An Introduction to Content Analysis [Web Page]. URL http://permanent.access.gpo.gov/websites/eric.ed.gov/ERIC_Digests/ed458218.htm [2005, January 10].

Stemler's treatment of content analysis provides a practical application approach to using the technique. The paper is published in the Educational Resources Information Center (ERIC) Digest and references key authors in the content analysis field, such as Krippendorff and Webber. It is the definition and description of the use of a priori coding found in this document that is used in the execution of this study.

CHAPTER 3

METHOD

Overall Methodology

Using SEI's Capability Maturity Model literature as a contextual framework, this study uses a qualitative approach, as defined by Creswell (2003), to identify a set of techniques that can be used to improve the management of software development contracts.

According to Creswell (2003), qualitative research is best suited for exploratory, open-ended studies that will result in the discovery of themes within the available data. It is the intent of this study to identify such themes within current literature as they relate to the management of project progress, deliverable quality, and the coordination of efforts in projects where all or part of the software development activities have been outsourced.

This study does not attempt to develop new methods of contract management. Instead it frames a set of contract management techniques that have already been identified and described within the context of academic, professional, and commercial literature available today. For this reason, the study is accomplished using a literature review to identify themes in the theoretical perspectives and research findings of previous studies (Leedy & Ormrod, 2001). The intent is to focus on specific areas of contract management that are identified as critical by the SEI Capability Maturity Model literature, and present a set of techniques to help an organization improve their contract management capabilities in each of the aforementioned areas.

In order to effectively use the Capability Maturity Model literature as a framework, this study is done in two stages. The first examines SEI's Capability Maturity Model literature to identify a set of key capabilities required to improve the management of

project progress, deliverable quality and the coordination of work within a software development contract. These findings are used to construct a set of a priori concepts (Stemler, 2001) that are used in the stage 2 analysis. The second stage uses the previously defined a priori terms to organize and identify a set of techniques within the literature that will support each of the identified key capabilities.

Literature Selection

Stage 1 –

Literature selected for the first stage of this study includes books, technical reports, and articles published by the Software Engineering Institute or by authors directly associated with them. The majority of the literature selected in this stage is pulled from the SEI's website (<http://www.sei.cmu.edu/cmm>). This site makes readily available a number of reports and other SEI publications with regard to the institute's various Capability Maturity Models. In addition to these materials, *The Capability Maturity Model: Guidelines for Improving the Software Process* is used from the researcher's personal library of books. The specific resources used for the initial analysis include:

Cooper, J., & Fisher, M. (2002). Software Acquisition Capability Maturity Model (SA-CMM) Version 1.03. Pittsburgh, PA: Software Engineering Institute.

Software Engineering Institute. (1994). The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley.

Stage 2 –


Literature selected for the second stage of this study includes books, journal and magazine articles, whitepapers, technical reports, and web-sites. Triangulation is used to

improve the validity of the study (Stemler, 2001); therefore, in order to incorporate multiple data sources, no preference is given to any one form or source of literature. For this reason, resources are gathered from academic, professional, government and even commercial sources.

Academic and professional materials are gathered primarily from library databases including: Inspec, EBSCO Host, Business Source Premier, Computer Database, and Computer Source. Materials are acquired through full text availability within the database whenever possible, and are otherwise requested on-line from the Oregon State Library.

Additional materials are identified through use of Ganthed.com (a professional Project Management website), university and government websites, and internet searches using the Google search engine.

Database and Internet searches are done using a relevance sampling technique. According to Krippendorff (2004), this type of sampling is used when the purpose of the study is to address specific research questions. As is prescribed by relevance sampling, materials are identified using a broad set of key words related to the research questions. The resulting list of resources is then reduced through the addition of limiting terms that narrow the focus of the search. Finally, documents are selected for inclusion through direct analysis of the document content (Krippendorff, 2004). The following words and phrases were used in various combinations to identify and limit relevant sources within the above mentioned databases and using the Google search engine.

 Capability Maturity Model (and CMM)

 Contract

- ✚ Contracting Out
- ✚ Management
- ✚ Outsource
- ✚ Outsourcing
- ✚ Quality
- ✚ Software Acquisition
- ✚ Subcontract (and sub-contract)
- ✚ Software Development Lifecycle
- ✚ Software Development Methodology
- ✚ Software Development Process

Initial searches using key words show that a large amount of material is available regarding the primary issues addressed by this study. For example, a search on the Computer Source database for “Outsourcing”, returns 5241 entries. Similarly, a search on Google for “Capability Maturity Model” returns 215,000 results. However, with the application of such limiting words as “contract management”, and “quality control”, the results can be reduced to a much more manageable set.

As is recommended by Leedy and Ormrod (2001), additional resources are located through the bibliographies of acquired sources. Furthermore, online databases are used to identify sources that cite the current reference, and additional searches are done to locate related works done by the same author.

Each of the identified materials is examined to determine if it is appropriate to the selected topic of this study. Articles are selected if they address one or more of the following:

- ✚ A general understanding of contracting, outsourcing, Capability Maturity Models, or software development processes
- ✚ The management of contracts with regard to project progress, deliverable quality, or the coordination of efforts
- ✚ Software Development outsourcing within an organization attempting to employ a Capability Maturity Model or a software development process

Data Analysis

Once relevant data sources have been collected, they are analyzed using content analysis strategy, specifically conceptual analysis strategy as defined by Palmquist (2004).

According to Palmquist (2004), conceptual analysis starts with the identification of research questions that can be answered through the analysis of data in the sample. The research questions being addressed within this study are:

- ✚ How can the hiring agency gain visibility into the progress of project activities done by the vendor?
- ✚ What can be done by the hiring agency to improve the management of deliverable quality throughout the life of the contract process?
- ✚ What methods are available for coordinating efforts between the hiring agency and the contract vendor?

To answer these questions, this study uses a two stage approach to the analysis of selected literature. In the first stage of the study, SEI's Capability Maturity Model literature is reviewed in order to identify contract management capabilities required to successfully manage the project progress, deliverable quality and coordination of efforts between the vendor and the hiring agency. These capabilities are then collapsed into more generic categories that can be used as a set of a priori terms (Weber, 1990) in the analysis of the more general contract management and outsourcing literature identified as part of stage 2. Stage 2 is designed to identify a set of practical techniques that can be employed in order to improve an organization's contract management processes.

Stage 2 analysis is accomplished using the eight steps of conceptual analysis identified by Palmquist (2004). The researcher's approach to each step is described in the subsequent paragraphs.

Level of Analysis – Data is coded according to passages of related text. Wherever possible this coding is done at a paragraph level, but longer passages are used where necessary in order to maintain the integrity of the data context.

Pre-defined or Interactive Categories – The set of a priori terms, as identified in stage 1, are used as a pre-defined categorization scheme (Stemler, 2001). The intended outcome is a distinct set of subcategories, represented as contract management techniques, identified by using an interactive approach to the text and letting the concepts emerge from the data.

Existence or Frequency Coding – Each technique identified through the analysis is coded for existence.

Level of Generalization – The goal of this study is to provide a general set of techniques from which a project manager can choose in order to improve contract management capabilities. It is not designed to statistically evaluate consistency or variances in themes. For this reason, the text is coded generally, according to the meaning behind the passage and does not focus on specific words or phrases.

Rules for Coding – In order to maintain the integrity of each technique identified, the a priori terms identified as part of stage 1 are treated as mutually exclusive (Weber, 1990). If a technique can conceivably fit within two or more a priori concepts, it is coded under the category most directly supported by the technique and any subsequent occurrences of the technique will be coded likewise.

Identified techniques are intended to represent a mutually exclusive set. To ensure this rule is attained the subcategories are compared to identify any duplicates or hierarchical relationships. In the case that hierarchical relationships are identified, the more general terms are used.

In order to ensure consistency, each passage of text is coded twice by the researcher in an attempt to catch any previous inconsistencies (Weber, 1990).

Handling of “Irrelevant” Information – Any techniques or themes identified in the literature that do not directly relate to one of the a priori terms established in stage 1 is considered out of scope and is thus ignored.

Coding the Texts – According to Leedy and Ormrod (2001), the first step in coding is to organize the collected data. The data organization is accomplished by reading through all

of the collected materials and marking passages of text where contract management techniques are identified. These techniques are then reviewed and those not related to the management of project progress, deliverable quality or coordination for work are eliminated from the set. Once these passages are identified and clearly marked, they are reviewed in detail and gradually placed into appropriate groupings (Leedy & Ormrod, 2001) based on the a priori categories from stage 1. At this point the data analysis results are ready to be further analyzed and framed for presentation.

Analyzing the Results – The purpose of stage 2 analysis is to identify a set of emergent subcategories (Weber, 1990) – each representing a possible technique for managing software development contracts. Within the context of each a priori category, techniques are identified according to emergent themes in the passages of text. The final sets of techniques are first organized in tabular format and then summarized in a narrative discussion according to the a priori category being addressed.

Data Presentation

The outcomes of this study are intended to (1) identify a set of key capabilities a project manager must try to achieve in order to successfully manage software development contracts, (2) present a list of techniques, with related narrative discussion, that a project manager can select from to improve his capability in the related area.

The set of key capabilities are identified in tabular format based on the findings of stage 1 analysis. Specifically, these capabilities are intended to provide a more focused, simplified list of capabilities from the Capability Maturity Model literature with relation to project progress, deliverable quality and coordination of efforts between the vendor

and the hiring agency. The resulting table will be structured with the broad categories of project progress, deliverable quality, and coordination of efforts serving as the Y axis and the name and description of the related key capabilities serving as the X axis.

The associated techniques are intended for use as a set of possible tools that can be used according to the environment and characteristics of a given project. These techniques are organized according to the general capabilities identified in stage 1 and are presented in a list with a brief description each technique. These are then augmented by a narrative discussion of each key capability.

The list format is intended to provide project managers with a quick reference representing the set of techniques with a brief description of each. The narrative section is then used to provide a discussion of each key capability, describing the value, the application, and key techniques discussed within the literature. Because each project and contracting situation is unique, it is the intent of this study to provide a toolbox of possible techniques from which a project manager can choose.

CHAPTER 4

ANALYSIS OF DATA

As described in the methods section of this study, the analysis of data is done in two stages. The first stage is used to define a set of key capabilities, needed by project managers, which generalize the recommended processes presented by SEI's Capability Maturity Model literature for Software Engineering and Software Acquisition. This is done with regard to three contract management categories: (1) project progress, (2) deliverable quality (3) and coordination of effort. The resulting analysis provides nine capabilities, three in each category, to be used as a priori concepts for the stage 2 analysis, which is intended to provide a set of techniques from which a project manager can select to more effectively manage contracted software development. The results of this second stage of analysis are focused on ways to improve the management of the contract and the resulting relationship. The techniques identified are intended for use in combination with solid project management techniques such as those presented by the Project Management Body of Knowledge (Project Management Institute, 2000).

Stage 1 – Key Capabilities

In order to identify this set of generalized key capabilities needed by project managers, the material presented within two of SEI's Capability Maturity Models is analyzed based on key process areas and supporting activities with relation to the following categories:

- ✚ Managing project progress
- ✚ Managing deliverable quality
- ✚ Coordinating efforts between the hiring agency and the vendor

The analysis is limited to those key process areas with direct relation to the management of contracted software development in the above mentioned categories. Within the Capability Maturity Model for Software Engineering this limitation narrowed the scope of analysis to the level 2, key process area of Software Subcontract Management. The Software Acquisition Capability Maturity Model, on the other hand, is more directly linked with the management of software contracts and therefore eight of the key process areas are analyzed. The key process areas used as data come from levels 2, 3, and 4 and include: (1) Requirements Development and Management, (2) Project Management, (3) Contract Tracking and Oversight, (4) Evaluation, (5) User Requirements, (6) Contract Performance Management, (7) Quantitative Process Management, (8) and Quantitative Acquisition Management. The coded activities within the level 4 key process areas of Quantitative process Management and Quantitative Acquisition Management are included in multiple key capabilities, as they suggest changes to the way activities within other levels are accomplished and do not stand alone. Key process areas from level 5 are not considered because level 5 relates only to the improvement of acquisition processes within the hiring organization and thus does not directly address the management of specific contracts.

The results of this analysis are presented in table 1 below. These key capabilities are intended to generalize the material covered by the Capability Maturity Model literature and are not intended to replace a complete understanding of the model. For organizations working towards the improvement of their formal assessment, these key capabilities represent only a subset of the activities required to achieve increased levels of maturity.

A table of the specific key process areas and activities and how they are mapped to the identified key capabilities are provided in Appendix A.

Table 1 – Key Contract Management Capabilities

Category 1: Project Progress	
Project Planning	<p>Project planning refers to the creation, tracking and regular update of materials that estimate project schedule, cost and resource allocation. These materials can be created by either the hiring organization or the contractor, but must be maintained through regular input from both.</p> <p>The Capability Maturity Model recommends, but does not require, that the vendor create and maintain a project plan that can be used by the hiring agency, in conjunction with their own plan, to manage project progress more effectively. Plans created by the vendor are approved by the hiring agency and then used to manage the work.</p> <p>Status, actual schedule, actual costs, issues, and funding are regularly tracked against the project plan.</p>
Progress Reviews	<p>Regular reviews of accomplishments, milestones, in-progress deliverables and assigned action items are scheduled and held to gain insight into the actual status of work being done by the contractor.</p> <p>The results of such reviews can be qualitative or quantitative in nature, but quantitative measures are required to acquire higher levels of process maturity.</p>
Progress Contract Controls	<p>The controls that will be used by the hiring agency to monitor and control contractor progress are clearly documented within the contract and are accordingly applied.</p> <p>Specifically the procedures that will be used to monitor progress and the deliverables the contractor will be expected to generate are completely defined within the contract.</p>

Category 2: Deliverable Quality	
Process Reviews	<p>The processes, activities and environment used to support the contractor's efforts are evaluated and approved by the hiring agency. Once processes are approved, the contractor is periodically audited for continued compliance.</p> <p>The results of such reviews can be qualitative or quantitative in nature, but quantitative measures are required to acquire higher levels of process maturity.</p>
Deliverable Reviews and Evaluations	<p>Concrete deliverables and other products created by the vendor are evaluated according to consistent and predefined criteria. These reviews are periodically done against user and technical requirements and include end user involvement wherever appropriate. These reviews should minimize duplication with reviews and evaluations already being done by the vendor. In situations where the content or quality is thoroughly evaluated through a vendor process, the products of their evaluation should be reviewed.</p> <p>Key deliverables go through acceptance testing before they are approved and accepted by the hiring agency.</p> <p>The results of such reviews can be qualitative or quantitative in nature, but quantitative measures are required to acquire higher levels of process maturity.</p>
Quality Contract Controls	<p>The evaluation criteria, including evaluation requirements, deliverable requirements, architecture compliance, acceptance criteria, evaluation activities, quality evaluations, and the evaluation schedule are detailed and agreed upon within the contract.</p> <p>When performing deliverable evaluations, the results of the evaluation are compared with the requirements in the contract to determine if the deliverable will be accepted. The contract details the conditions for when deliverables must be revised by the vendor.</p> <p>Specifically the procedures that will be used to monitor progress and the deliverables the contractor will be expected to generate are completely defined within the contract.</p>

Category 3: Coordination of Effort	
Relationship Management	<p>Effective relationship management includes activities to support the following:</p> <ul style="list-style-type: none"> • Development of a cooperative and productive environment involving the vendor, the hiring agency and the end user • Regular communication regarding project planning, status, issues, activities, results, evaluations, and quality • Identification, negotiation and management of critical dependencies • Vendor access to the needs and desires of end users • Definition of roles and responsibilities • Assignment of a dedicated and trained contract manager within the hiring agency
Issue Management	<p>Issues are identified, tracked and managed by the project team in a formal corrective action system. These issues are tracked with a description of the actions taken and the results of those actions through to resolution.</p>
Requirements Management	<p>Requirements to fulfill end user needs and desires as well as requirements of the contracted relationship are developed, clearly communicated and used to direct all other activity. Requirements are baselined and placed under formal change control to carefully manage modifications to any requirements.</p>

Stage 2 – Contract Management Techniques

This second stage of analysis is intended to identify and describe a set of practical techniques that can be used in various combinations to support the a priori concepts identified by the stage 1 analysis. To identify these techniques, 13 sources are evaluated for recommendations on how to most effectively manage contracts in these areas. A list of the sources used for this analysis is provided in Appendix B.

The techniques recommended in the literature are placed into a database, categorized based on the a priori categories from stage 1, and analyzed for emergent themes. Those recommended techniques that apply to the management of progress, quality or coordination but do not fall under one of the a priori concepts, are not included in the final analysis. Outcomes of this analysis concerning techniques to most effectively manage contracts are presented in a series of three tables below (Tables 2, 3 and 4).

Table 2 presents the list of techniques recommended to support the management of project progress. Techniques in Table 2 are divided according to the three related a priori categories from stage 1. These include Project Planning, Progress Review, and Progress Contract Controls. Following the table, additional discussion of each of these categories is presented.

Table 3 presents the key techniques recommended to support the management of deliverable quality. Again, these techniques are divided based on the related a priori concepts from stage 1. These categories include Process Review, Deliverable Review and Evaluation, and Quality Contract Control. Each category is further examined following the table.

Table 4 presents the final outcome which addresses techniques in support of the coordination of effort between the hiring agency and the vendor. Categories from the stage 1 analysis in Table 4 include Relationship Management, Issue Management and Requirements Management. Once again, discussion of each category follows the table.

Table 2 – Key Project Progress Capabilities & Supporting Techniques

Category1: Project Progress	
Project Planning Techniques	
Frequent Milestones	Include frequent milestones in project schedules. These milestones represent work to be done down to the subcomponent level, and are usually not more than 2 weeks apart. This is true of schedules created by both the vendor and the hiring agency.
Scheduled Reviews, Walkthroughs and Inspections	Be sure project plans account for the time and placement of formal reviews, deliverable walkthroughs and inspections. Request that the vendor’s plan include those reviews, walkthroughs and inspections that will be done internally to their process.
Coordinate Plans	Once a plan has been received from the vendor, integrate the content into the master plan being kept by the hiring agency. Additionally, require the vendor to represent key milestones from the master in their plan.
Regularly Updated Vendor Plans	To ensure that the plan is being properly used by the vendor team, require updated versions of their plan on a regular basis. Perform audits against the vendor plan to check accuracy and require re-planning if there is a variance of more than 10%.
Progress Review Techniques	
Progress Demos	Once construction has begun, require the vendor to provide periodic product demos. These demos are most useful at product milestones.
Clearly Communicated Review Results	Document review results and share them with the vendor. Share the information both in writing and verbally to allow for clear communication and the opportunity to address questions or concerns. Log any action items resulting from the review and track them through to completion.

Incremental Releases	Use an evolutionary release strategy to divide the work into small manageable pieces. The ability of the vendor to meet the incremental release dates shows progress, and the timeliness and cost of releases can be used to identify trends.
Formal Milestone Reviews	In addition to progress reports and other methods to evaluate on-going progress, hold formal reviews of the entire project effort at key milestones. Identify these milestones at the beginning of the project and plan to discuss: risks, dependencies, schedule, cost, performance, project changes, standing action items, constraints, assumptions, and critical success factors. Provide formal approval at these milestones through the signature of the hiring agency Project Manager.
Progress Reports	Require periodic progress reports from the vendor including at the least project schedule and cost updates.
Review In-Progress Deliverables	Require the vendor to send in-progress versions of the deliverables on a weekly or bi-weekly basis. This should include paper deliverables and actual product code.
Hold Regular Meetings	As with an internal project team, it is advisable to hold regular project status meetings with the entire project team including key players from the vendor agency. This can be accomplished through conference calls or in-person status review meetings every 1 to 2 weeks. Discuss progress, performance, and current issues. Consider holding review meetings at the vendor facility.
Track All Meaningful Progress	Track progress with regard to the project's critical success factors besides schedule and cost. These areas might include such things as the development of a support environment, creation of documentation, or software performance. To identify the areas of critical importance within a project use either the Practical Software Measurement (PSM) or the Goal-Question-Metric (G-Q-M).
Independent Verification and Validation	Bring in independent reviewers to determine if project progress is as reported by the vendor.
Measure Progress Quantitatively	<p><i>Development Process</i> – Compare the number of successfully completed components and configuration items against the total number planned for the final product (Marciniak & Reifer, 1990).</p> <p><i>Test Progress</i> – Track the number of successful tests to the number of tests performed and the number of tests planned (Marciniak & Reifer, 1990).</p> <p><i>Documentation Progress</i> – Track the progress of documentation</p>

	<p>against the schedule. The three areas of documentation to track are product, process, and support. When product documents fall behind the effect is fairly immediate, but those designed for process or support will represent potential problems later in the project (Marciniak & Reifer, 1990).</p> <p><i>Contractor Staffing</i> – Compare the vendor’s staffing rate to the expected need and track overtime rates (Svennberg, 2001).</p> <p><i>Schedule Progress</i> – Measure the number of milestones and project deadlines that are met versus those that are missed (Svennberg, 2001).</p> <p><i>Earned Value</i> – Predict final costs and schedule within a predictable range of values based on the actual schedule and costs as early as 15-20% into the project (Svennberg, 2001).</p>
Progress Contract Controls Techniques	
Contracted Milestones	Be specific in the contract to identify project milestones and important project deadlines.
Incentives and/or Penalties	Include progress incentives and/or penalties within the contract. It is recommended that incentives be used more heavily than penalties as they provide a more positive working relationship with the vendor. However, penalties provide a valuable balance to the any incentives used. Incentives and penalties are usually tied to schedule and cost.
Multiple Contract Awards	Break the project into small, manageable pieces and contract the sections individually. This provides the hiring agency with a way to reward the vendor with future contracts if they are successful.

Managing Progress through Project Planning

The schedule is used to drive a project and has a direct impact on cost and delivered scope. Therefore, unrealistic scheduling has the potential of causing more problems than any other activity (Marciniak & Reifer, 1990). Thus, effective project planning must occur regardless of the use of an outside vendor. However, the introduction of a contract requires the management of two organizations that will each need plans that can be coordinated and used together towards one common goal.

According to Brown et al (2002), it is important to develop a contract management plan early in the process. Although planning requirements can be built into the contract, Brown et al (2002) recommends that they be built into an initial planning deliverable that will be created through a joint effort between the vendor and the hiring agency. This ensures coordination of plans, starts the relationship on a positive collaborative note, and allows for changes to the plan without formal contract adjustments.

Although this research does not uncover specific metrics for measuring the success of project planning, it is clearly recommended by multiple sources to regularly audit the plan compared to the actual schedule. The results of these audits should be used to update the master plan held by the hiring agency, and ensure that the vendor's plan is being properly maintained. According to Svennberg (2001), the schedule should be updated whenever the variance becomes more than 10%.

Managing Progress through Progress Reviews

According to McConnell (1996), “the most significant risk associated with outsourcing is probably the loss of visibility needed to ascertain the project's progress” (pg. 499). In fact, it is common for vendors to report that the project is on time right up until the scheduled day of delivery and then require a substantial extension (McConnell, 1996). For this reason, it is critical that the hiring agency have methods in place to gain visibility into the progress of work being done.

In order to gain this visibility there are a number of methods that can be applied early in the process to improve the agency's ability to directly observe vendor behavior (Choudhury & Sabherwal, 2003). The work done by Choudhury and Sabherwal (2003)

shows that in the case studies of five outsourced software development projects, a number of the organizations found success using frequent conference calls, regular team meetings, and delivery of in-progress products and all five organizations found it beneficial to have vendor personnel located on site.

Managing Progress through Contract Controls

Although the development of a quality contract cannot cause a successful outsourcing relationship by itself, it is the only legal and binding means of control. It is important that the contract be complete and specific about deliverables, schedules, milestones and quality requirements (Binstock, 1999). The techniques presented here are only intended to supplement the control mechanisms provided by standard contracting methods.

The most popular means of controlling progress through the contract are to apply contractual incentives and penalties. Both are considered to be effective means of motivating the vendor. However, according to Marciniak and Reifer (1990), and AASHTO (2000) incentives can be much more effective than penalties because they promote a positive working relationship and can motivate the vendor to continuous improvement and innovation more effectively than penalties. A common incentive used is to place a portion of the profit into an incentive program. The vendor then earns this incentive through saved cost through schedule reduction or innovative ideas. It can be a good practice to pair this with a penalty for late delivery to keep the vendor from stretching the schedule to compensate for lost revenue if the bonus is not achieved (Marciniak & Reifer, 1990).

Table 3 – Key Deliverable Quality Capabilities & Supporting Techniques

Category 2: Deliverable Quality	
Process Review Techniques	
Vendor Process Evaluation and Approval	Evaluate and formally approve the processes that will be used by the vendor to manage progress, ensure quality, or to communicate with the hiring agency. Although the hiring agency must determine those processes that are most critical for success, it is recommended that at least these vendor processes be reviewed: (1) subcontract management, (2) deliverable and code inspection, (3) risk management, (4) quality control/assurance, (5) configuration management, (6) integration management, (7) test planning, (8) and testing
Assisted Vendor Process Improvement	If evaluation of vendor processes finds key processes to be non-existent or lacking, provide the vendor with support in the creation of an adequate process. This enables vendor self-control early in the project by providing them with processes that are likely to turn out acceptable deliverables.
Independent Verification and Validation	Bring in independent reviewers to determine if the vendor environment and processes are well formed, consistently applied and appropriately documented and followed.
Pilot Project	For long term contracting relationships, or large projects, begin the contract relationship with a small pilot project. Use the results of this pilot to evaluate vendor processes and their compatibility with those of the hiring agency. Based on these findings, choose to continue the relationship, use a different vendor, or assist in the modification of vendor processes.
Deliverable Review Techniques	
Acceptance Testing	Use internal staff to perform acceptance testing on all software products delivered by the vendor. Require the vendor to provide an on-site liaison to be present to answer questions while acceptance testing is being conducted.
Clearly Communicated Review Results	Document review results and share them with the vendor. Share the information both in writing and verbally to allow for clear communication and the opportunity to address questions or concerns. Log any action items resulting from the review and track them through to completion

Formal Deliverable Reviews	Hold formal product reviews at the completion of key deliverables. Ensure that deliverables function appropriately and are supported by the appropriate documentation. Prepare checklists in advance based on the identified requirements for quality, reliability, continuity of operation, documentation, and functionality.
Independent Verification and Validation	Bring in independent reviewers to determine the quality of the products delivered. For software deliverables, the independent test team will identify the limits of the code showing when and if it will fail.
Quality Diagnosis Sheets	Require the vendor to deliver code with a self-assessed quality diagnosis sheet. This form is designed by the hiring agency and is used to provide an analysis of remaining bugs, a justification of delays in quality improvement, a review history, and a description of any remaining problems.
Simulation Tools	Require the vendor to use simulation tools to model the expected runtime performance of larger systems. This allows the hiring agency to assess the ability of the proposed design to support performance requirements before spending the time and money to build it.
Walk-Throughs and Peer Reviews	Use peer reviews and walk-throughs to verify that the project deliverables meet requirements, are complete, and are consistent with previous deliverables. Use these methods to identify defects early in the process before they are propagated.
Requirements Trace	Trace the attributes of each deliverable back to the requirements provided to the vendor. It is important to track compliance to the entire set of requirements early on and throughout the life of the project.
Measure Quality Quantitatively	<p><i>Problem Reports</i> – If reports are kept with regard to the problems found while testing the delivered code metrics can be gathered regarding their prevalence and eventual resolution. A problem trend curve and an assessment of the number of problems in comparison to the total lines of source code can be used to determine if quality is improving as development continues (Marciniak & Reifer, 1990).</p> <p><i>Acceptance Test Results</i> – Evaluate the number of successful tests compared with the number of errors found (Svennberg, 2001).</p> <p><i>Software Maturity Index</i> – The Software Maturity Index (SMI) shows the stability of the delivered product. The product is said to be stabilizing as SMI approaches 1.0 (Svennberg, 2001).</p> <p><i>Evaluation Discrepancies</i> – Track the number of discrepancies</p>

	found while evaluating and testing the delivered products. Evaluate the results to identify trends (Svennberg, 2001).
Quality Contract Control Techniques	
Clearly Defined Acceptance Criteria	Be specific in the contract with regard to the expected processes and quality requirements. Include the acceptance criteria that will be applied to deliverables, the requirements around testing, code reviews, quality, documentation and communication of test and quality assurance results, and potential process audits.
Payments Tied to Successful Review	Tie payment for deliverables directly to the appropriate reviews and approvals. Use credits against future payments and delayed payments as penalties for work that does not meet the quality requirements.

Managing Quality through Process Reviews

If project management processes such as risk mitigation, quality assurance, project planning, subcontract management and issue management are not defined as part of the statement of work, they should be reviewed and approved by the hiring agency before other work begins (Reifer, 2002). Although the vendor is inevitably responsible for the quality of the resulting system, it is important for the hiring agency to understand and even have some control over the processes used to ensure this quality (Air Force Software Technology Support Center, 2000). In fact Choudhury and Sabherwal (2003) suggest that by evaluating and even assisting in the redesign of certain vendor processes the hiring agency can give the vendor more self-control later in the project.

However, even with initial approval of the vendor process, many authors recommend periodic audits of important processes. In particular, if there is any concern with regard to quality, it is important to pay attention to processes that will be used to prepare testing

(Marciniak & Reifer, 1990). For testing processes to be successful they must be tightly linked to the requirements. Thus test plans must be documented and reviewed using the quality assurance process used for other deliverables (Marciniak & Reifer, 1990).

Another, rather insightful, recommendation for evaluating the processes of a vendor that is likely to become a long-term partner is to engage that vendor in a pilot project to assess their processes first hand in a working situation. The lessons learned from this pilot project can help both agencies to improve processes as well as provide insight into estimating work for future efforts (Binstock, 1999).

Managing Quality through Deliverable Reviews

When software is developed through an outsourcing relationship the responsibility for quality control becomes largely the role of the vendor. Thus the hiring agency must take steps to ensure that quality standards are maintained (Ito et al., 1994). To be successful in managing the quality of deliverables acquired from the vendor, most of the sources reviewed for this study suggest that the hiring agency must at the very least perform acceptance testing which ensures that the product meets the requirements. Some go further to suggest that each requirement provided to the vendor should be traced back to the delivered product to ensure all have been satisfied.

Beyond the basic requirement to test the acquired software, it is commonly recommended that the hiring agency provide clear evaluation and acceptance criteria and then put delivered products through a formal review to determine if they meet the specified criteria. This can be accomplished through formal quality assurance programs, functional configuration audits (FCA), the physical configuration audit (PCA), review of

documentation and test results provided by the supplier (Reifer, 2002) or through predefined checklists based on quality control regarding functionality, reliability, and continuity of operation (Ito et al., 1994).

Furthermore, three sources recommend the use of regular code walk-throughs and peer review to ensure the content of materials are complete and consistent with previous deliverables, and the expectations of the hiring agency.

Managing Quality through Contract Controls

As you will see in the requirements section under Category 3, Coordination of Effort, it is important to include quality expectations, in very measurable and attainable language as part of the requirements that are included as part of the statement of work. Additionally, there are other contractual controls that can be included to help manage the quality of the final product. The first of these is to clearly identify the acceptance criteria and quality milestones in the contract. Once defined, these evaluation criteria can be used, without contention, as the basis for accepting the product (Svennberg, 2001). Once the acceptance criteria are clearly documented and agreed upon, then payment can be tied to the successful review and acceptance of the products delivered (Reifer, 2002).

Table 4 – Key Coordination of Effort Capabilities & Supporting Techniques

Category 3: Coordination of Effort	
Relationship Management Techniques	
Communication	Use a variety of communication techniques to facilitate timely, appropriate and clear communication among all members of the team. These techniques include: (1) travel between the vendor and hiring agency facilities, (2) inclusion of entire project team from both sides in team meetings, (3) a telephone list with contact information for all team members, (4) timely meeting minutes, (5) newsletters, (6) and written correspondence with regard to contract changes or warnings.
Partnership	If future outsourcing efforts are a possibility with the vendor, approach the vendor/agency relationship as a partnership. Attempt to create a long-lasting successful alliance.
Assembling the Project Team	Set up a project team that includes members from both the vendor agency and the hiring agency. Work with the vendor to ensure all project roles are filled and clearly documented.
Team Building	Engage the team in team building activities that include both vendor and hiring agency staff. It is a good idea to hold a project kick-off meeting that brings members of both agencies to the same site.
Personal Incentives	Provide personal incentives and encourage the vendor to do the same. However, do not provide monetary incentives to members of the hiring agency team if members of the vendor team will not receive similar incentives.
Conflict Management	Manage conflict directly and purposefully. Do not use contractual pressures until other methods of conflict resolution have been exhausted.
Co-location of Teams	Improve team relationships, visibility into progress, and ability to observe vendor behavior through stationing the vendor on-site or sending hiring agency staff to the vendor site.
Flexible Control Mechanisms	Build flexibility into contract control mechanisms in order to modify techniques based on the performance of the vendor and the evolving contracted relationship.

Issue Management Techniques	
Management Tools	Implement a formal issues management system to standardize and prescribe the tracking of issues. Create and maintain this system within the hiring agency. Communicate regularly with the vendor with regard to issues and their resolution. Each issue should include: a review of the problem, the reporting method, status, and correction activities.
Issue Resolution	Determine a process for resolving issues and make sure it is known by the entire team. An issue is usually resolved through explanation, correction, or preparation and approval of a change request.
Measure Issue Management Quantitatively	To measure issue management success quantitatively, measure trends around the types and frequency of problems and track the number and time until resolution of open issues.
Requirements Management Techniques	
Requirements Baseline	Create a requirements baseline before the RFP is put out to bid in order to have a clear starting point from which bids can be made.
Quality Requirements	Identify specific quality requirements in clear measurable terms to help vendors understand the definition and level of quality as it is required by the project. Quality requirements can include but are not limited to: reliability, maintainability, transportability, interoperability, efficiency, and scalability.
Requirements Detail	Provide a detailed, unambiguous, and measurable set of requirements that clearly define the needs of the hiring agency. These requirements must be: complete, consistent, verified, clear, achievable, measurable, and bound the functionality required. The set of identified requirements can include constraints, product specifications, quality and control, security, interface, conversion, configuration, and communications, warranty or continuing support needs, and business functionality.
Evaluation Criteria	Establish a set of evaluation criteria with which the vendor will determine if requirements have been satisfied by the resulting system and documentation.
Requirements Extraction and	Use hiring agency staff to extract, document and baseline the initial requirements set. Include these requirements as part of the RFP to

Presentation	assist the vendor in providing a valid bid.
Independent Verification and Validation	Use Independent verification and validation (IV&V) methods to gain an objective evaluation of the identified requirements
Joint Requirements Review	Review the requirements, in detail, with the vendor team before work begins.
Change Management	Manage the changes made to requirements after baseline through a formal change management process. Don't allow changes to requirements after the RFP has been put to bid until after a contract has been negotiated.
Outsourced Requirements	If the agency doesn't have the resources or skills to gather requirements, outsource the requirements gathering effort under a separate contract. Once the requirements are gathered, they can be used to support the larger RFP.
Peer Review	Use peer review to assess the quality and completeness of requirements before putting the RFP out to bid.
Prioritization	Set priorities on requirements to allow for scope changes and flexibility later in the project.
Management Tools	Use formal requirements management tools to ensure consistent, accessible treatment of requirements.
Measure Changes Quantitatively	<p><i>Quantity of Changes</i> – measure the number of changes made throughout the process (Marciniak & Reifer, 1990).</p> <p><i>Effort</i> – track the total effort, time and cost expended to accommodate changes (Marciniak & Reifer, 1990).</p> <p><i>Rework</i> – track the amount of rework required to accommodate changes (Marciniak & Reifer, 1990).</p> <p><i>Rate of Change</i> – measure the overall rate of change. The target value should be $\leq 1\%$ per month (Svennberg, 2001).</p>

Coordinating Efforts through Relationship Management

The discussion around the management of vendor relationships is best framed by Rubin (1997) who said that, “the most startling lesson is that good intentions and a 600-page contract are not enough to ensure the success of outsourcing agreements. The corollary to this lesson is that a successful agreement requires that all parties to it be successful.”

(pg.1)

According to Choudhury and Sabherwal (2003), if the vendor and the hiring agency can be positioned to have similar goals where the success of one becomes the success of the other, then a clan type relationship can occur. In this type of relationship the two can be counted on to strive for a common objective. These same authors also suggest that such a clan relationship is difficult to build in a contracted relationship. However, many other authors reviewed in this study suggest that a tight integration of vendor staff and the personnel of the hiring agency can be very beneficial. To this end, AASHTO (2000) recommends that outsourced projects be treated as a partnership between the two agencies. In such a relationship, both parties commit to partnering which includes (among other things): building a project team including members of both agencies, identifying a mutually accepted process for resolving disagreements, and performing joint reviews of project progress and completion.

As with any relationship, the nature of a vendor/buyer relationship will change over time, thus the controls used to manage the relationship should be flexible enough to adjust as the relationship does (Choudhury & Sabherwal, 2003).

Coordinating Efforts through Issue Management

Issue Management processes, or corrective action systems, are used to capture, manage, and resolve issues and action items identified through reviews, team meetings, technical discussions, audits and other project activities (Marciniak & Reifer, 1990). The management of such issues is not unique to the contracting relationship, but it can be a valuable tool to communicate and manage issues between the two agencies. Unlike other processes, issue management should be done by the hiring agency, except in environments with small projects and only one vendor.

Coordinating Efforts through Requirements Management

The art of gathering requirements is one that is well covered in software engineering literature. "In software, as in all fields of engineering, if the system's requirements are inadequately specified, the project will fail" (AASHTO Highway Subcommittee on Advanced Transportation Systems, 2000, pg.4-2). It is no surprise then that, the importance and process around the extraction and maintenance of requirements is possibly the most discussed capability represented in this study. Although any valid technique for improving requirements is likely to improve a vendor's ability to build an effective system, this study is carefully limited to requirements extraction and management techniques that will directly affect the success of the contract relationship.

From the literature reviewed, it is clear that requirements must be gathered in great detail and used to feed the RFP process. By providing the requirements as part of the RFP the vendor is given a much more concrete way of estimating the work to make an accurate bid (Perry & Devinney, 1997). Additionally, if the project is difficult to define in detail, it

may not be a good candidate for outsourcing. By gathering the requirements ahead of time, the agency can make a more educated decision.

The necessary detail is found in both the characteristics of each requirement and the type of requirements gathered. According to Marciniak and Reifer (1990), requirements should be complete, consistent, clear, achievable, verifiable, and bound the functionality required. They should also represent the needs of the hiring agency regarding constraints, product specifications, quality and control, security, interfaces, conversion, configuration, communications, and on-going support (Perry & Devinney, 1997).

In particular, a number of sources recommend the careful representation of quality requirements. According to Marciniak and Reifer (1990), “Simply to require a high-quality product is ineffective” (pg. 217) . Quality requirements must have the characteristics defined above for any other requirement and should be specific to areas such as: reliability, maintainability, transportability, interoperability, efficiency, or scalability.

Once the extracted requirements are considered complete, they should be baselined. This baseline then acts as the starting point and any subsequent modifications to the requirements are done through a change management process. These changes can then be measured using quantitative metrics.

CHAPTER 5

CONCLUSION

McConnell (1996) suggests that the increasing numbers of organizations turning to outsourcing for their software development activities are doing so in the hopes of attaining numerous benefits. Organizations are attracted by the promise of reduced time to delivery, reduced cost, and access to skills they do not otherwise hold. And while organizations may hope to find escape from the frustrations of managing in-house development by turning it over to someone else, they often find that the management of outsourced software development can be even more complex, requiring astute attention and management (McConnell, 1996).

As this paper reveals, project managers tasked with the control of contracted software development activities need to have a clear understanding of the capabilities and various techniques required to be successful in this arena. This study attempts to address this need through research into the following questions:

- ✚ How can the hiring agency gain visibility into the progress of project activities done by the vendor?
- ✚ What can be done by the hiring agency to improve the management of deliverable quality throughout the life of the contract process?
- ✚ What methods are available for coordinating efforts between the hiring agency and the contract vendor?

The results of analysis of literature selected for this study are framed for project managers in a set of four tables presented in Chapter 4. The tables show that attention to key

processes with regard to each of these questions can provide concrete and actionable ways to manage outsourced software development projects. An overview of the ideas presented in these tables, framed in terms of each of the key concepts in the original research questions, follows.

Project Progress

In order to attain visibility into the progress of project activities the project manager must find ways of implementing controls that will allow for the observation of vendor behavior (Choudhury & Sabherwal, 2003). The first step is to establish a clear and agreed upon project plan that will be used to manage and evaluate the progress of the work done. Then the hiring agency must put into place a series of controls that will allow visibility into the progress of work being done against the plan. Behavior controls such as regular progress reviews, team meetings, the delivery of in-progress products, and even the co-location of vendor and agency staff can be built into the project plan. When these behavior controls are put into place early on, they can help make the work being done by the vendor more transparent to the project manager (Choudhury & Sabherwal, 2003). Finally, some of these controls can be built into the contract to allow for incentives and penalties based on how well the vendor is able to meet project deadlines.

Deliverable Quality

Once the requirements for quality have been defined and handed over, the vendor is given a large amount of the responsibility to maintain quality control. Thus the methods that will be used to manage this quality become critical (Ito et al., 1994). Therefore, it is important for the hiring organization to have outcome controls in place that will help to evaluate the quality of products delivered (Choudhury & Sabherwal, 2003). These

controls should include at the least, formal reviews at milestones, acceptance testing and a trace back to requirements.

Additionally, the hiring organization should have a clear understanding of the processes used internally by the vendor to ensure quality. If these processes are reviewed and approved, and even re-designed at the beginning of the project, the vendor can be given a certain amount of self-control to manage the quality of deliverables (Choudhury & Sabherwal, 2003).

Coordination of Effort

The coordination of effort between the vendor and the hiring agency is accomplished by building a solid relationship in which the needs of the project are understood by both parties and each does their part to achieve the same objective. It is important at the beginning of the relationship to ensure that the project requirements are clearly defined, and the vendor has been given the opportunity to understand their meaning. Additionally, it is important for the hiring agency to build a relationship with the vendor that encourages teamwork where both agencies are committed to forming a successful partnership (Binstock, 1999). Finally, the hiring agency should put into place methods for capturing, communicating and managing issues in such a way that they are understood by all involved and can be resolved quickly.

Application of Techniques

The techniques identified in the analysis of this study should be carefully selected and applied based on the characteristics of the project being outsourced. To apply all of these techniques to a small, single-vendor project would likely create more overhead than

value. However, some subset of these techniques is likely to be valuable for most projects. For example, the analysis shows that the gathering of complete, measurable requirements to feed the RFP process is likely to be of value to any project, because it helps a perspective vendor to provide an accurate bid. However, only large or critical projects need to go through some sort of Independent Verification and Validation process (Marciniak & Reifer, 1990).

Furthermore, within the bounds of a single project there should be some flexibility with regard to the contract management techniques applied. As the relationship builds and trends of success or poor performance can be identified, the need for stringent contract controls may diminish or increase. Although the task of managing an outsourced software development effort should not be taken lightly, research presented in this paper shows clearly that through the application of appropriate management techniques the outsourcing relationship can be a successful one.

APPENDIX A

Table 1 – Cross-reference of Key Capabilities to Capability Maturity Model Activities

Category 1: Project Progress		
Project Planning Capability	<i>Software Subcontract Management</i>	Activity 5: A documented and approved subcontractor's software development plan is used for tracking the software activities and communicating status.
		Activity 7: The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management. Characteristic 4: Critical dependencies and commitments between the subcontractor's software engineering group and other subcontract groups are addressed.
	<i>Project Management</i>	Activity 5: The project team tracks project issues, status, execution, funding, and expenditures against project plans and takes action.
		Activity 7: The project team keeps its plans current during the life of the project as replanning occurs, issues are resolved, requirements are changed, and new risks are discovered.
	<i>Contract Tracking and Oversight</i>	Activity 2: The project team reviews required supplier planning documents which, when satisfactory, are used to oversee the supplier's effort.
		Activity 4: The actual cost and schedule of the supplier's effort are compared to planned schedules and budgets and issues are identified.
Progress Reviews Capability	<i>Software Subcontract Management</i>	Activity 7: The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management. Characteristic 9:

		Action items are assigned, reviewed, and tracked to closure.
		Activity 9: Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure.
	<i>Evaluation</i>	Activity 3: Planned evaluations are performed on the evolving products.
Progress Contract Controls Capability	<i>Software Subcontract Management</i>	Activity 3: The contractual agreement between the prime contractor and the software subcontractor is used as the basis for managing the subcontract. Characteristic 5: The subcontracted products to be delivered to the prime contractor. Characteristic 8: The procedures and evaluation criteria to be used by the prime contractor to monitor and evaluate the subcontractor's performance.

<i>Deliverable Quality</i>		
Process Reviews Capability	<i>Software Subcontract Management</i>	Activity 10: The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure.
		Activity 11: The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure. Characteristic 1: The subcontractor's plans, resources, procedures, and standards for software configuration management are reviewed to ensure they are adequate. Characteristic 3: The subcontractor's software baseline library is periodically audited to assess how

		well the standards and procedures for software configuration management are being followed and how effective they are in managing the software baseline.
	<i>Contract Tracking and Oversight</i>	Activity 3: The project team conducts periodic reviews and interchanges with the supplier team.
		Activity 6: The project team reviews and tracks the development of the software engineering environment required to provide life cycle support for the acquired products and issues are identified.
	<i>Contract Performance Management</i>	Activity 2: The supplier team's engineering process is appraised periodically and trends analyzed.
Deliverable and Product Reviews and Evaluations Capability	<i>Software Subcontract Management</i>	Activity 8: Periodic technical reviews and interchanges are held with the software subcontractor.
		Activity 12: The prime contractor conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure.
	<i>Contract Performance Management</i>	Activity 6: The end user periodically participates in the evaluation of evolving products to determine the satisfaction of operation requirements.
	<i>Evaluation</i>	Activity 3: The project's evaluation activities are planned to minimize duplication of the supplier's evaluation efforts and take advantage of the evaluation results, where appropriate.
Quality Contract Controls Capability	<i>Evaluation</i>	Activity 2: The project's evaluation requirements are developed in conjunction with the development of the contractual requirements.
		Activity 5: Results of the evaluations are analyzed and compared to the contract's requirements to establish an objective basis to support the decision to accept the product or to take

		further action.
	<i>Software Subcontract Management</i>	<p>Activity 3: The contractual agreement between the prime contractor and the software subcontractor is used as the basis for managing the subcontract.</p> <p>Characteristic 3: The requirements for the products to be developed.</p> <p>Characteristic 6: The conditions under which revisions to products are to be submitted.</p> <p>Characteristic 7: The acceptance procedures and acceptance criteria to be used in evaluation the subcontracted products before they are accepted by the prime contractor.</p>

Coordination of Effort

Relationship Management Capability	<i>Software Subcontract Management</i>	<p>Activity 7: The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management.</p> <p>Characteristic 1: The subcontractor is provided with visibility of the needs and desires of the product's customers and end users, as appropriate.</p> <p>Characteristic 4: Critical dependencies and commitments between the subcontractor's software engineering group and other subcontractor groups are addressed.</p> <p>Characteristic 5: Critical dependencies and commitments between the prime contractor and the subcontractor are addressed.</p>
		<p>Activity 11: The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure.</p> <p>Characteristic 2:</p>

		The prime contractor and the subcontractor coordinate their activities on matters relating to software configuration management to ensure that the subcontractor's products can be readily integrated or incorporated into the project environment of the prime contractor.
	<i>Contract Performance Management</i>	Activity 7: Contract performance management activities are performed to foster a cooperative and productive environment among the end user, project team, and the supplier team.
	<i>Contract Tracking and Oversight</i>	Activity 2: The project team conducts periodic reviews and interchanges with the supplier team.
	<i>Project Management</i>	Activity 2: The roles, responsibilities, and authority for the project functions are documented, maintained, and communicated to affected groups.
Requirements Management Capability	<i>Requirements Development and Management</i>	Activity 2: The project team develops, baselines, and maintains contractual requirements and places them under change control early in the project, but not later than release of the solicitation package.
		Activity 3: The project team appraises change requests of contractual requirements for their impact on the products being acquired.
		Activity 4: The project team appraises all changes to the requirements for their impact on performance, architecture, supportability, system resource utilization, evaluation requirements, and contract schedule and cost.
	<i>User Requirements</i>	Activity 2: The end user's requirements and their evaluation criteria are elicited.
		Activity 5: The end user's requirements are provided as input to the project's acquisition requirements development and management activities.
Issue Management	<i>Contract Tracking and Oversight</i>	Activity 7: Any issues found by the project team,

<p>Capability</p>		<p>during contract tracking and oversight, are recorded in the appropriate corrective action system, action taken, and tracked to closure.</p>
	<p><i>Project Management</i></p>	<p>Activity 6: The project team implements a corrective action system for the identification, recording, tracking, and correction of problems discovered during the acquisition.</p>

APPENDIX B

Literature Used to Conduct the Stage 2 Data Analysis

- AASHTO Highway Subcommittee on Advanced Transportation Systems. (2000). ITS Software: Effective Acquisition Practices. Washington, DC: AASHTO.
- Air Force Software Technology Support Center. (2000). Guidelines for Successful Acquisition and Management of Software-Intensive Systems (GSAM). Utah: Air Force Software Technology Support Center.
- Binstock, A. (1999) Outside Development Partners [Web Page]. URL <http://www.informationweek.com/shared/printHTMLArticle.jhtml?article=/757/prou.htm> [2005, January 31].
- Choudhury, V., & Sabherwal, R. (2003). Portfolios of Control in Outsourced Software Development Projects. Vol. 14(3), 291-314.
- Ito, Y., Ito, H., Obara, K., Izumi, H., & Kawasaki, R. (1994). A Practical Study on Software Quality Management for Sub-Contracted Products. The Fourth European Conference on Software Quality (pp. p. 375-384). Zurich, Switzerland: VDF Hochschulverlag AG an der ETH.
- Marciniak, J. J., & Reifer, D. J. (1990). Software Acquisition Management: Managing the Acquisition of Customer Software Systems. NY: John Wiley & Sons, Inc.
- McConnell, S. (1996). Rapid Development. Redmond, WA: Microsoft Press.
- Perry, W., & Devinney, S. (1997). Achieving Quality Outsourcing. Information Systems

Management, 14(2), 23-26.

Project Management Institute. (2000). A Guide to the Project Management Body of Knowledge. Newton Square, PA: Project Management Institute.

Reifer, D. J. (2002). Software Management. Los Alamitos, California: IEEE Computer Society.

Svennberg, D. (2001). Software Acquisition Management Guidelines. Sweden: Linkoping University.

Whitten, N. (1995). Managing Software Development Projects. NY: John Wiley & Sons, Inc.

REFERENCE LIST

- AASHTO Highway Subcommittee on Advanced Transportation Systems. (2000). ITS Software: Effective Acquisition Practices. Washington, DC: AASHTO.
- Air Force Software Technology Support Center. (2000). Guidelines for Successful Acquisition and Management of Software-Intensive Systems (GSAM). Utah: Air Force Software Technology Support Center.
- Bamberger, P. S. (1997). Essence of the Capability Maturity Model. 112-114.
- Binstock, A. (1999) Outside Development Partners [Web Page]. URL <http://www.informationweek.com/shared/printHTMLArticle.jhtml?article=/757/prou.html> [2005, January 31].
- Choudhury, V., & Sabherwal, R. (2003). Portfolios of Control in Outsourced Software Development Projects. Vol. 14(3), 291-314.
- Cooper, J., & Fisher, M. (2002). Software Acquisition Capability Maturity Model (SA-CMM) Version 1.03. Pittsburgh, PA: Software Engineering Institute.
- Creswell, j. W. (2003). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. CA: SAGE Publications.
- Fayad, M. E. (1997). Software Development Process: a necessary evil. Vol. 40(9), 101-103.
- Fielden, T. (2001). Process Makes Perfect. Vol. 23(25), 63.

Gainer Jeff. (1998) Process Improvement: The Capability Maturity Model [Web Page].

URL <http://www.itmweb.com/f051098> [2002, February 1].

Gyorkos, J., Rozman, I., & Leskovar, R. T. (2000) The General Effect of an Integrated Software Product Evaluation [Web Page]. URL

http://www.iscn.at/select_newspaper/assessments/unimar.html [2005, January 12].

Harris, J. (2001) The Project Manager Is the Keystone to Successful Delivery [Web Page]. URL <http://www.gantthead.com/articles/articlesPrint.cfm?ID=18203>

[2005, January 31].

Harris, J. (2002) The Governance of Project Management [Web Page]. URL

<http://www.gantthead.com/articles/articlesPrint.cfm?ID=112870> [2005, January 31].

Humphrey, W. S. (1999) Pathways to Process Maturity: The Personal Software Process and Team Software Process [Web Page]. URL

<http://interactive.sei.edu/features/1999/June/Background/Background.jun99.htm> [1999, April 31].

Ito, Y., Ito, H., Obara, K., Izumi, H., & Kawasaki, R. (1994). A Practical Study on Software Quality Management for Sub-Contracted Products. The Fourth European Conference on Software Quality (pp. p. 375-384). Zurich, Switzerland: VDF Hochschulverlag AG an der ETH.

Jacobson, T. (2004). IT Outsourcing In US Hospitals: Potential Benefits and Risks.

- Krippendorff, K. (2004). Content Analysis: An Introduction to Its Methodology. Thousand Oaks, CA: Sage Publications.
- Leedy, P. D., & Ormrod, J. E. (2001). Practical Research: Planning and Design. NJ: Merrill Prentice Hall.
- Marciniak, J. J., & Reifer, D. J. (1990). Software Acquisition Management: Managing the Acquisition of Customer Software Systems. NY: John Wiley & Sons, Inc.
- McConnell, S. (1996). Rapid Development. Redmond, WA: Microsoft Press.
- Merchant Internet. UK IT outsourcing, software development outsourcing, UK asp web hosting, application, development, provider.
- Palmquist, M. (2004) Content Analysis [Web Page]. URL <http://writing.colostate.edu/references/research/content/> [2005, January 23].
- Paulk, M. C. (1998) Using the Software CMM in Small Organizations [Web Page]. URL http://www.broy.in.tum.de/lehre/vorlesungen/vse/WS2004/1998_cmm-small.pdf [2005, February 1].
- Perry, W., & Devinney, S. (1997). Achieving Quality Outsourcing. Information Systems Management, 14(2), 23-26.
- Project Management Institute. (2000). A Guide to the Project Management Body of Knowledge. Newton Square, PA: Project Management Institute.
- Reifer, D. J. (2002). Software Management. Los Alamitos, California: IEEE Computer Society.

Riley, J. In-House Teams are Vital When Outsourcing. 46.

Rubin, H. A. (1997). Using metrics for outsourcing oversight. Vol. 14(2), 7-14.

Saiedian, H., & Kuzara, R. (1995). SEI Capability Maturity Model's Impact on Contractors. Vol. 28(1), 16-26.

Salwin, A. E. (1998). The Road to Successful ITS Software Acquisition. Washington, DC: Department of Transportation Federal Highway Administration.

Schaeffer, M. D. (1998) Capability Maturity Model Process Improvement [Web Page].

URL <http://www.stsc.hill.sf.mil/crosstalk/1998/05/cmmi.asp> [2005, February 1].

Software Engineering Institute. (About the SEI [Web Page]. URL

<http://www.sei.cmu.edu/about/about.html>.

Software Engineering Institute. (1994). The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley.

Stemler, S. (2001) An Introduction to Content Analysis [Web Page]. URL

http://permanent.access.gpo.gov/websites/eric.ed.gov/ERIC_Digests/ed458218.htm [2005, January 10].

Stone, G. (2003). Making the Most of the Software Development Process. Vol. 34(6), 10-13.

Svennberg, D. (2001). Software Acquisition Management Guidelines. Sweden:

Linkoping University.

techlistings.net. URL <http://www.techlistings.net/xlist/tech/bizsoft/contract?id=1>.

Weber, R. P. (1990). Basic Content Analysis. Newbury park, CA: Sage Publications.

Whatis.com. (2005). systems development life cycle - a Whatis.com definition. [Web Page]. URL: http://searchvb.techtarget.com/sDefinition/0,,sid8_gci755068,00.html

Whitten, N. (1995). Managing Software Development Projects. NY: John Wiley & Sons, Inc.

BIBLIOGRAPHY

Corbin, D. S. (1991). Establishing the Software Development Environment. Vol. 42(9), 28-31.

Gray, C., & Larson, E. (2000). Project Management: The Managerial Process. Boston: Irwin McGraw-Hill.

Hossein, S. (1995). SEI Capability Maturity Model's Impact on Contractors. 16-26.

Keuffel, W. (1991). Build it Right the First Time. Vol. 9(6), 102-105.

King, K. L. (2000). Staffing Strategies for Higher Education Information Technology: A Study of Four Approaches.

Perry, W. E. (1991). Qualifying Software Vendors Ensures Quality. Vol. 10(16), 92.

Plyler, R. W., & Young-Gul, K. (Spr93). Methodology Myths. Vol. 10(2), 39-44.

Software Engineering Institute. (The Team Software Process (TSP) and The Personal

Software Process (PSC) [Web Page]. URL <http://www.sei.cmu.edu/tsp>.

Tayntor, C. B. (1997). An Outsourcing Parable. Vol. 14(2), 66-69.

Topper, A., Ouellette, D., & Jorgensen, P. (1994). Structured Methods: Merging Models, Techniques, and Case. NY: McGraw-Hill, Inc.