INVESTIGATING TEAM COHESION IN COCOMO II.2000

by

Betty A. Snowdeal-Carden

EDWARD GOLDBERG, DM, Faculty Mentor and Chair

CLIFFORD BUTLER, DBA, Committee Member

LAWRENCE DOWDY, PhD, Committee Member

Susan Talley, EdD, Dean, School of Business and Technology

A Dissertation Presented in Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

Capella University

January 2013

Abstract

Software engineering is team oriented and intensely complex, relying on human collaboration and creativity more than any other engineering discipline. Poor software estimation is a problem that within the United States costs over a billion dollars per year. Effective measurement of team cohesion is foundationally important to gain accurate estimations of software development schedules, quality and cost. In light of recent team cohesion research, COCOMO II.2000 (Constructive Cost Model), a well-known industry software estimation tool, appears to contain a less than effective method in determining team cohesion. The COCOMO II.2000 TEAM variable represents the team cohesion measure of all stakeholders involved in the software development product life cycle and is one of five special factor variables aggregated into an exponent that represents risk to product development. An ineffective or inaccurate measure of team cohesion creates an exponential error in the software estimation results generated by the COCOMO II.2000. Using the Delphi method, a panel consisting of team cohesion, project management, and emotional intelligence experts have determined the COCOMO II.2000 TEAM variable does not effectively represent team cohesion as defined in this research.

Dedication

There are so many men and women who have given me encouragement, wisdom, and hope during this educational path, but three stand out more than others: Admiral Grace Murray Hopper, whose successful military and information technology careers provided guidance and encouragement; my husband, Steve, without whom none of this would have been possible; and, finally, my Lord Jesus, whose ultimate life example provided the hope needed in my darkest hours.

Acknowledgments

To those who assisted in making this dissertation possible: Dr. Edward Goldberg, a great mentor whose kindness and encouragement kept me focused and on track; to my committee members, Dr. Suzanne Kavli, Dr. Clifford Butler, and Dr. Larry Dowdy whose relationships provided great insight toward making this research a quality product; to Steve McConnell, author of *Software Estimation: Demystifying the Black Art,* who kindly provided charts for use in this dissertation; to Dr. Vanessa Druskat and Dr. Eduardo Salas who provided copies of illusive articles and words of encouragement; to Dr. Barry Boehm's assistant, Julie Sanchez, and Mr. Winsor-Brown for providing assistance with the COCOMO II.2000 model; to Carol Wellington who sent a copy of the GEQ manual and questionnaire for software engineers; to Dr. Barry Boehm, Dr. Randall Jensen, and Dr. Rapisarda for their wisdom and encouragement; to my husband, Steve, who provided unlimited hours of support and handsome smiles, and, finally, to the panel members whose whole-hearted participation made this study successful.

Table of Contents

List of Tables

List of Figures

CHAPTER 1. INTRODUCTION

Introduction to the Problem

Software development project management has been plagued with high failure rates over the past quarter century (Charette, 2005; Demarco, 1995, 1997; Demarco & Lister, 2003; Furton, 2003; Glass, 2005; Standish Group, 1999, 2001, 2002). Significant research has been accomplished toward determining the causes of high failure rates in software development project management, and in many cases, researchers have cited the project manager as the primary cause (Glass, 2005, 2006; Groth, 2004; Jorgensen & Molokken, 2006; Molokken & Jorgensen, 2003; Laird, 2006; Rubinstein, 2007; Standish Group, 1999, 2001; Xia & Lee, 2004).

To understand the significant affect project managers can have on the software on the software development process it is necessary to first briefly explore the responsibilities of this occupation. Project management is the process of assimilating tools and techniques that describe, organize, and monitor project activities. Project managers are responsible for managing the processes and applying the tools and techniques to complete the project (Heldman, 2004). In the realm of software development, project managers generally use software estimation tools to determine software development cost and schedules. As with any estimation tool, the ability for the software development estimation tool to accurately model the business processes of software development determines how well the tool can deliver an accurate estimation.

COCOMO II.2000 (Constructive Cost Model II; Boehm, Abts, Winsor-Brown, et al., 2000) is an "empirical, well documented, independent model" (Nasir, 2006, p. 2) that uses "formulae to predict cost (or effort) based on estimates of size and identification of specific attributes that enable calibration of data to fit the project" (Benediktsson & Dalcher, 2004, p. 5). COCOMO II.2000 has three sub-models, each one increasing in fidelity: (a) Applications Composition, (b) Early Design, and (c) Post-Architecture (Musilek, Pedrycz, Sun, & Succi, 2002). Of these three sub-models, the most defined is the Post-Architecture sub-model. COCOMO II.2000 estimates a range from best to worst outcomes regarding cost, effort and schedule (Oriogun, 2000). Within COCOMO II.2000 exist five Scale Factors (SF) that represent the "relative economies or diseconomies of scale" …[and] are a significant source of exponential variation on a project's effort or productivity" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 30-31).

TEAM, one of the five SFs (Table 1), is defined as the development team cooperation and cohesion scale driver which "accounts for the sources of project turbulence and entropy because of difficulties in synchronizing the project's stakeholders: users, customers, developers, maintainers, interfacers, [and] others" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 34). In short, TEAM, represents the team cohesion measure of the project stakeholders and henceforth is known as software development team.

Table 1. COCOMO II.2000 Scale Factors

Symbol	*Drivers*	*Name*
SF_1	PREC	Precendentedness
SF_2	FLEX	Development Flexibility
SF_3	RESL	Architecture and Risk Resolution
SF_4	TEAM	Team Cohesion
SF_5	PMAT	Process Maturity

Note. TEAM is the Scale Factor of interest for this study. From "Calibrating the COCOMO II.2000 Post-Architecture Model" by B. Clark, S. Chulani-Devnanzi, & B. Boehm, 1998. *Proceedings of the 1998 International Conference on Software Engineering,* pp. 477-480. Copyright 1998 by the Institute of Electrical and Electronics Engineers (IEEE). Adpated with permission.

Currently, the method used to determine the value of TEAM (Table 2) is the project manager's subjective answers to four simple questions: (a) what is the consistency of stakeholder objectives and cultures; (b) what is the ability and willingness of stakeholders to accommodate other stakeholders' objectives; (c) how much experience do the stakeholders have in operating as a team; (d) how much teambuilding exists to achieve shared vision and commitments (Boehm, Abts, Winsor-Brown, et al., 2000)?

Table 2. TEAM Rating Components

Characteristic	**Very Low**	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
Consistency of stakeholder objectives and cultures	Little	Some	Basic	Considerable	Strong	Full
Ability, willingness of stakeholders to accommodate other stakeholders' objectives	Little	Some	Basic	Considerable	Strong	Full
Experience of stakeholders in operating as a team	None	Little	Little	Basic	Considerable	Extensive

Characteristic	Very Low	Low	Nominal	High	Very High	Extra High
Stakeholder teambuilding to achieve shared vision and commitments	None	Little	Little	Basic	Considerable	Extensive

Note: From *Software Cost Estimation with COCOMO II.2000* (p. 36), by Boehm B., Abts C., Winsor-Brown, A., et al., 2000, Upper Saddle, NJ: Prentice Hall. Copyright 2000 by Prentice Hall PTR. Adapted with permission.

In light of recent team cohesion research, this research investigated the effectiveness of the COCOMO II.2000 TEAM method of measurement (Bollen & Hoyle, 1990; Carless & De Paola, 2000; Festinger, 1950; Knouse, 2007; Rainey & Schweickert, 1988; Rapisarda , 2002a, 2002b; Salas, Burke, Fowlkes, & Priest, 2004; Wellington, Briggs, & Girard, 2005). Additionally, this research investigates the possibility of using two other team cohesion methods of measure, the ECIv2.0 (Emotional Competence Inventory version 2.0) and the GEQ (Group Environment Questionnaire) modified for software engineer teams.

Background of the Study

Software development heavily depends upon the collaborative ability of the software development team, which is inclusive of all stakeholders. The key ingredient to any successful team is successful collaboration (Brooks, 1995; Druskat et al., 2006; Glass, 2003; Jones, 2006; Maxwell, 2001; McConnell, 1998, 2004, 2006). Team cohesion is defined as “a dynamic process, which is reflected in the tendency for a group to stick together and remain united in the pursuit of its goals and objectives” (Carron, 1982, p. 124). Software development is an “impressively complex socio-technical activity” where limited and/or strained interaction between team members weakens team cohesion (Sawyer, 2004, p. 95). Additionally, “ the centrality of [team] cohesion as a mediator of group formation, maintenance, and productivity has led some social scientists to deem it the most important small group variable (Golembiewski,1962; Lott & Lott, 1965)" (Bollen & Hoyel, 1990, p. 479).

In his article titled "Why Software Fails," Charette states "bad decisions by project managers are probably the single greatest cause of software failures today" (2005, p. 47). Additionally, Tiwana and Keil (2004) found the majority of the project risks identified was within the project manager's sphere of influence. The project manager is responsible for guiding executive management toward facilitating feasible projects, preparing realistic risk-assessed schedules, and gathering the best technical and human resources to form a productive and collaborative work environment while leading the software team toward the successful delivery of an on-time, on-target and on-budget software product (Druskat & Druskat, 2006; Jones, 2006). The software project manager is expected to perform these tasks while managing the project risks and successfully mitigating any unexpected circumstances that might thwart the successful goal of delivery (Boehm, 1991; Brooks, 1995; Furton, 2003; Glass, 2003; Hassan & Holt, 2003; Heldman, 2004; Humphrey, 2005; Jeffery & Scott, 2002; Maxwell, 2001; McConnell, 1998, 2004, 2006; Mills, 1999; Pressman, 2004; Saeki, 1995; Smith, Bohner, & McCrickard, 2005; Tomayko & Hazzan, 2004; Wallace & Kiel, 2004).

A high-risk project area is the unexpected loss or gain of one or more team members; specifically, estimating the amount of time to replace a team member, the time required for the replacement to become a productive member of the team, and how much the replacement changes the collaborative ability of the team (Tellioglu & Wagner, 1999). As with any unplanned event, the loss and gain of a team member generally creates stress among other team members. If a team member is stressed, withdrawn, or has conflict with another team member, the collaborative work of the team suffers, leading to poor quality and delayed deadlines (Brooks, 1995; Bushyacharu, 1996; Sikes, Gulbro, & Shonesy, 2010; Wolff, Druskat, Koman, & Messer, 2006). Key toward creating high team cohesion within a software development team is the necessity of creating a team with excellent technical skills and equivalent social skills (Sawyer, 2004). High team cohesion measures can lead toward more successful software development, that is, software developed on time, on target, and on schedule (Stevens, 1998).

Creating effective teams requires building social capital. Social capital is dependent upon "how a team deals with emotion at the individual, group and cross-boundary levels [and]…influences the development of, and engagement in, effective task processes, which influence team effectiveness" (Druskat & Wolff, 1999, p.6). In an article titled "Process or Behavior: Which is the Risk and Which is to be Managed," the author found "it is behavior and not the set of procedures, which is the risky factor" (Navare, 2003, p. 6). Further, a study observing the effects of disruptions on performance within software engineering teams determined teams who "suffered from communication problems proved to be the most dangerous in terms of encountering serious problems and failing to meet project milestones" (Karn & Cowling, 2005, p. 32).

One obvious facet of software development project failure is discipline immaturity (Pour, Griss, & Lutz, 2000; Thompson & Reed, 2005). The development of two very important volumes present a body of knowledge (BOK) toward maturation of software engineering and project management disciplines, specifically, the Project Management Institute's (PMI) 2004 *Project Management Body of Knowledge* (PMBOK) and the

Institute of Electrical and Electronics Engineers' (IEEE) 2004 *Software Engineering Body of Knowledge* (SWEBOK). Although these two volumes provide a foundation for establishing software development toward discipline maturity (Twaites & Sibilla, 2002), the ability to adequately model the collaboration level of a software development, which is directly linked to the production level of the team (Chai, 2003; Hoegl, Weinkauf, & Gemuenden, 2004; McConnell, 1998, 2006; Rifkin, 2001), needs further investigation. Each software project has its own unique risks that can lead to unexpected or unplanned events (Wallace & Keil, 2004, Chai, 2003; McConnell, 1998, 2006; Rifkin, 2001), however, if a project manager blindly uses an estimation tool that contains an inaccuracy the software development estimation will be inaccurate, thus leading the project toward failure. One such problem area associated with software development estimation yet to be investigated is the effectiveness of the COCOMO II.2000 TEAM variable.

Statement of the Problem

Effective measurement of team cohesion is foundationally important in gaining an accurate estimation of software development schedules and cost (Agarwal, Kumar, Yogesh, Bharadraj, Anantwar, 2001). In light of recent team cohesion research, COCOMO II.2000 (Boehm, Abts, Winsor-Brown, et al., 2000), a well-known industry software estimation tool, appears to have a less than effective method in determining team cohesion (Bollen & Hoyle, 1990; Carless & De Paola, 2000; Festinger, 1950; Knouse, 2007; Rainey & Schweickert, 1988; Rapisarda, 2002a, 2002b; Salas, Burke, Fowlkes, & Priest, 2004; Wellington, Briggs, & Girard, 2005). The COCOMO II.2000 TEAM variable is one of five special factor variables. The aggregate of the five special factor variables make up an exponential variable used within the COCOMO II.2000 model. An ineffective measure of team cohesion would present an exponential error in the software estimation results of the COCOMO II.2000 model.

Purpose of the Study

This study provides further evidence of how project managers are failing (Standish Group, 1999, 2001, 2002) to deliver software projects on time, target, and budget unbeknownst to them due to potential inaccuracies within a commonly used software estimation model. Specifically the purpose of this study is to investigate if the method used to determine the value of the COCOMO II.2000 TEAM variable effectively represents the team cohesion value and, therefore, effectively represents a true estimation of product completion. This study also investigates the effectiveness of two other team cohesion models, the GEQ and the ECIv2.0, as possible substitutes for the COCOMO II.2000 TEAM method. Further, this study investigated the effectiveness of the COCOMO II.2000 software estimation tool towards modeling the change of team cohesion (Wellington, Briggs, & Girard, 2005) over the life of the project.

Rationale

Recent research has determined the root causes of high software development project failure rates (Charette, 2005) were within the project manager's realm of influence (Tiwana & Keil, 2004). Software development project managers rely on software estimation tools to assist them in determining the feasibility and schedule of a software project (McConnell, 1998, 2004, 2006). COCOMO II.2000, a well-known and utilized software estimation tool (Benediktsson & Dalcher, 2004), contains a team cohesion variable that is determined by the project manager's subjective answers to four (Table 2) very simple questions (Boehm, Abts, Winsor-Brown, et al., 2000). Recent team cohesion research presents team cohesion models where all team members participate toward determining team cohesion (Rapisarda, 2002b; Wellington, Briggs, & Girard, 2005). Additionally, recent team cohesion research has determined that team cohesion changes over the lifetime of the project, another possible area that may not be correctly modeled within COCOMO II.2000 (Wellington, Briggs, & Gerard, 2005). The COCOMO II.2000 TEAM variable is one of five variables that when aggregated make up an exponential variable within the COCOMO II.2000 software estimation model. An error within the team cohesion measure of the COCOMO II.2000 software estimation model would cause an exponential error in the software estimation results. This research seeks to investigate the effectiveness of the team cohesion method used within COCOMO II.2000 and to determine if the COCOMO II.2000 effectively models the change of team cohesion throughout the life of the project.

Research Questions

This research used the Delphi Technique to gather data. Specifically, three rounds of e-mail were sent to a panel of experts to investigate the effectiveness of the COCOMO II.2000 method in determining team cohesion. The questions were designed to gain as much expertise from the panelists regarding the following: (a) the necessity for this study, (b) the effectiveness of the COCOMO II.2000 TEAM variable, (c) the effectiveness of the COCOMO II.2000 model in effectively modeling the change of team cohesion throughout the life of the software project, and (d) the effectiveness of using questions from the two other models: GEQ (Wellington, Briggs, & Girard, 2005) and the ECI.v20 (Rapisarda, 2002b). The first round of questions asked the panelists to determine the effectiveness of the COCOMO II.2000 methodology in measuring team cohesion. The second round of questions asked panelists their opinion of two other methodologies used to measure team cohesion. The third and final round asked the panelists for their expert opinion on any future research that should be accomplished to improve the team cohesion measurement within the COCOMO II.2000 model.

Significance of the Study

Software engineering estimation tools have improved the estimation process in advocating clear processes and establishing milestones toward measuring project success (Nasir, 2006). However, within COCOMO II.2000, the measurement of team cohesion, a human factor, and its effects on the development process appeared to be ineffectively measured. Specifically, the method used to measure team cohesion did not appear to effectively measure team cohesion nor did the model appear to account for the dynamic change of team cohesion over time (Wellington, Briggs, & Girard, 2005). According to McDonald and Edwards (2007) the development, interaction and personality influences of software engineering teams has been a concern from the 1960s to the present day (Gorla & Wah Lam, 2004) and "despite claims from leading figures in the field that it is fundamentally people that make the difference between software success and failure, a corpus of knowledge and good practice has failed to emerge"(McDonald & Edwards, 2007, p. 67).

Clearly, in light of previous research (Druskat & Druskat, 2006; Feyerherm & Rice, 2002; Hogan & Thomas, 2005; Peslak, 2005, 2006; Smith, Bohner, & McCrickard, 2005; Wellington, Briggs, & Girard, 2005), investigating the method that determines the value of the COCOMO II.2000 TEAM variable did bring to light how the project manager may be failing in delivering software projects on time, on target, and on budget, specifically by the use of an ineffective estimation tool. The investigation of two other team cohesion methods provided one potential replacement method for the COCOMO II.2000 TEAM method.

What can be taken away from this study is a refined knowledge of the effect team cohesion has on software development process, how the subjective opinion of the project manager is not the best method to measure team cohesion, and the discovery of other areas in software development where team cohesion measurement can be of use. Additionally, this investigation suggests further research toward validating the findings and investigating the effectiveness of software estimation tools in measuring today's software development processes as many of the current estimation tools were designed twenty years ago (Agarwal et al., 2001; Boehm, 2006; Boehm, Abts, & Chulani, 2000; Laird, 2006; Lum, Hihn, & Menzies (2006); Nasir, 2006).

Discussion of Terms

In order to provide the reader with as much understanding of the purpose for this research, the following terms have been ordered in such a way as to build upon one another. The three major areas linked and addressed in this research are software project management, team cohesion and emotional intelligence. Each area has aspects that need clarification; that is, in order to truly understand the chaos involved in managing a software project (Hassan & Holt, 2003), one must understand the foundational aspects of software development, software development methodologies, etc. Additionally, it is important the reader have at least a preliminary understanding of team cohesion, task cohesion, emotional intelligence, and group intelligence in order to fully grasp the depth of research represented. The following definitions should provide the clarity needed to fully understand the importance of this research.

Software Engineering

The term *software engineering* had its birth at the 1968 NATO conference (Poore, 2004). Software Engineering is defined by the Institute of Electrical and Electronics Engineers (IEEE) Computer Society's Software Engineering Body of Knowledge (SWEBOK) as "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software" (SWEBOK, 2004, p 1-1). Although software development and software engineering are frequently used interchangeably software development is only a part of the software engineering process. In an article titled "The Art, Science, and Engineering of Software Development," McConnell stated what makes software engineering so different from other engineering is that "software is so labor intensive that a significant amount of engineering energy must be focused on *project* goals in addition to *product* goals—on the means to the ends as well as the ends themselves" (McConnell, 1998, p. 119). In his book *Professional Software Development* McConnell also states "Brooks argues…that software is difficult because of its essential complexity, conformity, changeability, and invisibility" (Brooks as cited in McConnell, 2004, p. 38). Unlike architectural engineers or bioengineers where there is a single focus and a visible product, software engineers create tools for all industries and sciences. Therefore, software engineers must learn the science and/or business rules of the system that is to be modeled, simulated, designed and developed.

Software Development

Software is ubiquitous throughout day-to-day activities from the washing machine to the space shuttle, the automobile to the heart monitor, and the financial institutions to the power grid; software controls much of everything in today's world and software engineers were the ones who created these invisible applications that provide the tools needed to keep much of the world running. According to the IEEE Standard Glossary of Software Engineering Terminology (Std 610.12-1999) *software development* is the process where customer software requirements are implemented into a customized software product. Specifically, where "user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use" (IEEE, 1990, p. 67). Software developers or engineers, who are often referred to as programmers, design and create a software application. Programmers use one or more computer programming languages to create a software application. The environmental scope of the software application aids in determining the best computer language(s) and development model(s) to use in architecting and developing the resulting software product.

Software Development Life Cycle

The software development life cycle begins when the project manager estimates the feasibility of a software development opportunity. When a software development opportunity is determined feasible, and then successfully submitted for bid, and then awarded to the software company; then, and only then, the software development team is assigned the task to develop the software product. The assignment of the software development team initiates the requirements phase. The first iteration of the software development life cycle is terminated when the software product is successfully and satisfactorily delivered to the customer. The product now enters the maintenance phase. The software development life cycle is generally made up of five fundamental phases (a) requirements, (b) design, (c) implementation, (d) testing, and, (e) maintenance.

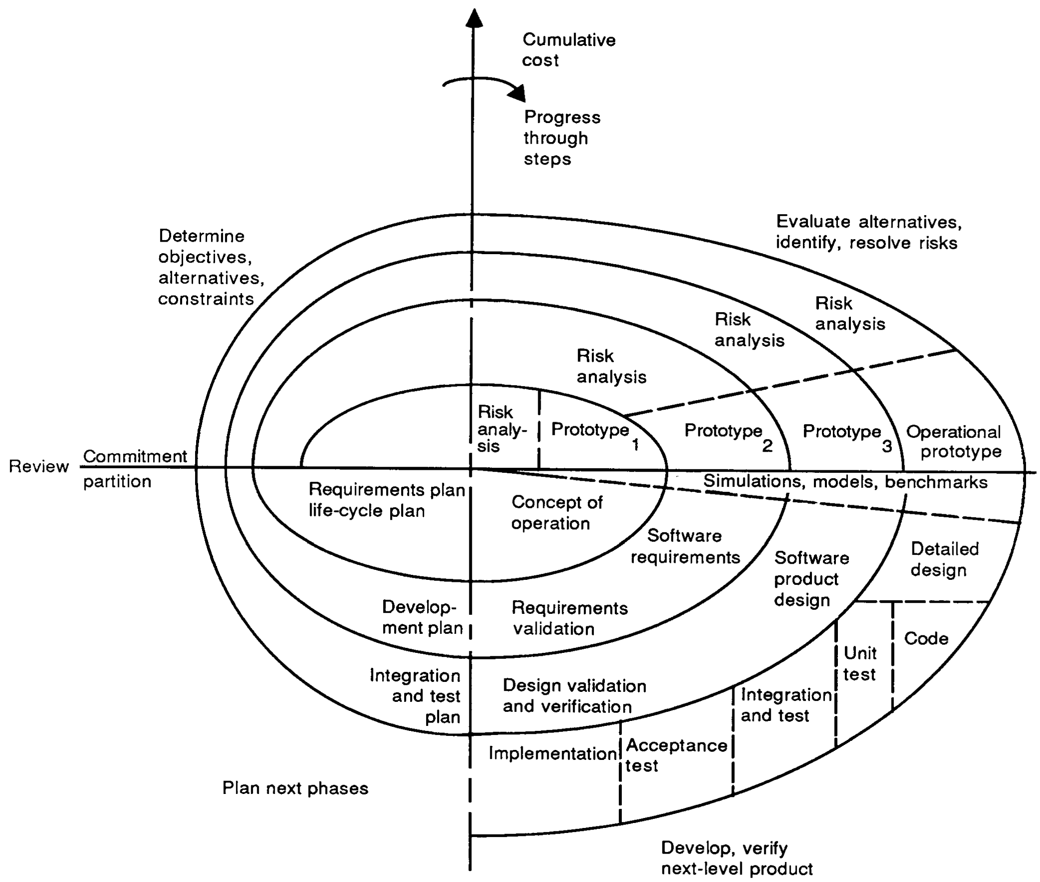

Figure 1. Boehm's Spiral Model for software development life cycle. From "A Spiral Model of Software Development and Enhancement," by B. W. Boehm, 1988, *IEEE Computer*, *(21)*5, p.64. Copyright 1988 by the Institute of Electrical and Electronics Engineers (IEEE), Inc. Reprinted with permission.

Software Development Life Cycle Phases

There are generally five phases within the software development life cycle: (a) the requirements phase, (b) the design phase, (c) the implementation phase, (d) the test phase, and the (e) maintenance phase (IEEE, 1990, p. 67).

The spiral model (Figure 1) is a common model used in software development projects and provides a cyclical picture of how the project evolves throughout the development life cycle. If the software product is determined to be feasible the project continues to the next phase, the requirements phase, otherwise, the project ends.

The requirements phase begins with the software development team thoroughly investigating the deliverables of the project. During the requirements elicitation, the development team analyzes each requirement for completeness of comprehension to ensure the end product is exactly what the customer is expecting. The requirements phase is completed when the customer and development team are in agreement of the specific deliverables required at the end of the project and is generally written in a document titled the Software Requirements Specification otherwise known as the SRS. Once the SRS is agreed upon by both parties the project manager begins scheduling the requirements to the developers. The developers use the requirements to begin the desgin phase.

The design phase begins when each developer is asked by the project manager to to submit their educated guess as to how long it will take them to complete their assigned requirements. During this time the developer becomes much more familiar with the assigned requirements and customer expectations so as to provide a quality estimation of completion to the project manager. The estimation is to include the number of man hours to (a) design, (b) implement, and (c) test the completed section.

The design phase can contain one or more of the following: (a) architectural design, (b) interface design and/or (c) detailed design. The selection of design phases depend on whether the product is to be built from scratch or if the development task is to update an existing software product. The architectural design is concerned with the development of the structure of the product; that is, how it will interact with the hardware and the foundational software functionality that will assist the higher level software of the product perform its tasks. The interface design is primarily concerned with how the user or another software product will interact with the product to be developed. The detailed design is concerned with all the functionality between the architectural design and the interface design; that is, all the algorithms that interact between the interface and the architecture of the software product (Gustafson, 2002).

Some of the potential deliverables within the initial stage of the design phase are a Software Design Document (SDD) which could contain use case and object models describing the proposed fucntionality of the product and/or an Interface Design Document (IDD) which contains models describing the architecture and the interface interaction and a data dictionary describing the data and the database architecture. Generally, software developers do not begin writing implementation code until these documents are agreed upon by the customer. After the customer and software development team are in agreement regarding the proposed design the software developers begin implementing the design.

The implementation phase is when the software developers finally begin writing software lines of code (SLOC) or function points (FP) toward completing the software design. During the implementation phase peer reviews are generally scheduled on a weekly basis. During a peer review, team members review randomly selected sections of one another's code looking for exceptionally and/or weakly developed code areas. The goal of the peer review is to ensure a developer thoroughly understands the design task and ultimately ensures a quality product is delivered. Peer reviews are preliminary testing processes that project the risk assessment of the implementation phase. During the implementation phase software developers generally work alone and can encounter unexpected situations that were not considered in the estimation process which could significantly delay the delivery of the product. A situation could further be compounded if other components of the product can not continue until the troubled component is completed. An estimation is an educated best guess and is most often optimistic. More than any other phase within the software development lifecycle, the implementation phase needs to be closely monitored by the project manager to ensure the product remains on schedule. The deliverable for this phase is the application itself. The implementation phase ends when the implemented design is ready for testing.

The testing phase generally follows seven types of testing in order. They are unit testing, integration testing, system testing, alpha testing, beta testing, acceptance testing, and, regression testing. Unit testing is accomplished by the developer who generally records this information in a document called unit testing and is provided as a deliverable of the test phase and handed off to the next increment of the test phase. Integration testing and system testing are usually done by a group of software engineers exclusive of the developer(s) and whose sole job is to test the developed software products for quality and accuracy. A document is usually generated from this phase and is attached to the unit testing documentation. The next test increment is the alpha testing where the customer's test team tests the product at the developer's site. All issues are captured in a defect document and are sent back to the development team for adjustment to the software as specified and in accordance with the initial requirements. The beta testing is accomplished by the customer's testing team at the customer's site. During beta testing all issues are captured in a defect document and sent back to the development team to adjust the software as specified and in accordance with the initial requirements. The next phase of testing is acceptance testing where the software product is tested against the initial requirements and is tested by a separate quality assurance team usually designated by the customer. The document followed for this process is called the Acceptance Testing Plan (ATP) or Software Quality Assurance plan (SQA plan). Again, all issues are captured and returned to the development team to adjust the software as specified and in accordance with the initial requirements. The last phase is regression testing. This phase only need take place if the development product was added to a previous product or inserted into a multiple product environment. The regression testing ensures the new product has not interfered with previous established capablilites (Gustafson, 2002). Once the customer has agreed the product meets all the intial requirements the maintenance phase of the software development life cycle begins.

The maintenance phase generally deals with product anomolies and enhancements. That is "a large part of maintenance deals with accomodating new or changed user requirements and adapting software to a changed environment. It is about evolution, rather than just maintenance" (Van Vliet, 2008, p.466).

Project Management

Within the Project Management Body of Knowledge (PMBOK) project management is defined as "the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements" (Project Management Institute, 2004, p6). A Project Management Professional (PMP) certification study guide describes project management as the process of bringing together a "set of tools and techniques—performed by people—to describe, organize, and monitor the work of project activities. Project managers are...responsible for managing the project processes and applying the tools and techniques used to carry out the project activities" (Heldman, 2004, p. 6).

Software Project Management

The *Computer Science Handbook (*Pressman, 2004) states software project management begins when the software project manager has chosen a specific development strategy to model the software development process. For example, the most common software development models are the (a) incremental model and (b) the spiral model. The incremental model "applies linear sequences in a staggered fashion as calendar time progresses each linear sequence produces a deliverable [or] increment of the software [McDermid and Rook, 1993]" (Pressman, 2004, p.108.6). The spiral model develops software "in a series of incremental releases [where] increasingly complete versions of the engineered system are produced" (Pressman, 2004, p.108.7). Furthermore, the *Computer Science Handbook* (Pressman, 2004) describes software project management as encompassing the following activities: "measurement, project estimating, risk analysis, scheduling, tracking, and control" (p. 108.14) while maintaining focus on the people, problem and process of the software development process. "The manager who forgets that software engineering work is an instensely human endeavor will never have success in project management" (Pressman, 2004, p. 108.11).

Software Estimation

A software development company uses software estimation tools to determine feasibility (Boehm, 2006) and to bid on software development opportunites (Nasir, 2006). As in any business the company with the best reputation and estimate generally gains the businesss opportunity. "Pre-bid estimation is paramount in getting business for the company. Accuracy of pre-bid estimation governs the smooth running and success of a project" (Nasir, 2006, p. 305). Software estimation generally takes place throughout the software development life cycle and is most assuredly used within the feasibility phase to determine if a software development company can develop the software product at a profit.

Generally speaking, a software measurement policy guides the software estimation process. The goal of software estimation (Figure 2) is to properly and accurately estimate the resources and time needed to complete the project in order to accurately determine the cost of the endeavor.

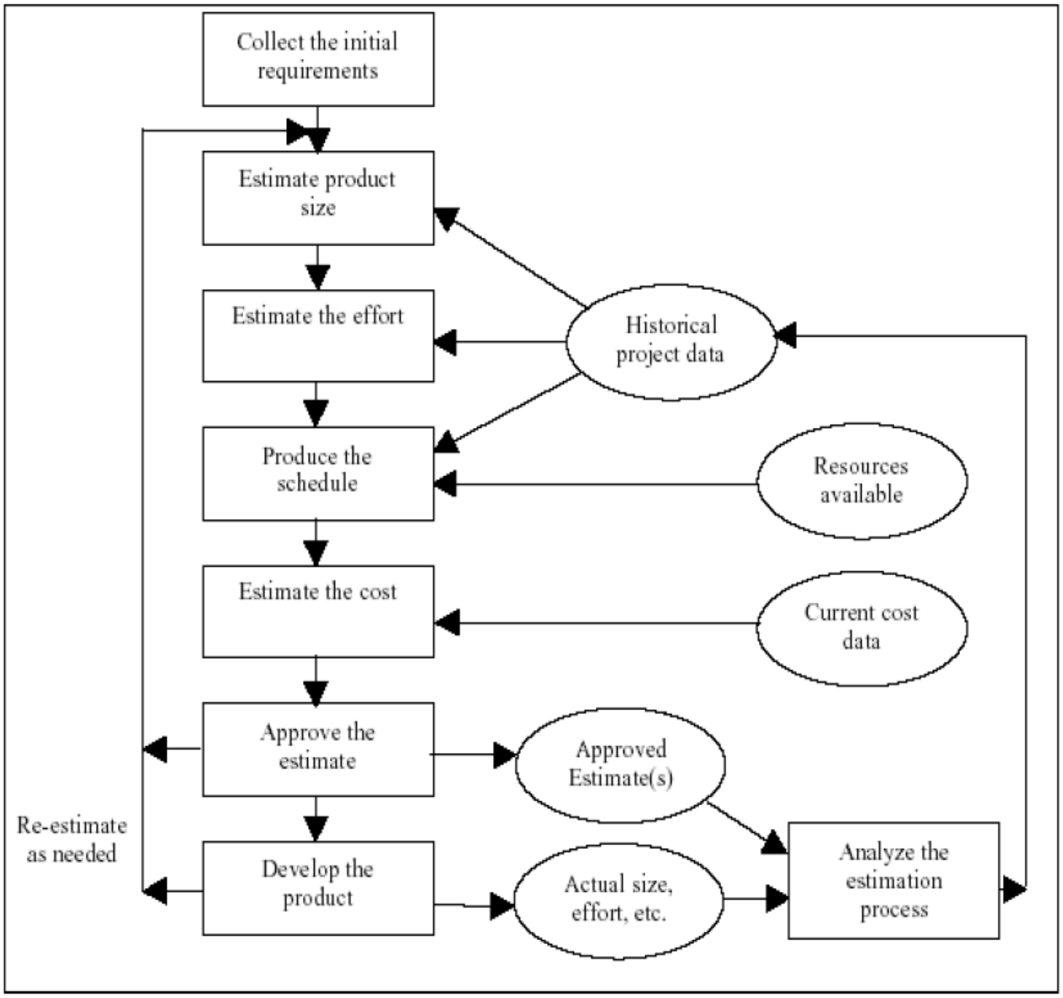

Figure 2. Basic Project Estimation Process. From "A Survey of Software Estimation Techniques and Project Planning Practices," by Nasir, M., 2006, *IEEE Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing* (SNPD'06). Copyright 2006 by The Institute of Electrical and Electronics Engineers (IEEE), Inc. Reprinted with permission.

The primary purpose of software estimation is "not to predict a project's outcome [but] to determine whether a project's targets are realistic enough to allow the project to be controlled to meet them" (McConnell, 2006, p. 13). Of key importance toward successful project estimation is historical data and/or metrics, that is, measurements of how the company performed in developing past software projects. However, if historical data is unavailable or inaccurate, the estimation will be inaccurate (Nasir, 2006). According to the Standish Group in 2004 "54% of projects were delivered late, 18% failed outright, and 28% were delivered on time and within budget" (McConnell, 2006, p. 24). Within McConnell's book *Software Estimation: Demystifying the Black Art* (2005), there are four generic sources of estimation errors: inaccurate information about (a) the project, (b) the software development organization's capabilities, (c) inaccurate estimation processes, and d) an extradordinary chaotic project environment.

There are many tools an organization can use toward software estimation and these tools generally use one of three methodologies toward software estimation: (a) the analogy method, (b) the top down method, and the (c) bottom up method (Nasir, 2006). The analogy method uses historical data of past projects to determine a software estimation. The top down method is "concerned with the overall characteristics of the system to be developed [and] takes into account integration, configuration management, documentation costs" (Nasir, 2006, p. 2). The bottom up method requires detailed requirements in order to estimate each component. The summation of each component's estimation determines the overall estimation (Nasir, 2006). According to McConnell (2006) size is the only factor that needs to be estimated. "Effort is computed from the size estimate by using historical productivity data. Schedule, cost, and features are computed from the effort estimate" (McConnell, 2006, p. 173). The software estimation tool used within this research is COCOMO II.2000 (Boehm, Abts, Winsor-Brown, et al., 2000) described in detail in the next section.

Constructive Cost Model II.2000 (COCOMO II.2000)

COCOMO II.2000 (Boehm, Abts, Winsor-Brown, et al., 2000) is an "empirical, well documented, independent model" (Nasir, 2006, p. 2) that uses "formulae to predict cost (or effort) based on estimates of size and identification of specific attributes that enable calibration of data to fit the project" (Benediktsson & Dalcher, 2004, p. 5). COCOMO II.2000 has three sub-models, each one increasing in fidelity, (a) Applications Composition, (b) Early Design, and (c) Post-Architecture (Musilek, Pedrycz, Sun, & Succi, 2002).

COCOMO II.2000 should be used for making financial decisions for software development projects, delineating project budgets and project schedules, to provide architectural discovery information toward determining the best fit for the time frame alloted, to assist in determining what portions of the project can be developed and what portions of the project can be reused or purchased, and to assist in creating investment strategies toward improving software capability via reuse, process maturity, outsourcing, etc. (Boehm, Abts, Winsor-Brown, et al., 2000).

The cost of the early prototyping efforts used to determine if the project can be accomplished or is feasible, including gathering initial requirements, is best determined by the Applications Composition model. The cost of the early design stage of software development which includes refining initial requirements toward determining the Function Point (FP), Use Case Point (UCP), or Software Lines of Code (SLOC) estimated to complete the project is best estimated using the Early Design model (Boehm, Abts, Winsor-Brown, et al., 2000). And finally, the most defined model, the Post-Architecture sub-model provides a range of cost, effort, and schedule estimates from best to worst case outcomes and gives the planner the ability to easily adjust the estimate to create what if scenarios and to show the effect that "adjusting requirements, resources and staffing might have on estimated costs and schedule e.g. for risk management or job bidding purposes" (Oriogun, 2000, p. 57).

COCOMO II.2000 accepts SLOC, FP and UCPs as its basic software size input (Nasir, 2006, p. 2). SLOC is one of the oldest units of measure in the software industry and is not a very accurate measure as it is highly dependent upon the language chosen to develop the software product (McConnell, 2006; Zuse, 1998). FP is a unit of measure based on function size and UCP is a unit of measure based on the number of use cases or scenarios the product needs to fulfill. COCOMO II.2000 is one of the most "widely utilized and accepted estimation methods in software engineering projects" (Benediktsson & Dalcher, 2004, p. 5). As an algorithmic tool it uses "formulae to predict cost (or effort) based on estimates of size and identification of specific attributes that enable calibration of data to fit the project [and] is expected to increase exponentially, rather than linearly, with project size" (Benediktsson & Dalcher, 2004, p. 5).

According to Boehm, Abts, Winsor-Brown, et al., the primary objectives of COCOMO II.2000 are

1. Provide accurate cost and schedule estimates for both current and likely future software projects
2. Enable organizations to easily recalibrate, tailor, or extend COCOMO II.2000 to better fit their unique situations
3. Provide careful, easy-to-understand definitions of the model's inputs, outputs, and assumptions
4. Provide a constructive model
5. Provide a normative model (2000, p. 3).

Intelligence Quotient (IQ) and General Intelligence (*g*)

In order to thoroughly understand the history of emotional intelligence a good understanding of the history and purpose of IQ is necessary. Charles Darwin's cousin Francis Galton who authored Hereditary Genius and founded the study of eugenics, that is, the study of human husbandry, introduced the study of human intelligence. Requiring an instrument to assist in determining the smartest human breeding specimens, Galton developed a testing booth called the *Anthropometric Laboratory* which contained seventeen devices to test physical abilities. "Galton thought the way to discern who was talented and who was not was to devise tests that measured people's physical energy, reaction times, and sensory acuity" (Murdoch, IQ: A Smart History of a Failed Idea, Kindle edition, Locations 268-73). In order to prove the measures of the *Anthropometric Laboratory* correlated with success in life, Galton developed the correlation coefficient that ironically proved the physiological tests of the *Anthropometric Laboratory* were not correlated with intelligence (Murdoch).

Furthering the investigation of determining how to measure intelligence, Charles Spearman, a former British Army officer, observed positive correlations between groups taking mental tests "that attempted to measure intelligence by measuring a person's ability to reason, draw analogies, and identify patterns" (Herrnstein & Murray, 1996, p. 2). Whenever a group did well on one mental test that same group would do well on a different mental test; additionally, if a group did poorly on one mental test they would do poorly on a different mental test; "As long as the tests involved cognitive skills of one sort or another, the positive correlations appeared" (Herrnstein & Murray, 1996, p. 3).

These observations provided evidence for a "unitary mental factor, which [Spearman] named *g*, for 'general intelligence'" (Herrnstein & Murray, 1996, p. 3).

James Flynn, the author of the Flynn Effect, which is the "systematic and pervasive rise in IQ scores [causing] IQ test norms to become obsolete over time (Flynn, 1984, 1987, 1998)" (Kanaya, Scullin, & Ceci, 2003, p. 778), stated g is best compared to the atom. What hold the components of intelligence together are g and what "acts as an atom smasher is cognitive trends measured over time" (Flynn, 2007, p. 25).

Furthermore, Gottfredson stated g can be extracted from the IQ test using factor analysis. "The ability to isolate *g* has revolutionized research on general intelligence, because it has allowed investigators to show that the predictive value of mental tests derives almost entirely from this global factor [*g*] rather than from the more specific aptitudes measured by intelligence tests" (Gottfredson, 1998, p. 25). Gottfredson further stated that g "describes mental aptitude rather than accumulated knowledge... [Additionally,] the predictive value of mental tests in the work arena stems almost entirely from their measurement of g, and that value rises with the complexity and prestige level of the job" (Gottfredson, 1998, pp. 25 – 28).

The American Psychological Association (APA) defines *intelligence* as the "general ability to think, reason, learn, apply knowledge, or deal effectively with the environment" and defines the Intelligence Quotient (IQ) as "relative intelligence of an individual expressed as a score on a standardized test of intelligence" (Tuleya, 2007, pp. 151-2). Historically, the intelligence test was introduced to America as a solution to what was considered a very serious issue at the time. Physicians did not have a scientific means to accurately define, label and measure the different levels of mentally challenged people. Additionally, society needed a means to determine who could be labeled mentally inferior in order to place these socially unacceptable individuals in an institution for the feeble-minded. Society associated the mentally inferior with the criminal ranks, considering them a safety problem and a costly burden. Furthermore, educators needed a cost-effective means to accurately assess the knowledge level of the flood of new students entering the schools due to the mandatory education law enacted early in the twentieth century. Physicians, society, and educators all found their solution in the Binet-Simon intelligence test introduced by Goddard, the first intelligence test to be given in America (Murdoch, 2007).

A colleague of Goddard's, Lewis Terman, introduced IQ to America when he presented a "more sophisticated revision of Binet's test" called the Stanford-Binet test. The Stanford-Binet test included the "now-famous Intelligence Quotient, which was based on a German psychologist's idea of dividing the subject's mental age by his chronological age" (Murdoch, 2007, Chapter 6, para. 26).   The German psychologist was William Stern and the IQ equation is as follows: IQ = (mental age /chronological age) x 100. Therefore, an 8-year-old child who answers questions typical of a 10-year-old child would have an IQ of 125. Furthermore, in a bestselling psychology textbook, Myers stated, "Most current intelligence tests, including the Stanford-Binet, no longer compute an IQ. …Although there is no longer any intelligence quotient, the term 'IQ' still lingers in everyday vocabulary as a shorthand expression for 'intelligence test score'" (Myers, 2004, p. 316).

Stephen Murdoch, author of *IQ: A Smart History of a Failed Idea* argued that the historical uses of IQ tests were appalling and vile and that it is the power of the IQ test to "describe individuals' abilities is very rough, so when they are used in education, employment, or elsewhere, the tests incorrectly predict many peoples' future behavior" (Murdoch, Afterword, para. 5). In 1994, Herrnstein and Murray presented *The Bell Curve,* a very controversial book about IQ and class that describes "IQ's relationship to crime, unemployment, welfare, child neglect, poverty, and illegitimacy; ethnic difference in intelligences; trends in fertility among women of different levels of intelligence; and what policy can do—and cannot do—to compensate for differences in intelligence" (Hernstein & Murray, 1996, back cover).

Recognizing the need to address the controversial debate of Herrnstein and Murray's book *The Bell Curve*, the Board of Scientific Affairs (BSA) of the APA "concluded that there was an urgent need for an authoritative report on these issues—one that all sides could use as a basis for discussion" (Neisser, Boodo, Bouchard, Boykin, Bordy, Ceci, … & Urbina, 1996, p.77). In a 1996 American Psychologist article titled "Intelligence: Knowns and Unknowns," the APA's task force addressed the controversy surrounding the concerns of Herrnstein and Murray's book *The Bell Curve* and questions in general about IQ.

> Because there are many ways to be intelligent, there are also many conceptualizations of intelligence. The most influential approach, and the one that has generated the most systematic research, is based on psychometric testing…Psychometricians have successfully measured a wide range of abilities, distinct from one another and yet intercorrelated [*g*]…Standardized intelligence test scores ("IQs"), which reflect a person's standing in relation to his or her age cohort, are based on tests that tap a number of different abilities. (Neisser, et al., 1996, pp. 77-97)

Emotional Intelligence (EI)

The history and foundation of Emotional Intelligence (EI) began in the 1920s when E. L. Thorndike (1920) suggested another form of intelligence, other than cognitive intelligence, existed and for which he termed *social intelligence.* Thorndike defined social intelligence as "the ability to understand men and women, boys and girls-to act wisely in human relations" (1920, p. 228). In the most recent edition of the American Psychological Association's (APA) *Thesaurus of Psychological Index Terms,* Emotional Intelligence (EI) is defined as the "ability to monitor and appraise one's own and other's feelings and emotions and to use this information to guide thinking and action" (Tuleya, 2007, p. 98). Between the time of Thorndike's suggestion of social intelligence and the APA's definition of emotional intelligence, Goleman introduced EI to the masses with his bestselling book *Emotional Intelligence: Why It Can Matter More Than IQ* (1995). The title of this book was apparently determined by an article Goleman came upon during his research, an article by Mayer and Salovey, where the initial definition for the EI construct was described "in terms of an individual's ability to perceive emotion in self and others, to understand emotion, and then to manage emotion in self and others" (Ashkanasy & Daus, 2005, p. 26). Goleman's book presented three areas of concern challenging further research in the area of EI:

1. Emotional Intelligence is a psychological construct relatively independent from traditional IQ and personality.
2. Emotional Intelligence is important because it can account for additional variance in outcomes, including academic and vocational outcomes, when IQ and personality variables are controlled.
3. Unlike traditional IQ and personality constructs, emotional intelligence can be meaningfully developed in individuals (Emmerling, 2008, p.70).

Soon after the publication of Goleman's book (1995), Bar-On "quickly recognized the potential of the measure he had developed for his dissertation work, and rebadged his scales as the EQ-I [Emotional Quotient Inventory], a multidimensional questionnaire measure of emotional intelligence" (Ashkanasy & Daus, 2005, p. 26). EQ is an acronym for emotional quotient similar to the acronym IQ that is defined as Intelligence Quotient. Bar-On defines EQ as the metrical unit of EI; therefore, EQ and EI are used synonymously within many EI articles.

Team Cohesion

Initially, team or group cohesion was historically defined as a unitary construct similar to Festinger's conceptualization of team cohesion (Carless & De Paola, 2000). Festinger began examining "…individual forces hypothesized to influence members to remain a part of the group" (Bollen & Hoyle, 1990, p. 481). However, recognizing the difficulty in ascertaining which forces were most important or should be measured as a conglomerate, Festinger "proposed that cohesion be reconceptualized as 'the resultant of all forces' that influence members to remain in the group" differing from his initial definition which focused on "the causes of cohesion (forces acting upon group members); the revised definition proposed by Festinger (1950) focused on the effects of cohesion (the resultant forces)" (Bollen & Hoyle, 1990, p.481).

Furthering the research of conceptualizing cohesion as an independent construct, Bollen and Hoyle, "recognizing that there exists no true definition of cohesion" proposed a theoretical definition of "specifying elements of a group member's perception of his or her group membership that might reflect a tendency to cohere" and called it perceived cohesion (1990, p. 482). The formal definition of *perceived cohesion* is that it "encompasses an individual's sense of belonging to a particular group and his or her feelings of morale associated with membership in the group" (Bollen & Hoyle, 1990, p. 482).

Additionally, Carless and DePaola determined that task cohesion, compared to social cohesion and individual attraction to the group, "was the better predictor of work-group performance" (Carless & DePaola, 2000, p. 85). Furthermore, Knouse argues that cohesion develops as a result of the task at hand and not social interaction among team members" (Knouse, 2007, p51). However, *The Comprehensive Handbook of Psychological Assessment: Vol 4. Industrial and Organizational Assessment* describes team cohesion as having two subcategories: (a) socially based and (b) task based. Where *socially based cohesion* "refers to a positive attitude toward developing and maintaining interpersonal relationships within a team" and *task based cohesion* refers to a "task oriented belief about achieving the team's goals through commitment to the team approach (Rainey & Schweickert, 1988)" (Salas, Burke, Fowlkes, & Priest, 2004, p. 438).

Finally, two team cohesion models are used within this research, the ECIv2.0 (Emotional Competence Inventory) and the GEQ (Group Environment Questionnaire). Using the ECI, Rapisarda (2002b) associated emotional intelligence competencies with team cohesion and defined the concepts of *team cohesion* as follows:

> a team is defined as 'individuals who see themselves and who are seen by others as a social entity, who are interdependent because of the tasks they perform as members of a group, who are embedded in one or more larger social systems (e.g., community, organization), and who perform tasks that affect others (such as customers or coworkers)' (Guzzo & Dickson, 1996, p.308). Performance is defined as the team reaching 'profitability, cycle times, volume,' efficiency, goal attainment (Goleman, 1998, p.301). Cohesiveness is defined as 'the forces acting on members to remain in the group. These forces may depend on the attractiveness or unattractiveness of either the prestige of the group, members in

> the group, or the activities in which the group engages' (Festinger, 1950, p.274). (Rapisarda, 2002a, p. 366)

Rapisarda concluded nine emotional intelligence competencies are correlated with team cohesion. The questions associated with these competencies were investigated as an effective replacement for the COCOMO II.2000 TEAM method.

Wellington, Briggs, and Girard (2005) used a modified Group Environment Questionnaire (GEQ) for software engineers in order to measure software teams' team cohesion. They defined the concepts of team cohesion as follows:

> First, a 'team' is not just a group of people. It is important that a team is a group of people that share an identity and a purpose... 'Cohesion' is the degree to which the team sticks together as they pursue the team's purpose. Clearly, software engineering teams meet our criteria and this definition of cohesion has value because it is the degree to which they work together and should reflect some aspect of their ability to succeed. (Wellington, Briggs, & Girard, 2005, p. 1)

This study investigated the effectiveness of the team cohesion correlated questions of the ECIv2.0 and GEQ models to improve the team cohesion measure of the COCOMO II.2000 model, therefore, it is important to understand the definitions behind both models.

The Emotional Competence Inventory Version 2.0 (ECIv2.0)

The Emotional Competence Inventory 2.0 (ECIv2.0) is described as a 360-degree tool "designed to assess the emotional competence of individuals and organizations" (Wolff, 2005, p. 2). A 360-degree tool requires "people who work or live with a person to assess the frequency of their use of various behaviors. Often those asked include their Boss, Peers and Subordinates at work" (Boyatzis & Sala, 2004, p. 155). Dr. Goleman's book *Working With Emotional Intelligence* (2000) and Dr. Richard Boyatizis's *Self-Assessment Questionnaire* (SAQ), as well as the competencies defined in the Hay/McBer's *Generic Competency Dictionary* (1996), provides the foundational work of this instrument. The ECIv2.0 Technical Manual (Wolff, 2005) defines EI as “the capacity for recognizing our own feelings and those of others, for motivating ourselves and for managing emotions effectively in ourselves and others” (Wolff, 2005, p.2). Additionally, the manual defines an emotional competence as a “learned capacity based on emotional intelligence that contributes to effective performance at work” (Wolff, 2005, p.2).

The ECIv2.0 measures 18 competencies grouped into four clusters: (a) Self-Awareness, (b) Self-Management, (c) Social Awareness, and (d) Relationship Management; these clusters and their associated competencies are listed in Table 3.

Table 3. Norms and descriptive statistics for ECIv2.0

ECIv2.0 Cluster	Competency	Average-Item				Scored			
		Self (n=19997)		Total Others (n=21156)		Self (n=6372)		Total Others (n=6433)	
		Mean	D	Mean	D	Mean	D	Mean	D
Self-Awareness	Emotional Self-Awareness	4.10	0.56	3.07	0.70	3.03	0.58	2.85	0.41
	Accurate Self-Assessment	3.87	0.47	3.57	0.49	2.88	0.52	2.88	0.45
	Self-Confidence	4.01	0.60	4.15	0.41	3.57	0.45	3.71	0.31
Self-Management	Emotional Self Control	3.60	0.56	3.73	0.47	2.79	0.59	3.01	0.46
	Transparency	4.11	0.51	3.49	0.52	3.17	0.50	3.05	0.38
	Adaptability	3.96	0.49	3.69	0.43	3.65	0.38	3.69	0.27
	Achievement	4.04	0.49	3.69	0.50	3.02	0.51	2.99	0.37
	Initiative	3.68	0.52	3.27	0.48	3.61	0.40	3.64	0.30
	Optimism	4.24	0.49	3.93	0.46	3.27	0.48	3.23	0.37
Social Awareness	Empathy	4.13	0.50	3.89	0.46	3.74	0.35	3.67	0.33
	Organizational Awareness	3.90	0.62	3.62	0.56	2.94	0.57	3.04	0.39
	Service Orientation	4.29	0.50	3.98	0.55	3.74	0.36	3.75	0.28
Relationship Management	Developing Others	4.01	0.55	3.62	0.58	3.00	0.59	2.97	0.45
	Inspirational Leadership	3.76	0.66	3.66	0.58	2.81	0.61	2.91	0.48
	Change Catalyst	3.83	0.56	3.57	0.51	2.84	0.58	2.82	0.40
	Influence	3.80	0.59	3.50	0.54	2.89	0.58	2.97	0.40
	Conflict Management	3.34	0.57	2.92	0.51	3.62	0.39	3.67	0.26
	Teamwork & Collaboration	4.12	0.47	3.94	0.43	3.11	0.52	3.10	0.40

Note: *Emotional Competence Inventory (ECI) Technical Manual* (p. 41) Prepared by Steven Wolff, 2005, Boston: Hay Group McClelland Center for Research and Innovation. Copyright by Hay Group.

Each EI competency can be exhibited at one of four levels. Any particular question in the ECI represents one level of one competency, thus there are 72 questions (18 competencies times 4 levels)" (Wolff, 2005, p. 6).

Although the ECIv2.0 Technical Manual (Wolff, 2005) presents several studies conducted to determine various forms of validity for the ECI, this research reports on the most recent research completed by Byrne (2003). Content, construct and criterion validity assist in determining if an instrument actually measures what it is intended to measure. Content validity determines "whether it samples the relevant material it purports to cover" (Wolff, 2005, p. 12). Construct validity determines if the instrument measures the characteristic of interest. For instance, the ECIv2.0 should "relate positively with self-esteem (convergent) and negatively with depression (convergent), and not relate with cognitive ability (discriminant)" (Wolff, 2005, p. 12). Lastly, criterion validity determines to what degree the instrument measures correlate with some outcome criteria. For example, a test could be conducted to determine "whether a manager high in EI tends to have lower turnover rates than those managers low in EI" (Wolff, 2005, p. 12). Further, when the measure of interest, in this case the ECI, and the outcome are assessed simultaneously, concurrent validity is tested. However, when the outcome is collected, "some period of time after the variable of interest is assessed," predictive validity is tested (Wolff, 2005, p. 12). All four of these measures have been assessed on the ECI v2.0.

Byrne (2003) surveyed 325 graduate students from three universities of which 59.8% were males, 56.7% were listed as Caucasian, and whose mean age was 31.23 years, and mean work experience was 8.463 years. The results showed "the ECI was predictive of leadership and related work behavior and explained significant variance in performance after age and personality variables were controlled" (Wolff, 2005, p. 12).

GEQ

The Group Environment Questionnaire (GEQ) is designed for athletic teams and contains an 18-item Likert survey with answers ranging from 1 = *strongly disagree* to 9 = *strongly agree* that is taken by each team member. The GEQ is "designed to assess four specific manifestations" of cohesiveness within teams and is based upon "the assumption that cohesion—a group construct—can be assessed through the perceptions of individual group members" (Carron, Brawley, & Widmeyer, 2002, pp. vii-viii).

The GEQ conceptual model for cohesion evolves from three fundamental assumptions: (a) cohesion can be assessed through the perception of group members; (b) "social cognitions that each group member holds about the cohesiveness of the group are related to the group as a totality and to the manner in which the group satisfies personal needs and objectives"; and, finally, (c) there are two fundamental focuses to a group member's perceptions: a task orientation and a social orientation (Carron, Brawley, & Widmeyer, 2002, pp. 9-10). The task orientation represents motivation towards achieving the group's objectives and the social orientation represents motivation toward maintaining social relationships within the group. Further, the conceptual model contains four constructs: (a) Group Integration-Task (GI-T), (b) Group Integration-Social (GI-S), (c) Individual Attractions to the Group-Task (ATG-T), and (d) Individual Attractions to the Group-Social (ATG-S), the definitions of which can be seen in Table 4.

With regard to instrument reliability of the GEQ, the dynamic multidimensional attributes of the cohesion construct lend itself to low internal consistency values on some scales. “Depending upon when a group is assessed along its continuum of group development, members may or may not have sufficient information to evaluate the cohesiveness represented by a specific scale” (Carron, Brawley, & Widmeyer, 2002, p. 26). A lower internal consistency could be indicative of an immature group development stage where “members have unreliable cognitions about the aspect of cohesiveness” (Carron, Brawley, & Widmeyer, 2002, p. 27). With regard to validity of the GEQ instrument, of the five most important validity measures; e.g., content validity, concurrent validity, predictive validity, factorial validity and construct validity, the first four have been successfully measured for the GEQ. The research described in Appendix D presents evidence for the content, concurrent, predictive and factorial validity of the GEQ.

The GEQ has been tested over the last 20 years (Table 5) with reports of internal consistency measures similar to those originally recorded. However, Paskevich (1995) reported higher internal consistency values and Salimen and Luhtanen (1998) reported lower internal consistency values (Carron, Brawley, & Widmeyer, 2002).

Table 4. Specific Constructs Constituting Perceived Cohesiveness

Desc	Construct	Num	Definition / Survey Questions Associated with Construct
Represented by "I," "my," and "me" perceptions.	Group Integration – Task (GI-T)		Individual team member's feelings about the similarity, closeness, and bonding within the team as a whole around the group's task
		Num	*Survey Questions Associated with Construct*
		1(*)	I do not enjoy being a part of the social activities of this team
		3(*)	*I am not going to miss the members of this team when the project ends*.
		5	Some of my best friends are on this team.
		7(*)	*I enjoy hanging with other people more than with my teammates.*
		9	For me, this team is one of the most important social groups to which I belong.
	Group Integration – Social (GI-S)		Individual team member's feelings about the similarity, closeness, and bonding within the team as a whole around the group as a social unit
		Num	*Survey Questions Associated with Construct*
		2(*)	*I am not able to utilize my time to the best benefit of the team.*
		4(*)	*I am unhappy with my team's level of desire to produce a quality product on time*.
		6(*)	This team does not give me enough opportunities to improve my personal performance.
		8(*)	*I do not like the style of work on this team*.
Represented by "us," "our," and "we" perceptions.	Individual Attractions to the Group – Task (ATG-T)		Individual team member's feelings about his or her personal involvement with the group's task, productivity, and goals and objectives
		Num	*Survey Questions Associated with Construct*
		10	Our team is united in trying to reach its goals for performance.
		12	We all take responsibility for any losses or poor performance by our team.
		14(*)	*Our team members have conflicting aspirations for the team's performance*.
		16	*If members of our team have problems on the project, everyone wants to help them so we can get back together again*.
		18(*)	*Members of our team do not communicate freely about each other's responsibilities*.
	Individual Attractions to the Group – Social (ATG-S)		Individual team member's feelings about his or her personal acceptance and social interactions with [the group]
		Num	*Survey Questions Associated with Construct*
		11(*)	Members of our team would rather go out on their own than get together as a team.
		13(*)	Our team members rarely party together.
		15	Our team would like to spend time together in the off-season.
		17(*)	Members of our team do not stick together outside of practice.

Note: (*) Items must be reverse-scored (i.e., 9 = 1, 8 = 2, 7 = 3, 6 = 4, 4 = 6, 3 = 7, 2 = 8, and 1 = 9)
Italics represent questions modified for use in the software engineering version of the GEQ (Wellington, Briggs, & Girard, 2005). Data and tables consolidated from G.E.Q. *The Group Environment Questionnaire Test Manual* (pp.10-18) by A. V. Carron, L. R. Brawley, & N. W. Widmeyer, 2002, Morgantown, WV: Copyright 2002 by Fitness Information Technology, Inc. Adapted with permission.

Table 5. GEQ Cronbach Alpha Values for Internal Consistency Measures

Source	GI-T	GI-S	ATG-T	ATG-S
Widmeyer, Brawley & Carron (1985)	$\alpha = 0.70$	$\alpha = 0.76$	$\alpha = 0.75$	$\alpha = 0.64$
Paskevich (1995)	$\alpha = 0.73$	$\alpha = 0.75$	$\alpha = 0.84$	$\alpha = 0.62$
Salimen & Luhtanen (1998)	$\alpha = 0.67$	$\alpha = 0.49$	$\alpha = 0.60$	$\alpha = 0.69$

Note: From G.E.Q. *The Group Environment Questionnaire Test Manual* (p.26) by A. V. Carron, L. R. Brawley, & N. W. Widmeyer, 2002, Morgantown, WV: Copyright 2002 by Fitness Information Technology, Inc. Reprinted with permission.

Specifically, the research has shown: (a) that cohesion predicts resistance to disruption, (b) perceptions of cohesion are negatively related to increasing group size, (c) task-oriented leadership style has a strong relationship to task cohesion and person-oriented leadership style has a strong relationship with social cohesion, (d) team building has a positive influence on task cohesion, (e) cohesion is positively related to both role clarity and role acceptance, (f) social loafing is negatively associated with task cohesion, (g) attributions of responsibility for failure are team enhancing rather than self-enhancing, (h) task cohesion is positively related to collective efficacy, and finally, (i) that coordination and communication are related to team cohesion (Carron, Brawley, & Widmeyer, 2002).

Assumptions and Limitations

As with all studies the first and foremost limitation is the elicitation and participation of panel members. An additional limitation of this study is the accurate and timely reporting of each panelist. Other limitations of this research are the scope of the literature review, the selection of the panel members, and the researcher's interpretation of the collected data. With regard to the scope of the literature review, the investigation of the literature found recent research regarding the measurement and interpretation of team cohesion. Although this researcher used all available means, from the university libraries to professional libraries and online journals, it is possible that not all-pertinent information was reviewed for this study; therefore the assumption of this study is that all pertinent information was reviewed at the time this research was conducted.

The researcher's worldview is that of a project manager, a senior software engineer, and a scientist. The researcher has spent the last twenty-five years developing software for the Department of Defense (DoD) and holds a Bachelor of Science in Physics and a Master of Science in Management Information Systems. However, the researcher has been studying emotional intelligence and its importance in creating ideal work teams for the past fifteen years and has encountered, first hand, projects that have failed due to poor project management. This researcher is very motivated in discovering how the software estimation practice can be improved and in investigating how the effects of emotional intelligence, team cohesion, and task cohesion can assist in improving the measurement of software estimation. Therefore, this researcher's motivation may bring an unintended bias to the research.

Nature of the Study

Using the Delphi method (Brown, 1968; Fischer, 1978; Linstone & Turoff, 2002; Okoli & Pawlowski, 2004; Skulmoski, Hartman, & Krahn, 2007; Stewart, 2001) this research elicited the opinions of recognized experts in the fields of project management, software development, emotional intelligence, and team cohesion. The experts, called panelists henceforth, received a total of three e-mail interviews. Panelists were blind copied (bcc'd) in each e-mail, isolating each panelist's identity. The blind copy format provides a more robust e-mail interview process because each panelist focused answering the questions rather than their ego (Fischer, 1978). The panelists provided guidance and perspectives regarding the effectiveness of the COCOMO II.2000 TEAM variable and expert opinion on two other team cohesion methods. An edited copy of the first two chapters of this study was attached to the first e-mail interview. The edited copy contained Chapters 1 and 2 excluding any references to the panel members.

Organization of the Remainder of the Study

The remainder of the study is organized into five chapters, a list of references and appendices. Chapter 2 presents a review of the latest information regarding the importance of team cohesion, software estimation, and emotional intelligence and how the three interact with one another. Chapter 3 presents the methodology of the study delineated in the following manner: (a) the purpose, (b) the design and methodology, and the (c) sample and setting of this study. Chapter 4 presents the findings of the study and discusses the collection, analysis, and interpretation of the results. Chapter 5 presents the conclusions and recommendations of the research and the study is concluded with the reference list and the appendices.

CHAPTER 2. LITERATURE REVIEW

The Importance of Team Cohesion in Software Development

The ability of a software development team to solve complex tasks collaboratively is a major determining factor toward project success (Chiang & Mookerjee, 2004; Kang, Yang, & Rowley, 2006; Mills, 1999). Team cohesion is considered to be high when software development team members are respected as individual contributors to the success of the project and can easily collaborate in an open environment without fear of judgment. However, when unmanaged conflict exists within the team, communication and collaboration become strained, leading to high stress levels and dysfunctional attitudes. Team cohesion has been successfully correlated with work performance; that is, the level of commitment and collaborative behavior each team member exhibits is correlated with team cohesion (Pillis & Furumo, 2006; Rapisarda, 2002b) and when unmanaged conflict remains unchecked it destroys team cohesion (Kang, Yang, & Rowley, 2006). Additionally, a recent study examined the relationship between EI and the perception of team cohesion and determined “a statistically significant positive correlation between an individual’s level of emotional intelligence and his or her corresponding perception of team cohesiveness” (Sandvig, 2008, p. ii). Furthermore, “managers need to be aware of the implications of hiring or placing individuals with low levels of emotional intelligence on teams [especially]…if the individuals will work in highly collaborative team environments, such as software development projects” (Sandvig, 2008, p.ii).

Team Cohesion in Software Development

In software development teams, the ability for a new team member to be as productive as an exiting team member is quite difficult, and almost impossible (Brooks, 1995; Eskerod & Blichfeldt, 2005). Sawyer (2004) determined that "software development is an impressively complex socio-technical activity" (p. 95) where limited and/or strained interaction between team members reduces team cohesion. Therefore, establishing and maintaining team cohesion is critical to the success of a software development project. Maxwell (2001) stated in his best selling book *The 17 Indisputable Laws of Teamwork,* "bad attitudes…will always cause dissension, resentment, combativeness, and division on a team [and] they will never go away on their own if they are left unaddressed. They will simply fester and ruin a team—along with its chances of reaching its potential" (p.112).

In an article titled "Who Should Work With Whom? Building Effective Software Teams," Gorla and Wah Lam determined intra-team communication and coordination affect project performance and "it appears the human aspects of software development are more important than the technological aspects for better performance" (2004, p. 79). Additionally, Duke Corporate Education's book *Building Effective Teams* stated "team capability is more than just the sum of individual skills and knowledge, it is…understanding the unique competencies that each person brings to the group…and helping them to work collaboratively" (2005, p. 26). In another article titled *Examining Team Cohesion In Software Engineering,* Wellington, Briggs and Girard (2005) found that team cohesion measures change over time and should be measured in intervals throughout the project life cycle. Additionally, within software engineering, cohesion measurements should consider both the social aspect of the team, as well as, the team's connection to ensuring project success. "Indeed, the centrality of cohesion as a mediator of group formation, maintenance, and productivity has led some social scientists to deem it the most important small group variable (Golembiewski 1962; Lott & Lott 1965)" (Bollen & Hoyel, 1990, p. 479).

Measuring Team Cohesion

In the past decade, the ubiquitousness and necessity of work teams have prompted concerns of team performance and absenteeism, thus, instigating renewed research within team cohesion and its potential measurement benefits; that of developing a higher functioning team and discerning potential areas of concern (Chiang & Mookerjee, 2004; Dawson, 2000; Higgs, 2005; Hoegl, Weinkauf, & Gemuenden, 2004; Hogan & Thomas, 2005; Kang, Yang, & Rowley, 2006; Karn & Cowling, 2005; Sawyer, 2004; Sawyer & Guinan, 1998; Somech, Desivilya, & Lidogoster, 2008; Stevens, 1998). However, as one team cohesion research group found in reviewing previous literature regarding measurement and conceptualization, the research "revealed serious concerns about clarity and precision among researchers [and] …the lack of a widely accepted theoretical definition…As a result, the rather large literature on group and social cohesion was recently dubbed, *a legacy of confusion* (Mudrack 1989)" (Bollen & Hoyle, 1990, p. 497).

Wech, et al. (1998) found in their cross-level examination of perceived task performance that work group cohesiveness affects individual performance and organizational commitment, additional facets of team cohesion. Specifically, the commitment of each group member to successfully complete the assigned group task influences performance of the individual and, thus, the cohesiveness of the group. The more complex and socio involved group tasks, if successfully completed, the higher the team cohesion measurement and organizational commitment (e.g., Gully, Devine, & Whitney, 1995; Klein & Mulvey, 1995; Mullen & Cooper, 1994; Schachter, Ellertson, McBride & Gregory, 1951)

Continuing the correlation between task commitment and team cohesion, Carless and DePaola (2000) developed a three factor model called the Team Cohesion (TC) scale with constructs of "(a) task cohesion, the extent to which the team is united and committed to achieving the work task; (b) social cohesion, the degree to which team members like socializing together; and (c) individual attraction to the group, the extent to which individual team members are attracted to the group" (Carless & DePaola, 2000, p. 79). The ten variables used to measure the correlations between these three factors are as follows: (a) team morale/spirit, (b) social support, (c) communication/cooperation within team, (d) workload sharing, (e) participation, (f) goal interdependence, (g) preference for team work, (h) job satisfaction, (i) team effectiveness, and (j) team work performance (regional manager ratings). In validating the TC scale, Carless and DePaola found "stronger correlations...between task cohesion and the work-group characteristics of social support and communication/cooperation within the team than between social cohesion and work group characteristics" (2000, p. 85).

Finally, Carless and DePaola determined that task cohesion, compared to social cohesion and individual attraction to the group, "was the better predictor of work-group performance." (Carless & DePaola, 2000, p. 85) Further research has established task cohesion as the better predictor of performance. Specifically, "Gulley et al (1995) found that task interdependence was a key factor in creating cohesion. Task interdependence requires group coordination, cooperation, and mutual performance monitoring as well as sustained communication among members" (Knouse, 2006, p. 590). Additionally, Mullen and Copper (1994) found task commitment to be the strongest of the three types of cohesion when compared with the interpersonal attraction and identity (group pride) (Knouse, 2006). Knouse (2007) argues that "highly cohesive teams tend to be more successful and efficient in problem solving...particularly for complex and extended tasks…[and] tend to persevere in the face of stress, whereas less cohesive teams tend to fly apart (Knouse, 2007, p.49)." Knouse continues to argue that cohesion develops as a result of the task at hand and not social interaction among team members and stated "As one manager I worked with said 'I do not have to like the members of my team, but I had better be able to respect what they can contribute to the mission of the team (Knouse, 2007, p51).'" Clearly, task cohesion is the key factor in measuring team cohesion.

Furthermore, when measuring team cohesion, all team cohesion instruments mentioned in the previous research were designed for all team members to participate in the survey in order to accurately determine the team's team cohesion measure; therefore, the fact that COCOMO II.2000 only requires the assessment of one individual to determine the team's team cohesion measure presents a very real problem toward providing an effective measure. In his article "3-D Chess: Boosting Team Productivity Through Emotional Intelligence," Barth quotes John D. Mayer, professor of psychology at the University of New Hampshire, as stating individuals' assessments of their own EI are "'not highly correlated with their actual emotional intelligence'" (2001, p. 4). Barth further quotes Mayer's retelling of a situation where a concerned supervisor who sensed something was amiss with an off-site team determined through the use of emotional intelligence testing, which requires all team members to participate, an underperforming team member who had a very low EI score. The team was covering for him because the underperforming team member was "politically connected to another supervisor in the firm [and] his fellow team members felt they couldn't criticize him" (Barth, 2001, p. 4). In this instance a supervisor could not determine on his own what was causing the team to underperform. Only through all team members participating in an EI assessment did the supervisor discover the problem. One individual's subjective assessment of how well a team coheres is not as effective as the ability of all team members' participating in the measure to assess the cohesion of its team.

Software Estimation Problems

The historical problem with software estimation within America is the high failure rate recorded within the biennial Chaos reports produced by The Standish Group. The Standish Group reports between the years of 1994 and 2004 on average 20% of projects were on time and on budget, while 55% were late and 25% failed (McConnell, 2006, p25). According to an article titled "Why Software Fails," the annual cost for failed or troubled software is $60 to $70 billion in the United States alone (Charette, 2005). Clearly, the current software estimation practices are not working. According to Humphrey's article, "Why Big Software Projects Fail," the problem is with the "imprecision and inaccuracy of most software project plans" (Humphrey, 2005, p. 27). Towards improving the current estimation practices, Laird (2006) offers three golden rules to software estimation in her "Limitations of Software Estimation" article; they are (a) require the justification of all estimations, (b) use previous projects to validate and tune estimation methods and tools, and (c) estimation accuracy is correlated with training; therefore, "educate your estimators" (p. 45).

The ability to successfully estimate software projects enables one to be ahead of the competition. The ability to successfully deliver on pre-bid estimation is paramount to a successful software company. The ability to provide an accurate estimation is the foundation of the pre-bid process (Nasir, 2006). In order to provide an accurate estimation, a software company can use historical data from previous projects to assist in the estimation process. However, if the current project is completely different than anything the company has produced the risk is increased by +/- 25% (Laird, 2006). Therefore, it is imperative for a company to have established software processes throughout the life cycle of software development in order to provide metrics for use in future project estimations.

In his book *Facts and Fallacies of Software Engineering*, Robert Glass stated "The most important factor in software work is not the tools and techniques used by the programmers, but rather the quality of the programmers themselves" (Glass, 2003, p. 11). He further stated "…The two causes of runaway [projects] that stand head and shoulders above all others are poor (usually optimistic) estimation and unstable requirements" (Glass, 2003, p. 28). Within the thesis of their famous book *Peopleware: Productive Projects and Teams,* DeMarco and Lister state "The major problems of our work are not so much technological as sociological in nature" (DeMarco & Lister, 1999, p. 4).

Software Estimation With Constructive Cost Model (COCOMO) II

There are many approaches to estimating software development, from estimating the software lines of code to using expert judgments as in function point and object point analysis to empirical measurement tools such as COCOMO II.2000 one of the most "widely utilized and accepted estimation methods in software engineering projects" (Benediktsson & Dalcher, 2004, p. 5).

COCOMO II.2000 has three models: (a) the Application Composition model that supports prototyping activities, (b) the Early Design model that supports exploring architectural alternatives and incremental development strategies and (c) the Post-Architecture model that supports the development and maintenance phases of software development (Boehm, Abts, Winsor-Brown, et al., 2000). COCOMO II.2000, the most detailed sub-model within COCOMO II.2000 is an "empirical, well documented, independent model" (Nasir, 2006, p. 308) that accepts SLOC, function points and use-case points as its basic software size input.

Within the software development cycle, the software estimation tool could be utilized after the requirements elicitation and the entire spectrum of the development phase that is broken into prototyping, architectural investigation and final architectural framework design. Within COCOMO II, the Early Design and Post-Architecture models use the same "functional form to estimate the amount of effort and calendar time it will take to develop a software project" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 13) enabling a wide spectrum of measurement for each subject software team interviewed. Therefore, if a subject team is in the design or implementation phase, their team cohesion is measured using the same equation (Figure 3).

In this equation, PM represents the effort expressed in person months. Specifically, the variable *PM* represents Person Months or the amount of time one person would work for one month toward the development of the software product.

A – Effort coefficient that can be calibrated; 2.94 is the nominal COCOMO II.2000 calibrated value

$$PM = 2.94 \prod_{i=1}^{17} EM_i \, S^{0.91+0.01\sum_{i=1}^{5} W_i}$$

B – The scaling base-exponent for the effort equation; 0.91 is the nominal COCOMO II.2000 calibrated value.

Figure 3. Summarized COCOMO II.2000 estimation model. From "On the Sensitivity of COCOMO II.2000 Software Cost Estimation Model" by P. Musilek, W. Pedrycz, N. Sun, & G. Succi, 2002. *Proceedings of the Eight IEEE Symposium on Software Metrics (METRICS'02), Edmonton, Alberta, Canada,* p. 363-379. Copyright by IEEE. Adapted with permission.

Additionally, the number of person-hours per person-month, PH/PM is an "adjustable factor with a nominal value of 152 hours per Person-Month" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 29).

The variable *S* represents the estimated size of the software project to be developed in thousands of Software Lines of Code (KSLOC) or unadjusted function points (UFP) converted to KSLOC.

The constant *A* "approximates a productivity constant in PM/KSLOC for the case where E = 1.0 [additionally] productivity changes as E increases because of the non-linear effects on Size" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 29). The constant *A* is also a calibration constant that is initially set when the model is calibrated using a project database reflecting a global productivity average.

The variable *EM* represents the average of the effort multiplier (EM) values associated with each cost driver. Cost drivers are defined as those factors which increase cost such as the number of person months needed to accomplish the project. These cost drivers have "qualitative rating levels that express the impact of the driver on development effort [and] can range from [Very] Low to Extra High" (Boehm, Abts, Winsor-Brown, et al., 2000, pp. 29-30). EM is an aggregate value of 17 effort multipliers (Table 6) representing the characteristics of the development process which affect the effort needed to complete a software project.

Table 6. COCOMO II.2000 Cost Drivers

Symbol	***Drivers***	***Name***
SF_1	PREC	Precendentedness
SF_2	FLEX	Development Flexibility
SF_3	RESL	Architecture and Risk Resolution
SF_4	TEAM	Team Cohesion
SF_5	PMAT	Process Maturity
EM_1	RELY	Required Software
EM_2	DATA	Data Base Size
EM_3	CPLX	Product Complexity
EM_4	RUSE	Required Reusability
EM_5	DOCU	Documentation Match to Life-cycle Needs
EM_6	TIME	Time Constraint
EM_7	STOR	Storage Constraint
EM_8	PVOL	Platform Volatility
EM_9	ACAP	Analyst Capability
EM_{10}	PCAP	Programmer Capability
EM_{11}	AEXP	Applications Experience
EM_{12}	PEXP	Platform Experience
EM_{13}	LTEX	Language and Tool Experience
EM_{14}	PCON	Personnel Continuity
EM_{15}	TOOL	Use of Software Tools
EM_{16}	SITE	Multi-Site Development
EM_{17}	SCED	Required Development Schedule

Note: From **"**Calibrating the COCOMO II.2000 Post-Architecture Model" by B. Clark, S. Chulani-Devnanzi, B. Boehm, 1998. *Proceedings of the 1998 International Conference on Software Engineering,* pp. 477-480. Copyright by IEEE. Reprinted with permission.

These weighted effort multipliers, whose nominal weight is 1.0, are grouped into four categories (product, platform, personnel, and project) and their product is used to adjust the effort.

The exponent *W* (Figure 3) represents five scale factors (Table 1) (precedenteness [PREC] which "identifies the newness of the project," development flexibility [FLEX] which identifies the "degree of requirements, schedule, interface, [and] …flexibility," architecture/risk resolution [RESL] which represents the "degree of risk present," team cohesion [TEAM] which provides a measure of "project turbulence and entropy of the project team," and process maturity [PMAT] which "uses [a] CMM questionnaire to determine [a] weighted average" (Bernheisel, 1997, p. 33) that accounts for the "relative economies and diseconomies of scale encountered as a software project increases its size based on different nominal values and rating schemes" (Musilek et al. 2002, p.16). The TEAM survey consists of four questions (Table 2) that the project manager answers to determine the TEAM value. The project manager is to evaluate the team by answering each of the four questions with one of six weighted answers (Table 7).

Table 7. COCOMO II.2000 TEAM weighted answers and their values

Very Low	Low	Nominal	High	Very High	Extra High
5.48	4.38	3.29	2.19	1.10	0.0

Note: From **"**Calibrating the COCOMO II.2000 post-architecture model" by B. Clark, S. Chulani-Devnanzi, B. Boehm, 1998. *Proceedings of the 1998 International Conference on Software Engineering,* pp. 477-480. Copyright by IEEE. Reprinted with permission.

The exponent *E* "is an aggregation of five scale factors (SF) that account for the relative economies or diseconomies of scale encountered for software projects of different sizes" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 30). These five scale factors (Table 6) are "a significant source of exponential variation on a project's effort or productivity variation" (Clark, Devnani-Chulani, & Boehm, 1998, p.478). When E = 1.0 the economies and diseconomies of scale are in balance, however, when E > 1.0 the project is exhibiting diseconomies of scale which are generally attributed to growth of interpersonal communications overhead and growth of large-system integration overhead. The five scale factors (Table 6) are (a) Precendentedness (PREC), (b) Development Flexibility (FLEX), (c) Architecture and Risk Resolution (RESL), (d) Team Cohesion (TEAM), and (e) Precendentedness (PREC). TEAM is the variable of concern for this research. TEAM is to represent the software development team's team cohesion measure. TEAM's value is determined by the weighted answers (Table 7) of four very simple questions (Table 2) asked of the project manager. Furthermore, a project manager may not have a day-to-day work relationship with the software development team, as many project managers are managing multiple projects simultaneously (Patanakul & Milosevic, 2009; Tullett, 1996). Additionally, there is also the possibility that a project manager will not have any work history with the software development team members, which makes subjectively estimating their cohesion extremely difficult.

TEAM takes into account the sources of project turbulence and entropy due to challenges in synchronizing the stakeholders, who are considered to be the users, customers, developers and others. As with most projects the challenges "arise from differences in stakeholder objectives and cultures; difficulties in reconciling objectives; and stakeholders' lack of experience and familiarity in operating as a team" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 34). These concerns are measured by the project manager's subjective answers (Table 7) to four questions (Table 2). "The final rating is the subjective weighted average of the listed characteristics" (Boehm, Abts, Winsor-Brown, et al., 2000, p. 34).

Regarding the accuracy and validity of the COCOMO II.2000 model Boehm stated the estimation accurancy is within 30% of the actual development effort (Boehm, 2000). Musilek, et al., (2002) conducted a sensitivity analysis against the COCOMO II.2000 model using mathematical analysis of the estimating equation, Monte Carlo simulation and error propagation and concluded the most critical input variables in order of influence are (a) size S, (b) effort multipliers EM_i followed by (c) scale factors W_i.

According to Musilek, et al., (2002) “size and effort multipliers are very important and adequate time and resources should be devoted to their accurate evaluation. The scale factors are much less important and could be neglected (set to their nominal values) if necessary” (p. 19). Additionally, Musilek, et al., (2002) found the COCOMOII model to have very high sensitivity when cumulative errors are considered. That is to say “a relatively small error in determining projected size of the system could lead to a large error of the estimated value of effort” (p. 19).

In a 1997 DoD research study where the objective was to determine the “accuracy (goodness of fit) of the [COCOMO II.1997] model in default (uncalibrated) and calibrated modes, and validate the model’s use by…[DoD] agencies to estimate program costs and schedules” Bernheisel (1997) found the COCOMO II.1997 model “does tend to produce low estimates [and] the exact reason is unknown but…important to understand” (p. 80). Additionally, Bernheisel (1997) found the use of regression analysis using Mean Magnitude of Relative Error (MMRE) less than 0.25, Relative Root Mean Square (RRMS) less than 0.25, and Prediction Level (Pred) of 0.25 for 75% of the time improved the accuracy of the COCOMO II.1997 results; and concludes with “highly [recommending] this model be used by all Cost Analysts in DoD” (p. 87). Furthermore, Bernheisel (1997) stated “quality, robust data and experienced [COCOMO II.1997] model users will have a great impact on the accuracy of the model, however, when this is not possible, the use of regression techniques can be used to improve the overall accuracy of the estimates” (p. 87).

Nguyen, Steece, and Boehm (2008) present yet another means of improving the calibration of COCOMO II.2000, that of constrained regression. Recognizing that a large number of parameters, twenty-three in COCOMO II.2000, can "lead to strong collinearity, heteroscedasticity, and highly variable prediction accuracy" [and that] "no single model can fit all environments without being adapted or calibrated" they investigated the use of constrained multiple regression. Specifically, they ran "a cross-validation procedure and compare[d] the prediction accuracy from different approaches such as least squares, stepwise, Lasso, and Ridge regression." They found that the regression model "minimizes the mean of relative error and imposes non-negative coefficients" and is thus considered "a favorable technique for calibrating the COCOMO model parameters" (Nguyen, Steece, & Boehm, 2008, p1).

Emotional Intelligence (EI)

The history and foundation of EI began in the 1920s when E. L. Thorndike (1920) suggested another form of intelligence existed which he termed social intelligence and defined it as "the ability to understand men and women, boys and girls-to act wisely in human relations" (1920, p. 228). Additionally, Thorndike (1920) visualized a system in which "we shall be able to make up a bill of specifications on the sort of intellect and character required for a certain job, select men efficiently instead of haphazardly, and train them according to their individual needs instead of indiscriminately' (p. 234)". However, Thorndike was very skeptical that an adequate measurement tool could be developed (Grenier, 2004).

While research in the area of social intelligence continued to be sporadic over the next 60 years, the advances in human intelligence became well formed. However, the completed models of human intelligence still could not answer the question of how some individuals with high IQs are not as successful as those with lower IQs; that is, although intelligence theory ultimately advanced toward the development of a widely accepted measurement tool; e.g. the IQ test; Robert Sternberg noticed that success could not be measured strictly by intellect. Accordingly, in his *Theory of Successful Intelligence* (1999), he defines intelligent people as “those who have the ability to achieve success in life, given one’s personal standards, within one’s socio-cultural context” (1999, p. 293). Moreover, Sternberg’s belief, that the ability to achieve success depended on the capitalization of strengths and compensation of weaknesses, lay the foundation for the study of EI (Grenier, 2004).

The competencies listed for EI are traditionally known as soft or social skills, that is, the ability to successfully interpret emotion and use emotion as an influencing tool; whereas the IQ test was designed to measure how well an individual would do in an academic environment; e.g., cognitive abilities (Herrnstein & Murray, 1996, pp. 1-4; Kemper, 1999, p 16). In today’s team driven industry the ability to successfully collaborate and work efficiently in a team setting are key to business success and are as important, if not more important, as cognitive ability. For example, in 1950, eighty Ph.D.’s were given personality and IQ tests along with interviews while at Berkeley. Forty years later, after estimating their success via experts in their fields, it was discovered that “social and emotional abilities were four times more important than IQ in determining professional success and prestige (Feist & Barron, 1996)” (Cherniss, 2000, p. 5). Additional research at Yale University (Barsade, 2002) discovered the emotions of one person infected the group; that is, “the results indicated that the actor was able to infect the group with his emotion, and good feelings led to improved cooperation, fairness, and overall group performance” (Cherniss, 2000, p. 7). While describing the job of a corporate consulting engineer whose job it is to “rescue projects in danger of going off the track,” Susan Ennis of Bank Boston stated “‘What makes the difference was not their brain power...but their emotional competence…It’s their abilities to listen, to influence, to collaborate, and to get people motivated and working together well’” (Goleman, 2000, p29).

Team Emotional Intelligence and Team Cohesion

As stated earlier, the primary work unit of the 21st century is the team, that is, a team of individuals, managed by a project lead and/or a project manager, working toward completing an assigned task. In a recent article titled *The Impact of Emotional Intelligence on Work Team Cohesiveness and Performance* (Rapisarda, 2002a), a *team* is defined as “individuals who see themselves and who are seen by others as a social entity, who are embedded in one or more larger social systems (e.g., community, organization), and who perform tasks that affect others (such as customers or coworkers) (Guzzo & Dickson, 1996, p. 308)” (Rapisarda, 2002a, p. 366). Rapisarda defines *cohesiveness* as “the forces acting on members to remain in the group” and stated that “cohesiveness did in fact promote productivity (Summers, Coffelt, & Horton, 1998) and usually increased performance (Worchel, Cooper, & Goethas, 1991)” (Rapisarda, 2002b, p. 15).

Technical skills can make a company competitive but it is the collaborative success of the employees, customers and suppliers that are the foundation of the true ability to perform better than the competition (Goleman, 2000). Unresolved conflict (Sikes, Gulbro, & Shonesy, 2010) is the great enemy of a successful project, which is defined as a project delivered on time, on budget, and on target. “Groups that ignore inappropriate member behavior in an attempt to avoid conflict decrease their ability to solve problems…[which] frequently results in hostility and reduced performance (Nemeth & Staw, 1989)" (Wolff, Druskat, Koman, & Messer, 2006).

In examining the role of EI as a predictor of group effectiveness, Jordan and Ashkanasy (2006) discovered that high emotional self-awareness not only predicts team effectiveness but also provides the resources for team members to affectively resolve team conflict. Additionally, Elfenbein (2006), while researching team EI effects on performance, discovered that "greater accuracy in understanding colleagues' positive emotions predicted better team performance" and suggests managers use EI as a tool to create and establish group norms (Elfenbein, 2006, p. 180). Recognizing the importance of conflict resolution within the project team, Sunindijo and Hadikusumo (2005), discovered that project managers with high EI measures tend toward collaborative and accommodating conflict resolution styles of leadership which has proven best in resolving conflict because it yields a win-win situation (Bushyacharu, 1996).

An emotionally intelligent leader, according to Goleman, is self-aware, motivated, empathetic, can self-regulate emotions, and has good social skills (2004, p.88). Additionally, an emotionally intelligent leader can lead his/her group toward becoming an emotionally intelligent group (Leban, 2003, 2004) through the establishment of emotionally intelligent group "norms that build trust, group identity, and group efficacy" (Druskat & Wolff, 2001, p. 89). According to Druskat & Wolff (2001), a team, made up of emotionally intelligent members, does not necessarily make an emotionally intelligent team; it is the creation and reinforcement of a "team atmosphere in which the norms build emotional capacity (the ability to respond constructively in emotionally uncomfortable situations) and influence emotion in constructive ways" (p. 82). Groups that are emotionally intelligent have been found to be more cohesive, productive, loyal, and are able to handle work environment conflicts in a productive manner (Rapisarda, 2002b). The only research found to date specifically discussing the association between team cohesion and EI is *The Impact of EI on Work Team Cohesiveness and Performance* (Rapisarda, 2002b) which determined the EI competencies correlated with team cohesiveness: (a) achievement orientation, (b) empathy, (c) influence, (d) communications, (e) leadership, (f) conflict management, (g) self-control, (h) adaptability, and (i) building bonds.

Conclusion

Therefore, what can be concluded from this overview of software engineering, software estimation, team cohesion, emotional intelligence and the COCOMO II.2000 estimation model's TEAM variable is that

1. Software development has incurred a staggering rate of failure
2. The amount of complex tasks required of a software development team are at very high levels and the success of collaborative communication between team members is of the utmost importance toward successful software development that is delivered on-target, on-budget, and on-time.
3. Team cohesion cannot be effectively measured using one individual's subjective opinion.
4. ECIv2.0 and GEQ questions have been successfully correlated with team cohesion.
5. Emotional Intelligence and Group Emotional Intelligence have had recent breakthroughs in measuring team cohesion.
6. Team cohesion is better measured using the task cohesion focus.

It is the goal of this research to determine if the current COCOMO II.2000 TEAM method is effective in measuring team cohesion. Furthermore, if the COCOMO II.2000

TEAM method is determined to be ineffective, it is the goal of this research to investigate the effectiveness of the GEQ method and the team cohesion correlated questions of the ECIv2.0 as substitutes for the current COCOMO II.2000 TEAM method. Lastly, it is the goal of this research to determine via expert opinion what, how, and when questions should be asked in determining a software development team's team cohesion measure.

CHAPTER 3. METHODOLOGY

This research initially began as a longitudinal and quasi-experimental study where a pretest and posttest was to be given to software development teams at the beginning and end of a three-month period. The collected data would have been quantitative and the unit of analysis would have been the team cohesion measure of the software development team. Using this research design, this researcher intended to provide quantitative results that would determine if the COCOMO II.2000 TEAM method of measuring team cohesion was effective. A power analysis tool determined 250 software development teams of ten plus members were required in order to present valid research. The research proposal was reviewed and approved by the IRB and Capella School of Business and Technology, however, after six months of unsuccessfully eliciting 250 software development team participants, this researcher and her committee determined to redirect the research design toward using the Delphi method.

Fortunately, this researcher had made significant contacts with experts in the project management, software engineering, emotional intelligence, and team cohesion industries while eliciting participants for the original research design. Using e-mail, this researcher contacted these experts individually, shared with them the change of research design, and asked if they would be interested in participating on a Delphi panel where the goal was the investigation of the effectiveness of the COCOMO II.2000 TEAM model; i.e., the four questions the project manager answers to determine the team's cohesion value. All contacted experts stated they would be very interested in participating on the panel. Therefore, this research continued forward with the use of the Delphi method (Brown, 1968; Fischer, 1978; Linstone & Turoff, 2002; Okoli & Pawlowski, 2004; Skulmoski, Hartman, & Krahn, 2007; Stewart, 2001) toward gaining knowledge of the effectiveness of the COCOMO II.2000 TEAM variable. Further, this research asked the panelists' to investigate the effectiveness of two other team cohesion models as substitutes for the COCOMO II.2000 TEAM method. Finally, this research asked the panelists' to provide guidance on how often a software development team should be measured for team cohesion, as it has been determined that team cohesion changes over time (Wellington, Briggs, & Girard, 2005).

Research Design

The Delphi method (Brown, 1968; Fischer, 1978; Linstone & Turoff, 2002; Okoli & Pawlowski, 2004; Skulmoski, Hartman, & Krahn, 2007; Stewart, 2001) is concerned with the opinion and ideas of experts. This research is considered qualitative in that a survey with open-ended questions (Robson, 2002) was used to gain the data for this research via the panelists. Additionally, this research is considered evaluative in that it is evaluating an existing entity, specifically, the TEAM variable within the COCOMOII.2000 software estimation model and that the data collected is evolving through each round. According to Stewart (2001), qualitative research is concerned with gaining greater depth of understanding and insight into a particular phenomenon. That is "the analysis process is interpretive and the researcher viewed as the instrument of analysis" whereas quantitative research is concerned with discovering "causal links or associations" within data retrieved from an area of study (Stewart, 2001, p. 922). Stewart further argues, "although an interpretative analysis of qualitative data happens in the first stage, the Delphi is fundamentally reductionist in nature…[and] the researcher employs a standardised, 'objectified' technique to interact with the participants" (Stewart, 2001, p. 922). Based upon the arguments of Stewart (2001), the Delphi method would be considered a mixed methodology, however, in this research, the design leans heavily toward a qualitative methodology due to the open-ended question e-mail interviews within this design and therefore should be considered qualitative.

Additionally, from a real-world perspective, this research design is considered qualitative or flexible; the research approach is considered an evaluation, and the research strategy a case study (Robson, 2002). According to Robson, the purpose of an evaluation research approach is to "assess the effects and effectiveness of something, typically some innovation, intervention, policy, practice or service" (Robson, 2002, p. 202). This research evaluated the effects and effectiveness of the COCOMO II.2000 TEAM method in measuring team cohesion. Additionally, Robson describes a case study as a

> well-established research strategy where the focus is on a case (which is interpreted widely to include the study of an individual person, a group, a setting, an organization, etc.)" and defines the case study as a "strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence (Robson, 2002, p. 178).

As the COCOMO II.2000 model is considered a quantitative or empirical model, the TEAM variable also wears that label. However, the TEAM variable is measuring a qualitative phenomenon, specifically team cohesion, in an empirical manner, therefore, this study falls into line with Robson's case study; e.g. this research is looking at an innovation, that is, a software estimation tool, and specifically a single instance of an empirical variable (TEAM) to determine its effectiveness.

Furthermore, the Delphi method, initially developed by the RAND Corporation to elicit a consensus opinion to provide long-range social and technical forecasting expertise (Dalkey & Helmer, 1963), uses a panel of experts to elicit expert ideas and/or opinions toward an area of study. The Delphi methodology replaces face-to-face debate with sequential e-mail interrogations where each panelist provides responses to questions regarding the research subject in anonymity. The anonymity aspect of the Delphi method curtails any concern of ego and heated debate that would detract from the timely success of the research. Additionally, anonymity allows each expert panelist the time to form his/her response without being put on the spot and affords each panelist opportunity to digest conflicting opinion without the detraction of knowing who the individual respondent was or what they may be representing.

A panel of project management, software development, emotional intelligence, and team cohesion experts has been elicited to participate in this research via e-mail. The experts' opinions and ideas regarding the effectiveness of the COCOMO II.2000 TEAM variable was elicited via three interrogative rounds of e-mails where the answers of each participant remained unattributed. The instrument used for this research is completely described in the Instrumentation/Measures section of this document.

Sample

The sample of this research is made up of a panel of experts representing the following areas of expertise: (a) project management, (b) software estimation, and (c) team cohesion measures. The demography of the panel members are as follows: there are a total of eight panel members, 80% are men and 20% are women; four panelists represent the area of team cohesion, two panelists represent the area of software estimation, two represent the area of project management. A brief description of the panelists is listed next.

Team Cohesion (1) Panelist

Associate Professor of Organizational Behavior and Management and author of award winning research that examines how teams and leaders effectively manage complex interpersonal and coordination challenges in cross-functional, cross-cultural and self-managing work environments – which often involves aspects of emotional intelligence. Research has appeared in prestigious journals such as the Academy of Management Journal, Harvard Business Review, Human Relations, the Journal of Applied Psychology, Leadership Quarterly, Sloan Management Review, and Small Group Research.

Team Cohesion (2) Panelist

Professor in the Departments of Organizational Behavior, Psychology, and Cognitive Science and Human Resources is author of more than 100 articles on leadership, competencies, emotional intelligence, management education, and thematic analysis. Also holds a BS in Aeronautics and Astronautics from MIT and a MS and Ph.D. in Social Psychology from Harvard University.

Team Cohesion (3) Panelist

Trustee Chair and Professor of Psychology has a Ph.D. in industrial and organizational psychology and has co-authored over 300 journal articles and book chapters and has co-edited 15 books. Currently serves on the editorial boards of Journal of Applied Psychology, Personnel Psychology, Military Psychology, Interamerican Journal of Psychology, Applied Psychology: An International Journal, International

Journal of Aviation Psychology, Group Dynamics, and Journal of Organizational Behavior and is past Editor of Human Factors journal. Expertise includes helping organizations on how to foster teamwork, design and implement team training strategies, facilitate training effectiveness, manage decision making under stress, develop performance measurement tools, and design learning environments.

Team Cohesion (4) Panelist

A department chair, professor of computer science, and, a leader in large software development organizations involved in building operating systems and real-time embedded applications. A consultant that uses the combination of academic and industrial experience to help companies question their assumptions about development processes to improve their agility and product quality.

Project Management (1) Panelist

Practiced computer and systems engineering for the past quarter century for the DoD and is now a Senior Principal Systems Engineer. Has a PhD in Electrical Engineering and has written several books on the systems aspects of project management.

Project Management (2) Panelist

As a CEO and Chief Software Engineer this panelist holds a Bachelor of Science and a Master of Science degree in software engineering. He currently consults, teaches, oversees software development practices, and is the author of award winning books and articles. He has expertise in project estimation, software construction practices, agile and rapid development methodologies, and outsource software management. Has been recognized as one of the three most influential people in the software industry, was Editor in Chief of IEEE Software magazine, and serves on the Panel of Experts that advises the Software Engineering Body of Knowledge (SWEBOK) project.

Software Estimation (1) Panelist

Has a Master of Science in Electrical Engineering from California Institute of Technology and a Bachelor of Engineering Science from Rensselaer Polytechnic Institute. He has been involved in such programs as International Space Station SEPG, C-17 Software Program Management, software engineering technology transfer at Douglas Aircraft Co., Modula-2 for Volition Systems, and UCSD Pascal for General Automation and has over 25 years of experience in software development, engineering and management while working on aerospace, industrial and commercial applications.

Software Estimation (2) Panelist

Has more than 40 years of practical experience as a computer professional in hardware and software development, and for the past 30 years, has been actively engaged in software engineering methods, tools, quality software management methods, software schedule and cost estimation, and management metrics. Retired as Chief Scientist of the Software Engineering Division and was responsible for research in software engineering methods and management.

Setting

The setting consists of e-mail correspondence between the researcher and the expert panelists. Three rounds of surveys were sent to the panelists. Upon the completion of each round, each panelist's comments was consolidated and placed in a unattributed summary and forwarded to each panelist for additional review and comment. At the end of the second round the consensus was collected and summarized into a final executive summary that was attached to the final survey. Although panelists' had the option to comment on other panelists' responses none of the panelists exercised this option.

Instrumentation / Measures

The instruments used within this research were three rounds of e-mail interviews sent to each panelist in anonymity. The first round e-mail (Appendix A) contained the following questions:

1. Do you believe this research has provided enough evidence to warrant the investigation of the effectiveness of the COCOMO II.2000 TEAM variable? Why or Why not?

2. In light of the team cohesion literature review within this research do you believe the COCOMO II.2000 TEAM variable effectively measures team cohesion? Why or Why not?

3. Is the project manager's subjective assessment of the software development team an effective method to measure team cohesion? Why or Why not?

4. Do the four survey questions used to determine the COCOMO II.2000 TEAM variable's value effectively assesses a software development team's team cohesion measurement? Why or Why not?

5. Based upon your expertise and the literature review of this study does the COCOMO II.2000 TEAM variable take into consideration how team cohesion changes over the life cycle of the project? Why or Why not?

6. What suggestions, if any, would be the best way to introduce the dynamic phenomena of team cohesion changing over the life-cycle of the project into the current COCOMO II.2000 model?

7. In your expert guidance how important is task cohesion as a factor of success in software development? Why?

The second round e-mail (Appendix B) contained a summary of the panelists' responses from Round 1 and asked the panelists the following questions:

1. The ECI model was used in previous research that determined nine emotional intelligence competencies strongly correlated with team cohesion (Rapisarda,

2002b). The survey questions associated with these nine competencies are attached to this email. In your opinion would these questions create a more effective method in determining the team cohesion value for the COCOMO II.2000 TEAM variable? Why or Why not?

2. In your opinion, would the questions of the GEQ survey modified for software engineers which directly correlate to team cohesion (Wellington, Briggs, & Girard, 2005), attached to this email, be a more effective method in determining the team cohesion value for the COCOMO II.2000 TEAM variable? Why or Why not?

3. Based upon your expertise and the research literature presented in this study how often should a team cohesion survey be administered during the life cycle of a software development project? Why?

4. When a team member exits and a replacement is brought on board when should team cohesion be measured? When the team member exits? Soon after the team member exits? After the replacement is hired but before they report for work? Soon after the replacement is hired? Why or Why not?

The third and final e-mail (Appendix C) contained a summary of the previous two e-mails and the following questions:

1. In your expert guidance, what research should be accomplished next to improve software estimation? How can the use of emotional intelligence assessments assist in improving software estimation?

2. This is the final e-mail. You have received a summary of the results to date of the panel. Do you have any further comments?

The results of the survey questions were consolidated, summarized, and finally, interpreted as expert opinion.

Data Collection

The data, the responses from the e-mail interviews, was collected as described in the Instrumentation/Measures section of this document. Summaries of the responses to the e-mail interviews were developed and presented as an attachment in follow on e-mails. After all three interviews were accomplished; all corresponding data was reviewed, analyzed, and summarized and can be reviewed in Chapter Four.

Data Analysis

The collected data was reviewed, summarized, and analyzed by the researcher with the intent of ascertaining if the goals of this research were accomplished. Chapter Four contains a histogram of the gender and expertise breakdown of the panelists. Additionally, each interview question and panelist's responses are presented in table format and a histogram presents the responses by percentage and occupation category. The interview data are organized into expertise categories and the values of the data are presented in a histogram format. Additionally, the researcher gleaned noted suggestions and observations of each panelist, and presented these observations in a within Chapter Five.

CHAPTER 4. RESULTS

Delphi Participants

The panelists are well-known experts and published professionals in their respective fields. The panelists' biographies are listed in Chapter Three. Within the following results each panelist is identified by their primary area of expertise and a number, as is listed in the Sample section of Chapter 3 of this document. Although each panelist had opportunity to comment on the responses of other panelists, none of the panelists exercised that option.

Delphi Results

Panelists were sent three rounds of e-mail surveys along with a summary of the previous round's results. The results are grouped into three expertise categories: (a) team cohesion, (b) project management, and (c) software estimation and are visually represented using a 100% stacked column chart.

Round 1 Responses

The Round 1 questions were to determine if the effectiveness of the current method of measuring team cohesion within COCOMO II.2000 was an adequate representation of team cohesion, as it is known today. Each response to the questions are presented with a 100% stacked column representation of the panelists' responses, a table that organizes the panelists' responses by expertise, and an analysis of the panelists' responses.

Round 1-Question 1 responses.

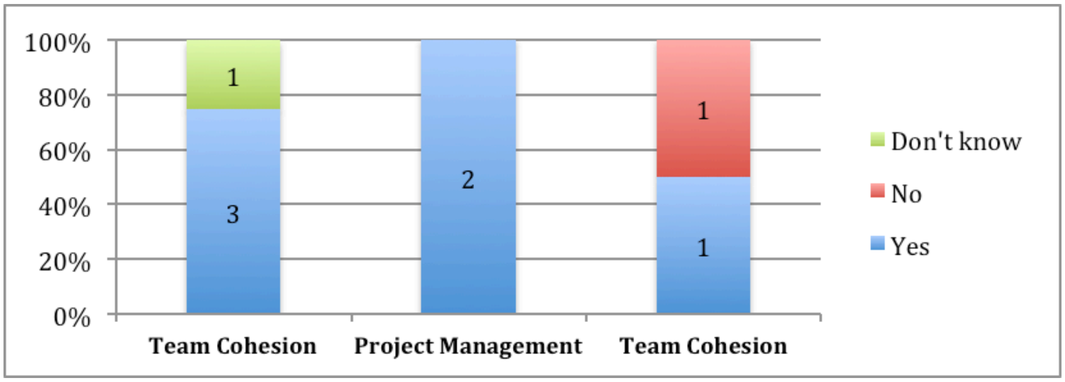

Figure 4. Round 1-Question One responses distributed by expertise

Table 8. Round 1-Question 1 responses by expertise

Round 1 Question 1	**Do you believe this research has provided enough evidence to warrant the investigation of the effectiveness of the COCOMO II.2000 TEAM variable? Why or Why not?**
Team Cohesion (1)	Yes, for two reasons; (1) I don't believe the project manager is best suited to effectively measure the team. In my experience team managers have inflated views of a team's dynamics – perhaps because team members don't always provide team mangers with an accurate view of what goes on in the team; (2) The four questions do not measure what I believe are the most important issues for determining whether a team is functioning well.
Team Cohesion (2)	"No comment."
Team Cohesion (3)	"No; team cohesion is much more complex than what is articulated in these two chapters and the chapters do not provide a coherent, precise set of theoretical drivers guiding the research. The link to EI is not clear nor [do] I think [EI] is relevant. I must confess I am not a fan of EI and its link to team effectiveness is weak at best."
Team Cohesion (4)	"Yes."
Project Management (1)	"Yes."

Round 1 Question 1	Do you believe this research has provided enough evidence to warrant the investigation of the effectiveness of the COCOMO II.2000 TEAM variable? Why or Why not?
Project Management (2)	"Yes. I agree that having one person assess the four attributes within the TEAM factor accurately for an entire workgroup is prone to an inherent subjectivity, which leads to error. As the workgroup becomes larger, the ability of one estimator to accurately assess the 4 TEAM factors diminishes, yet the team factor is a scaling factor that has greater influence at larger project sizes. Thus I would predict that the TEAM adjustment would introduce more error at larger project sizes, which is exactly the opposite of what we want."
Software Estimation (1)	"Yes. The research really shows that the definition of the TEAM parameter is focusing on an area that is outside software development and is more appropriate to the COSYSMO model. The parameter is looking [more] at the system acquisition (stakeholders, etc.) than it is looking at the dynamics of the software development itself. As a simple example, look at the productivity (cost, schedule) of a traditional software project versus that of an agile development. None of the current TEAM parameters or questions is concerned with the structure and environment of the 'team'."
Software Estimation (2)	"Yes; it is always good to question things that look like assumptions."

Although seven of the eight panelists affirmed this research provided enough evidence to warrant the investigation, only four of the seven provided supporting arguments. Two of the four panelists argued against the subjective viewpoint of the project manager in providing the team cohesion measure primarily because "team managers have inflated views of a team's dynamics" and because the subjectivity of one person regarding a group is "prone to an inherent subjectivity, which leads to error." Much of the recent team cohesion measurement research presented within this research affirms that position.

One of the four panelists stated, "It is always good to question things that look like assumptions." This researcher is unaware of any 'assumptions' made or mentioned within this research regarding COCOMO II. His statement infers the TEAM scale factor looks like an assumption. This researcher is unsure of the direction or purpose of the statement other than its face value.

Of even more interest is the supporting argument from Software Estimation (1) panelist whose obvious expertise is in Software Estimation. His response has deeper analysis in that he breaks down the various stages of development and the different COCOMO models used within estimation. Although this research primarily focuses on the Post Architecture sub model of the COCOMO II.2000 estimation model, this panelist provides insight that the TEAM variable measurement questions are more suited for the COSYSMO model.

The COSYSMO model was developed after the COCOMO II.2000 model and is used for systems (hardware) development where as COCOMO II.2000 is used for software (software) development. The COSYSMO model also has a team cohesion variable very similar to the COCOMO II.2000 model (Valerdi & Boehm, 2010). This panelist argues the questions used in the COCOMO II.2000 model do not adequately address the team dynamics of the actual development but more at the acquisition stakeholders level. However, the COCOMO II.2000 model defines the TEAM scale factor as a variable that represents the sources of entropy due to difficulties between stakeholders where stakeholders include users, customers, developers, maintainers, interfacers, and others (Boehm, Abts, Winsor-Brown, et al., 2000).

Additionally, the COCOMO II.2000 model was designed for use throughout the entire life cycle of the software development process. Specifically, COCOMO II.2000 provides a three-model approach to the software development lifecycle: (a) the application composition model, (b) the early estimation model, and the (c) post architecture model. The application composition model supports the initial phases of prototyping and discovery; the early estimation model supports the architectural framework discovery and development; and the post architecture model supports the implementation and maintenance of the software development life cycle. The early estimation model and the post architecture model use the same mathematical formulae and scale factors (Table 6) but differ in the number and type of effort multipliers used. The early estimation model uses seven effort multipliers (Boehm, Abts, Winsor-Brown, et al., 2000, pp 52-55), whereas, the post architecture model uses seventeen effort multipliers (Table 6). Furthermore, the TEAM scale factor is used in the early design and post architectural models that cover the majority of the requirements, development, implementation, and maintenance phases of the software development life cycle.

The intent of the COCOMO II.2000 authors is the TEAM scale factor is to represent team cohesion throughout the software estimation life cycle. Therefore, this researcher disagrees with Software Estimation (1) panelist's suggestion that the definition of the TEAM scale factor, specifically the questions that are asked to determine the TEAM scale factor value are more related to the very early stages of the software development life cycle. Additionally, based upon Software Estimation (One) panelist's statement "The parameter is looking [more] at the system acquisition (stakeholders, etc.) than it is looking at the dynamics of the software development itself" there might be some confusion regarding the definition of stakeholders which primarily refers to the customer and the software development company executives, however, Boehm, Abts, Winsor-Brown, et al., 2000, made it very clear that stakeholders was all inclusive of customer representatives, company executives, project managers, developers, interfacers, etc. This researcher believes Software Estimation (1) was stating the questions used to determine the TEAM scale factor did not go into more detail regarding the team enviroment and dynamics, which this researcher and other panelists agree upon.

Only one panelist didn't think this research presented a complete representation of team cohesion but did not comment if the recent team cohesion research has shown the need to investigate the COCOMO II.2000 TEAM variable. In particular, this panelist states that "team cohesion is much more complex than what is articulated" in this research. This panelist's expertise is in the areas of organizational behavior, psychology, cognitive science and human resources. This researcher would be very interested in further investigating this panelist's concerns regarding the complexity of team cohesion and what areas of team cohesion were not presented in this research. Additionally, this researcher would ask the panelist if he thinks the COCOMO II.2000 TEAM measurement methodology represents a good team cohesion measure. This researcher believes he would say that the COCOMO II.2000 TEAM measurement methodology must be improved based upon his other answers. However, although this researcher does not doubt the panelist's argument, this researcher also believes the team cohesion research presented has shown the weakness of the COCOMO II.2000 team cohesion measurement methodology.

Additionally, this panelist has concerns regarding the link of EI to team cohesion and, more specifically, does not believe this research provides direct linkage to team cohesion. EI in this research has an indirect link, specifically; this research presented past research where team cohesion competencies were correlated to certain questions of the ECIv2.0 (Rapisarda, 2002b). EI is presented in the definition of terms within this research so the reader can have a layman's understanding of what EI is, how it is associated with the ECIv2.0, and how the Rapisarda (2002b) research was able to correlate team cohesion with certain questions of the ECIv2.0. This research's main focus is about determining if the team cohesion measure in COCOMO II.2000 needs to be improved upon based on the recent team cohesion research to date.

Finally, one panelist chose to not comment. As this panelist's area of expertise is associated with EI and team cohesion this researcher is very concerned about the non-response. Initially, this panelist was very reluctant to participate due to professional obligations; however, he did ultimately agree to participate. Therefore, this researcher believes this panelist did not have time to completely read through this research in order to answer this question to meet the first round deadline; especially based upon this panelist's later answers.

In conclusion, the panelists' confirmation regarding this researcher's premise that the COCOMO II's team cohesion model should be investigated based upon recent team cohesion research was very refreshing and hopeful toward affirming why some past software development projects may have been unsuccessful due to an error in the estimation and not necessarily due to mismanagement by the project manager.

Round 1-Question 2 responses.

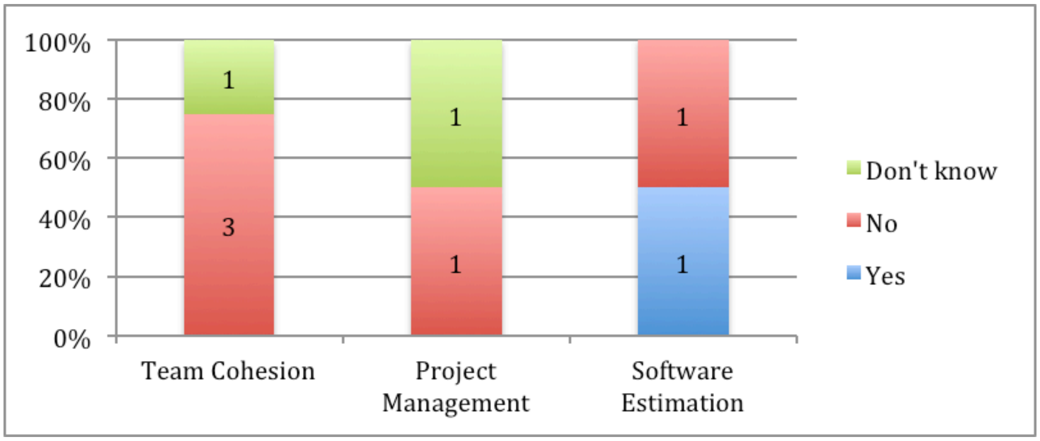

Figure 5. Round 1-Question 2 responses distributed by expertise

Table 9. Round 1-Question 2 responses by expertise

Round 1 Question 2	**In light of the team cohesion literature review within this research do you believe the COCOMO II.2000 TEAM variable effectively measures team cohesion? Why or Why not?**
Team Cohesion (1)	"No; the items in the COCOMO II.2000 do not measure team cohesion – especially as defined in the literature review. The items can, however, serve as an indicator of whether a team may get into problems in the future. For example, determining whether stakeholders' objectives are consistent – is an excellent early indicator of whether a team will run into conflicts or disagreements in the future. However, these items do not measure cohesion."
Team Cohesion (2)	"Don't know."
Team Cohesion (3)	"No, that is not clear to me. Need more details and some of what I read is beyond my expertise as an Organizational Psychologist."
Team Cohesion (4)	"No, I see nothing that assesses the extent to which the skill sets of the members of the team are compatible. In addition to appropriate technical skills (at all levels from junior to senior), a cohesive team also requires individuals to differ in soft skills like the ability to lead, the ability to work toward consensus, the ability to execute another's vision, the ability to see the big picture, the ability to see the details, etc. In addition, everyone needs appropriate communication skills."
Project Management (1)	"Don't Know."

Round 1 Question 2	**In light of the team cohesion literature review within this research do you believe the COCOMO II.2000 TEAM variable effectively measures team cohesion? Why or Why not?**
Project Management (2)	"No, I don't believe the paper stated it directly, but I think the COCOMO II.2000 version of TEAM slices off a strange piece of team cohesion. The factor might be more accurately titled 'Shared Vision' or 'Alignment on Objectives.' As mentioned in question one the subjectivity of the inputs is also a problem for accurate use of TEAM in COCOMO II."
Software Estimation (1)	No, "the team parameter may measure the cohesion at the acquisition level, but ignores the development that is included in the spiral model."
Software Estimation (2)	"Yes; TEAM is for ALL stakeholders, many of whom are NOT under the project management span of control."

Round 1 Question 2 is the very heart of what this research is attempting to prove; using a panel of experts to determine whether or not the COCOMO II.2000 TEAM scale factor effectively measures team cohesion as it is intended within the COCOMO II.2000 model.

Five of the eight expert panelists do not believe COCOMO II.2000 TEAM measures team cohesion and of those five three are team cohesion panelists. This researcher was very happy to have her supposition affirmed, however, two of the panelists did not comment and one panelist's response was ambiguous. As these panelists did participate more in rounds two and three this researcher can only presume they did not have time to read the literature review in order to meet the deadline to answer the Round 1 questions. Did their don't know responses harm this research? It certainly didn't help and leaves the reader and researcher to wonder if these panelists' fully participated in this round. Furthermore, Team Cohesion (3)'s "No, that is not clear to me" response is ambiguous. Is this panelist stating the presentation of team cohesion within this research compared to the COCOMO II.2000 team cohesion measure is not clear or that the COCOMO II.2000 Team methodology used to measure team cohesion is not measuring team cohesion? This researcher would like to gain clarification of this response and believes this response is equivalent to a non-response.

Of the five responses that have supporting arguments only one of the five affirms that the COCOMO II.2000 model effectively measures team cohesion. Unfortunately, his response of "Yes; TEAM is for ALL stakeholders, many of whom are NOT under the project management span of control" raises questions. This researcher is unsure how or what the project manager's span of control has to do with effectively measuring team cohesion. Team cohesion is what it is, regardless of the project manager's span of control. In light of the recent team cohesion research, the members of a team should all participate in the measurement of team cohesion. It appears the panelist is arguing the best the COCOMO II.2000 model could do toward measuring team cohesion is to use the project manager's subjective opinion because the project manager could not enforce all stakeholder's to participate in a team cohesion measurement survey. If that is the argument, then, the COCOMO II.2000 model should state the team cohesion measure is subjective and, therefore, highly prone to error.

Additionally, an argument could be made to completely remove the team cohesion scale factor from the COCOMO II.2000 model since this research has presented how error prone the subjective opinion of one person can be. However, this researcher believes, if quality software estimation is the goal of the COCOMO II.2000 model, the TEAM scale factor should remain and must be modified to be an effective measure of team cohesion. Furthermore, this researcher agrees that all stakeholders should be considered part of the team, that is, all individuals who have a deciding factor in how the project is to be completed should be considered stakeholders.

With regard to the other four panelists' responses, they all agree team cohesion is not effectively measured within the COCOMO II.2000 model based upon the recent team cohesion research. The Software Estimation (1) panelist reaffirmed his previous argument that the COCOMO II.2000 TEAM scale factor only considers the acquisition phase of the software development life cycle. This response possibly infers there should be two COCOMO II.2000 TEAM scale factors. One for the acquisition phase, which would be included in the first of the three COCOMO II.2000 sub-models, and one team cohesion measure that would be used in the last two COCOMO II.2000 sub-models. Remember, each of the three models increases in fidelity as the project becomes more defined. However, even with two separate TEAM scale factors, they would still need to be reinvestigated to truly represent team cohesion in that particular phase. Additionally, it should be noted that the first sub model within COCOMO II, the application composition model, does not contain a TEAM scale factor.

The Project Management (2) panelist presented a very interesting observation in that he believes the COCOMO II.2000 team cohesion measurement methodology seems to "[slice] off a strange piece of team cohesion" and that the measurement would be more "accurately titled 'Shared Vision' or 'Alignment on Objectives.'" This panelist's response is in alignment with Software Estimation (1)'s response mentioned in the previous paragraph. That is, it would appear to these two that the COCOMO II.2000 team cohesion methodology does not measure team cohesion as a whole but only measures a portion of team cohesion. This researcher is interpreting Project Manager (2)'s response as suggesting the COCOMO II.2000 team cohesion methodology is weak and not that there should be two additional measures to the COCOMO II.2000 model.

However, as discussed earlier, the possibility of having different team cohesion measures to better suit each phase of the software development life cycle is certainly an idea. Further on a panelist even suggests researching how the software development life cycle changes team cohesion to find a more appropriate means of measuring team cohesion within software development.

Team Cohesion (1) and Team Cohesion (4) are in complete agreement that the COCOMO II.2000 TEAM scale factor measurement methodology does not measure team cohesion. Although Team Cohesion (1) suggests the current measurement methodology for the TEAM scale factor would be a good indicator in determining if the stakeholders' could face collaborative difficulty, Team Cohesion (4) provides a more detailed argument that there aren't any items within the TEAM scale factor measurement methodology that "assesses the extent to which the skill sets of the members of the team are compatible." She further argued there should be a means to determine if individual team members have good communication skills and if team members differ in skills such as leadership, "the ability to work toward consensus, the ability to execute another's vision, the ability to see the big picture, the ability to see the details, etc." Research has been Rapisarda (2002b) and Pillis and Furumo (2006) presented research that argues the level of commitment and collaborative behavior each team member exhibits is correlated with team cohesion (Pillis & Furumo, 2006; Rapisarda, 2002b). Of further interest Rapisarda's (2002b) research describes the emotional intelligence competencies correlated with team cohesion which include the desired items Team Cohesion (4) argues are not in the COCOMO II.2000 TEAM scale factor methodology. Further research should be accomplished toward using emotional intelligence and task cohesion to develop a more thorough measure of team cohesion to be used within the COCOMO II.2000 estimation model.

Round 1-Question 3 responses.

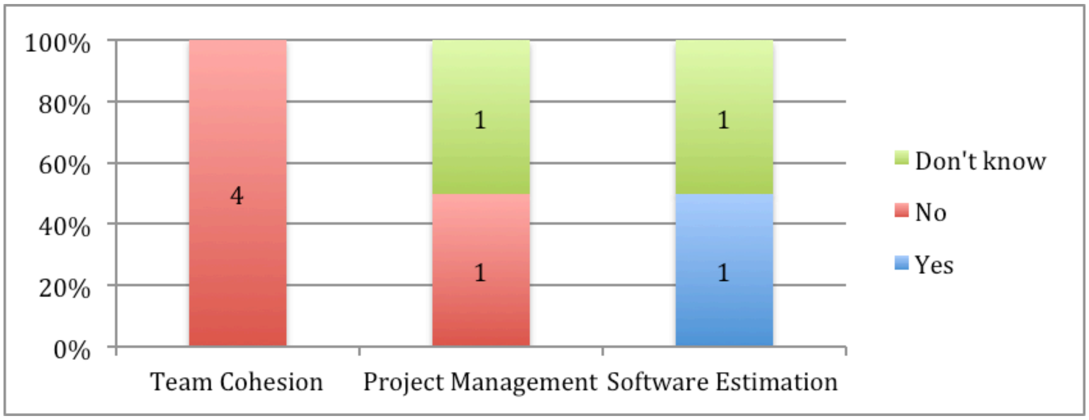

Figure 6. Round 1-Question 3 responses distributed by expertise

Table 10. Round 1-Question 3 responses listed by expertise

Round 1 Question 3	Do you believe the project manager's subjective assessment of the software development team is an effective method to measure team cohesion? Why or Why not?
Team Cohesion (1)	"No. I do not believe that the project manager is the best person to answer these questions unless he or she goes out and collects information from stakeholders to answer the questions. He or she needs to get honest answers from the stakeholders, especially for questions 1, which is about consistent objectives and 2, which asks whether stakeholders are willing to accommodate one another's needs."
Team Cohesion (2)	"No, self-assessment of anyone on behavior or real human behavior has been shown to be delusional since the 1950's. They are filled with social desirability and the individual's value dispositions. The best way to assess any characteristic of a team is to get the shared views among the team members, AND a boss's view—but treat them as two variables."
Team Cohesion (3)	"No, it takes a team to evaluate a team. I prefer direct observation with precise protocols that guide the observations."
Team Cohesion (4)	"No. If you don't spend your days in the trenches, the subtleties of how the team members interact will be lost. How we behave in meetings and when management is around differs from how we behave in the trenches."

Round 1 Question 3	**Do you believe the project manager's subjective assessment of the software development team is an effective method to measure team cohesion? Why or Why not?**
Project Management (1)	"This all depends on the project manager. If the model depends on a subjective assessment of a person, the model will only be as good as the person using it. Therefore, the model should have a range of estimates based on the expertise of the project manager. Inexperienced and relatively poor-performing project managers should provide estimates like Estimate Length of Project: six months +/- 50% while expert project managers, ones with good track records can issue estimates with smaller ranges."
Project Management (2)	No; see previous answers.
Software Estimation (1)	"Possibly. The manager's assessment is colored by the manager's culture. There is a big difference between classical project management and project leadership. I suggest looking at the first chapters of Hershey and Blanchard's text *Management of Organizational Behavior* and the difference between Theory X and Theory Y management."
Software Estimation (2)	Yes, see previous answer (#2)

Five of the eight panelists emphatically agree the subjective measure of the project manager is an inadequate and problematic means of measuring team cohesion and state the same argument this researcher has presented. Additionally, airing on the side of caution, this researcher interpreted two of the responses as don't know; however, these responses could very easily be interpreted as no answers, which would provide an overwhelming majority of the panelists in agreement that the subjective opinion of the project manager is an inadequate means of measuring team cohesion. To further the argument of counting the two panelists' responses as no responses, neither panelist provided an affirmative argument. As the reader can see, the reason this researcher interpreted them as don't know was due to Software Estimation (1)'s written response of "possibly" and Project Manager (2)'s response of "it depends." However, both panelists' argued the subjective measure of a project manager would be colored based on the project manager's culture and expertise in successfully measuring projects.

As this researcher has argued there are times when a project manager does not have the history or continual observation with the team under measurement as they may not work with them directly day in and day out. The benefits of a tool that could measure team cohesion from the team's perspective would provide a more comprehensive team cohesion measure. The ECIv2.0 model uses a 360-degree measurement technique that gathers team members' responses from within and responses from team members' colleagues from without. That is, each team member assesses himself and four other team members. Additionally, each team member selects four external colleagues to assess his team cohesion measure. All surveys are used to determine the team's cohesion measure. In this manner, a fairly well rounded presentation is provided for each team member. The ability to work professionally and collaboratively in meeting the task at hand and ultimately reach the desired goal of a completed project on time, on target, and on budget is the outcome sought and the thing that is to be measured and that thing is team cohesion.

Software Estimation (2) had the only affirmative response regarding the effectiveness of the project manager's subjective measure of team cohesion and points to his previous answer to Round 1 question two where he stated the team cohesion measure is for all stakeholders many of which are not under project management control. Although there will be stakeholders, also known as team members, who will not be under the control of the project manager; e.g. customer representatives, a good project manager should be able to convince all stakeholders of the importance and benefit of measuring team cohesion that will improve the goal of delivering a project delivered on-time, on-target, and on-budget. However, if the customer representatives do not participate, a project manager can provide a zero team cohesion value with regard to the customer representatives' team cohesion measure and assess the estimation accordingly. Additionally, knowing this to be a very real issue, the COCOMO II.2000 TEAM methodology should provide a means for incorporating a delta in estimating team cohesion. That is an offset variable that would present a percentage of team members that did participate within the measurement. This could be used in place of assessing zeros for non-participating members or could be used inclusive of the zero team cohesion values. Regardless, a means must be created to assess those non-participating members accurately within the team cohesion measure.

Round 1-Question 4 responses.

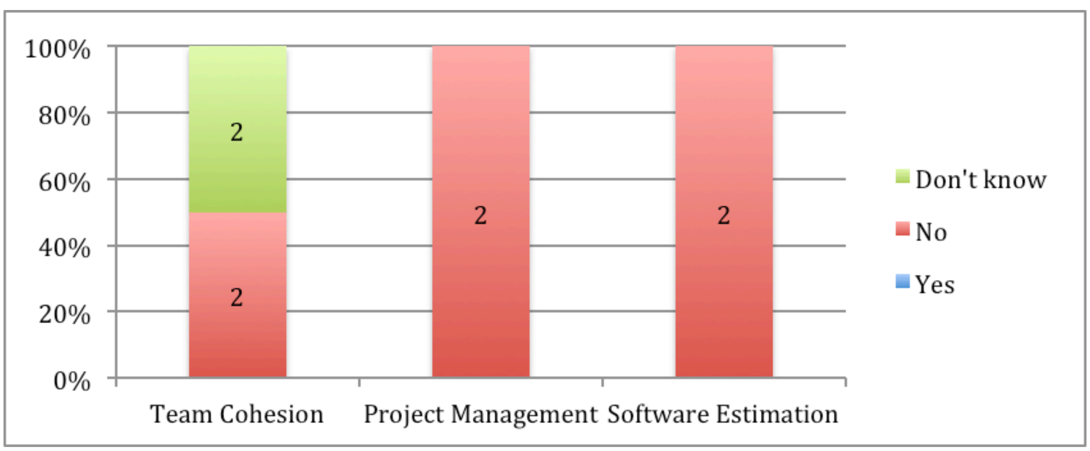

Figure 7. Round 1 – Question 4 responses distributed by expertise

Table 11. Round 1 – Question 4 responses by expertise

Round 1 Question 4	Do you believe the four survey questions used to determine the COCOMO II.2000 TEAM variable's value effectively assesses a software development team's team cohesion measurement? Why or Why not?
Team Cohesion (1)	"No, because these items do not measure cohesion. The items are likely a good indicator of whether a team would be more or less likely to develop cohesion, but they do not measure cohesion as you have defined it in your literature review. However, to be frank, I am not sure why you believe cohesion is the best construct to measure if you are trying to predict the effectiveness of a software development team. I'd be more interested in shared goals (which is asked in the COCOMO II.2000), the quality of communication among stakeholders, and perhaps some other things."
Team Cohesion (2)	Don't know. Four questions test the "lower limit of reliability and factor models. Often, 6+ items are needed for a stable measure."
Team Cohesion (3)	"No, four questions are not enough."
Team Cohesion (4)	"It is at best a 'swag' for the reasons I stated in previous questions. Also, some of those things should have been addressed in building the team. For example, why would you put someone on the team if they aren't willing to accommodate the needs of their teammates? And, what project manager, who wants to seem competent, would rate the consistency of their objectives and cultures as "very low" even if they are? Isn't it their job to set that consistent vision? Since the COCOMO data is generally shared with upper management, there is risk that the project manager may not see or report the reality for this question."
Project Management (1)	"No. I don't understand how four questions with six possible answers can evaluate something as complex as the interaction of a group of people."
Project Management (2)	"No. I think they measure an interesting attribute of software projects, but as I stated in #2, I think it's more like 'Alignment on Objectives' than team cohesion. People can be aligned on objectives and still dislike each other at the personality level, leading to friction and low productivity. TEAM doesn't account for that."
Software Estimation (1)	"From the point of view of team cohesion at the development level, NO."
Software Estimation (2)	"No, but it is not just about the "development team" as commonly used, i.e., the team of software developers (presumably selectable resources under the project manager's span of control)."

Within in this round six of the eight panelists directly affirmed and two indirectly affirmed the four survey questions do not adequately measure team cohesion based upon the team cohesion research presented. As with the interpretation of Round 1 question three, the two don't know responses within this round can easily be interpreted as a no response as each panelist did not provide an affirmative argument that the four questions effectively assess team cohesion. Additionally, Team Cohesion (1), Team Cohesion (4), Project Management (2), and Software Estimation (2) all provided very interesting responses.

Team Cohesion (1) stated, "not sure why you believe cohesion is the best construct…to predict the effectiveness of a software development team. I'd be more interested in shared goals...[and] the quality of communication among stakeholders" This researcher believes the questions of the GEQ and ECIv2.0 team cohesion models answer the above concerns, specifically questions within the Adaptability, Empathy, Conflict Management and Teamwork and Collaboration sections of the ECIV2.0 and questions 4, 10, 12, and 14 within the GEQ. These questions either directly or indirectly address the concerns of shared goals and quality of communication. Additionally, as this research has mentioned, software development projects have a very high failure rate and the ability of a software development team to solve complex tasks collaboratively is a major determining factor toward project success (Chiang & Mookerjee, 2004; Kang, Yang, & Rowley, 2006; Mills, 1999). Furthermore, team cohesion has been deemed the most important small group variable in measuring group formation, maintenance, and productivity (Bollen & Hoyel, 1990). Finally, most software projects fail due to the imprecision and inaccuracy of the project plan (Humphrey, 2005). Therefore, the ability to successfully measure team cohesion allows the project manager to better estimate the time needed to meet project milestones.

Team Cohesion (4) stated “what project manager, who wants to seem competent, would rate the consistency of their objectives and cultures as ‘very low’ even if they are?” This concern is precisely why team cohesion should include the entire team members within its measurement process, very similar to 360-degree evaluations. The project manager’s goal is to layout a successful project plan (McConnell, 2006). Additionally, as team cohesion changes over time (Wellington, Briggs & Girard, 2005) the project manager will need to continually assess the team and its progress to adjust the schedule as necessary. Estimating a software schedule is an art and a science. A project manager who has many years experience as a software engineer will use their past experience and knowledge of the company along with estimation measures such as team cohesion to determine the project milestones and how much buffer should be given to prepare for those inevitable unexpected situations that arise when developing software. Furthermore, historical metrics of past development also assist the project manager in planning the milestone dates. Team Cohesion (4)’s concerns are a good part of the reason why this research has been undertaken.

Project Management (2) stated “people can be aligned on objectives and still dislike each other at the personality level, leading to friction and low productivity. TEAM doesn’t account for that.” This researcher has cited many of Project Management (2)’s published material in this research and his statement is precisely why this research will open the door to better estimation practices. Project managers must look at all variables in the software development life cycle in order to accurately estimate a project plan. Additionally, it has been said team members do not have to socialize together outside of work to be successful; the key is that they respect each other’s professionalism and ability to allow the project goals to be met (Knouse, 2007). If one team member does not respect another team member's ability or professionalism to meet the project goals there will be conflict.

According to Software Estimation (2)'s response, it would appear that he does not understand this research's definition of team also includes the stakeholders e.g., those outside of the realm of the software development group, as part of the 'development team'. Apparently, this researcher needs to make the team definition within this research clearer. This research is not primarily concerned with measuring only developers, as they are only part of the software development team; this research is interested in measuring all stakeholders who participate in the development of the software product.

The other panelists all expressed concerns that the type and amount of questions used for TEAM are not adequate enough to meet the definition of team cohesion presented in this research and agree there should be more than four questions and all team members should participate in order to present an accurate team cohesion measure. Overall this researcher is very satisfied with the panelist's responses to this question.

Round 1-Question 5 responses.

Figure 8. Round 1-Question 5 responses distributed by expertise

Table 12. Round 1-Question 5 responses by expertise

Round 1 Question 5	Based upon your expertise and the literature review of this study does the COCOMO II.2000 TEAM variable take into consideration how team cohesion changes over the life cycle of the project? Why or Why not?
Team Cohesion (1)	"No, because it measures cohesion at one point in time. I fully agree that a measure like this should be used several times over the life of a long project. One snapshot, especially if taken at the beginning of a project, will not measure changes over the life cycle of a project. Questions 1, 2, and 4 in the COCOMO II.2000 might be good questions to ask multiple times. But, they do not measure team cohesion."
Team Cohesion (2)	"The nature of the items would determine whether the measure is sensitive to trait or state level characteristics. We know from the psychological literature, as well as physiological assessments, that state items are different in phrasing from trait items and it is really important to know and test the difference. Otherwise, the lack of change would be misinterpreted."
Team Cohesion (3)	"Not sure…theoretically it might show it but needs to be tested moment-to-moment as the team performs."
Team Cohesion (4)	"If the model was re-run, certainly new values could be entered. However, how often is that actually done? And, as the project progresses, my earlier concerns about the project manager not wanting to see/report really would increase."
Project Management (1)	"I cannot tell. If a project manager reassesses the TEAM measure every time he reassesses the project schedule, cost, etc. the TEAM measure will be more helpful. If the TEAM measure is used only once, it isn't of much value."
Project Management (2)	"It depends on whether the project is [re-estimated] using COCOMO II. The answers to the questions will presumably become more accurate as the estimator is able to see how well the group of stake holders actually works together."
Software Estimation (1)	"No. An evaluation of the acquisition team does not reflect significantly on the development team performance."
Software Estimation (2)	"No, but COCOMO II.2000 was never intended to look at 'team cohesion changes over the life cycle of the project'."

All but three panelists were unsure if the COCOMO II.2000 TEAM variable takes into consideration how team cohesion changes over the life cycle of the project. As this research has presented evidence that a team's cohesion changes over time (Wellington, Briggs & Girard, 2005) a software estimation model should attempt to accurately represent the variables of the software development process and should have a method to measure the change of the software team's cohesion over the life cycle of the project. Additionally, as stated earlier, the COCOMO II.2000 model contains three phases that are to represent the software development life cycle: specifically, three sub-models, each one increasing in fidelity, are (a) the Applications Composition, (b) Early Design, and (c) Post-Architecture (Musilek, Pedrycz, Sun, & Succi, 2002). Both the Early Design and Post-Architecture phases use the TEAM variable and both phases are where the majority of the development and implementation of the software product take place. Furthermore, according to authors of COCOMO II.2000, one of the primary objectives is to provide accurate cost and schedule estimates for both current and likely future software projects.

Therefore, this researcher is baffled at Software Estimation (1)'s response that COCOMO II.2000 "was never intended to look at team cohesion changes over the life cycle of the project" especially when each of the three COCOMO II.2000 phase increase in fidelity over the lifecycle of the project. Additionally, Software Estimation (2)'s response "An evaluation of the acquisition team does not reflect significantly on the development team performance" has this researcher confused. Presumably Software Estimation (2) believes the TEAM variable is to estimate only the acquisition team, however, the COCOMO II.2000 manual specifies the model is intended for use throughout the software life cycle to allow for a more comprehensive estimate as each of the three COCOMO II.2000 phases are entered. Therefore, Software Estimation (1)'s argument that team was never intended to look at cohesion changes over the life cycle of the project is incorrect and Software Estimation (2)'s response needs more clarification as to why he believes only the acquisition team is being evaluated.

Team Cohesion (1) argues that one measurement in time is not adequate enough to accurately represent team cohesion for the project and further states the team should be measured multiple times throughout the project life cycle. With regard to the other panelists' responses they are all unsure if the TEAM model takes into consideration the team cohesion changes over time but do agree if the model was run subsequently through out the project life cycle this could attempt to address the team cohesion changes. The important take away here is the need to re-measure at a minimum within each of the last two phases of the COCOMO II.2000 model as team cohesion does change over time. Apparently, this researcher did not provide clear enough instruction as to who the TEAM variable is measured against and did not present that the COCOMO II.2000 model does not provide any inference of re-measuring the TEAM variable or that some type of offset is provided to allow for the plus or minus of team cohesion change. Overall, this researcher is partially satisfied with the panelists' answers. Specifically, there is more uncertainty than certainty in the panelists' responses and of those definitive responses the arguments are questionable. This researcher recognizes the question could have been better defined and the confusion might have been avoided if who is to be measured were more clearly defined in the question and if COCOMO II.2000 directions regarding the use of the TEAM variable over the lifecycle of the software development were better articulated in this question.

Round 1-Question 6 responses.

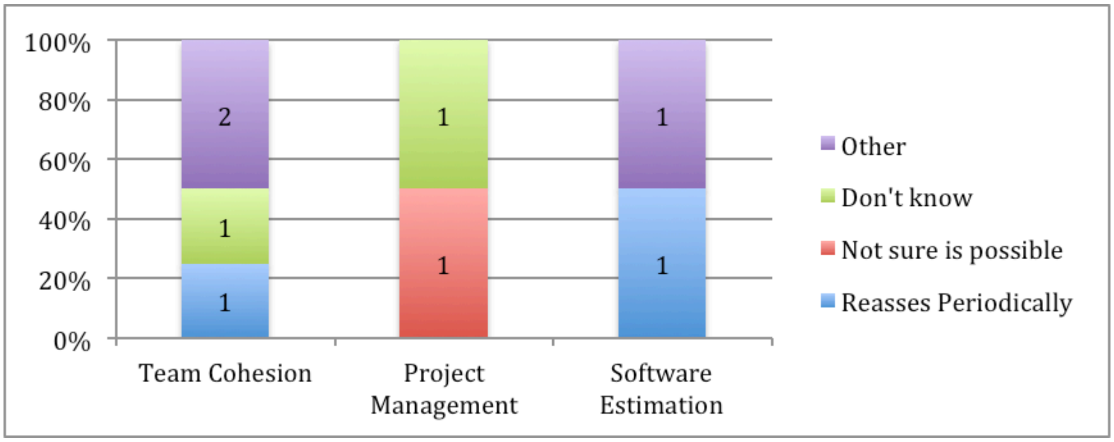

Figure 9. Round 1-Question 6 responses distributed by expertise

Table 13. Round 1-Question 6 responses by expertise

Round 1 Question 6	What suggestions, if any, would be the best way to introduce the dynamic phenomena of team cohesion changing over the life-cycle of the project into the current COCOMO II.2000 model? Why or Why not?
Team Cohesion (1)	My suggestion would be to measure it periodically. Richard Hackman and Ruth Wagemen have done research (*The Academy of Management Review 2005)* on when coaching is useful in teams – what I remember of the research is that the launch point, the mid-point, and the ¾ point of a team's time together are critical points for assessing team dynamics and taking action.
Team Cohesion (2)	"Delicately, and I usually recommend serving a good red wine – it helps."
Team Cohesion (3)	"See model by Marks et al in Academy of Management Review 2001."
Team Cohesion (4)	No answer
Project Management (1)	"See previous answer."
Project Management (2)	"I'm not sure [it is] possible. The paper cites *Peopleware*, but it doesn't cite one of *Peopleware*'s most sobering propositions, which is that it is impossible to tell exactly what will make a team gel. Some teams just seem to gel, and some don't."
Software Estimation (1)	"I would focus on the ACAP and PCAP ratings which are really driven by the organization management and environment. Capability is much broader than IQ and problem solving skills. The dynamics of the team are much more significant drivers than intelligence. What are the personnel differences between the traditional cube farm organizations and the highly productive agile organizations? There aren't that many differences in personnel, yet productivity is much greater."

Round 1 Question 6	**What suggestions, if any, would be the best way to introduce the dynamic phenomena of team cohesion changing over the life-cycle of the project into the current COCOMO II.2000 model? Why or Why not?**
Software Estimation (2)	"Re-assess the four questions for ALL stakeholders periodically."

As was expected, there are varied answers to this question as the question is considered an open question with the only guidance of introducing the phenomena of team cohesion change over the lifecycle of the project. This researcher interpreted the non-response of Team Cohesion (4) and Project Manager (1) responses as don't know responses. Team Cohesion (1) and Software Estimation (2) state team cohesion should be measured periodically. Software Estimation (2) suggests reassessing the TEAM questions for all stakeholders periodically. However, this researcher does not believe the four TEAM questions are enough to adequately portray team cohesion and the other panelists support this assessment via their responses to the previous questions. Team Cohesion (1) suggests measurement take place at the beginning, midpoint, and three quarter marks of the project, however, the individual research she cites suggests the beginning, midpoint, and endpoint (Hackman & Wagemen, 2005) times to intervene in order to address expected team dynamic situations in the context team coaching. Regardless, the suggestion is clear that a team needs to have their team cohesion measured periodically. Project Management (2) states he does not believe it is possible to measure the dynamic phenomena of team cohesion over the lifecycle of a project and suggests that the authors of *Peopleware* stated, "it is impossible to tell exactly what will make a team gel." Although Demarco and Lister stated, "You can't make teams jell" (1999, p. 132) they didn't say it is impossible to tell exactly what makes a team jell. Demarco and Lister describe characteristics of a jelled team as having low turnover, a strong sense of identity, a sense of eliteness, joint ownership of the product, and the obvious enjoyment people take in their work (p. 127).

Team Cohesion (2)'s response was very surprising and a little disappointing as this researcher was really hoping for further avenues of research to improve team cohesion. However, the response did give this researcher a chuckle, and, after some thought, this researcher agrees that a good glass of red wine can make most things enjoyable. With regard to the "delicately" suggestion, this researcher agrees that presenting a model that measures the dynamic phenomena of team cohesion changing over time should be done cautiously to ensure all phenomena variables are considered.

With regard to Software Estimation (1)'s suggestion to focus on the ACAP and PCAP ratings, this researcher thought an explanation of the ACAP and PCAP ratings would be helpful for the reader to understand this panelist's response. Within COCOMO II.2000 model there are cost drivers. These cost drivers are a culmination of scale factors and effort multipliers. Furthermore, these values are the subjective opinion of the project manager.

$$PM = 2.94 \underbrace{\prod_{i=1}^{17} EM_i}_{\text{Product of 17 Effort Multipliers}} S^{0.91+0.01 \overbrace{\sum_{i=1}^{5} W_i}^{\text{Sum of 5 Scale Factors}}}$$

Figure 10. Post Architecture Model Equation. From “On the Sensitivity of COCOMO II.2000 Software Cost Estimation Model” by P. Musilek, W. Pedrycz, N. Sun, & G. Succi, 2002. *Proceedings of the Eight IEEE Symposium on Software Metrics (METRICS’02), Edmonton, Alberta, Canada,* p. 363-379. Copyright by IEEE. Adapted with permission.

Specifically, 17 effort multipliers and five scale factors (Table 6) are used within the Post Architecture model of COCOMO II.2000, which is the third and final phase of the estimation model. This research focused on the TEAM scale factor in the Post Architecture Model. The mathematical representation of the COCOMO II.2000 Post Architecture model is presented in Figure 12. The Post Architecture model equation states the number of Person months (PM) it will take to create a project of size (S) is equal to the COCOMOII.2000 productivity constant 2.94 times the product of the 17 Effort Multipliers times the size. The size is equal to the size of the software unit of measure, e.g. 1000 SLOC, to the power of 0.91, the COCOMO II.2000 scaling base-exponent for the effort equation, plus the product of 0.01 and the sum of the five scale factors. TEAM is one of the five scale factors. The Analyst Capability (ACAP) and the Programmer Capability (PCAP) values represent the capability of the analysts and the capability of the programmers respectively. Analysts are considered to be personnel who work on requirements, high-level design, and detailed design. The PCAP represents the capability of the programmers as a team, that is, their programming ability as a team and their ability to communicate and cooperate as a team, not the individual programmers' experience (Boehm, Abts, Winsor, et al., 2000, p. 47). The panelists were asked to provide suggestions for the best way to introduce the dynamic phenomena of team cohesion changing over the lifecycle of the project in the COCOMO II.2000 model. Unfortunately, the ACAP and PCAP are only used in the final phase of the model and, therefore, would not represent the delta of team cohesion over the entire life cycle of the project. However, the fact that the COCOMO II.2000 Post Architecture model indirectly considers a facet of team cohesion within the PCAP variable provides a foothold to introduce a team cohesion variable throughout the three phases of the COCOMO II.2000 estimation model. This researcher believes team cohesion should be considered in all three phases of the COCOMO II.2000 model as team cohesion has been proved to change over the life cycle of the project. Additionally, regardless of who make up the members, the unit is still a team, and the cohesion measurement should be of the team in each phase. This researcher suggests the COCOMO II.2000 model authors consider implementing a team cohesion variable in each phase of the three phases within the model.

Of additional interest is Software Estimation (1)'s argument that the PCAP and ACAP variables limit the scope of capability through the anemic measurement of only IQ and problem solving skills. He also argues the open office area is more conducive to productivity rather than the restricted structural work environment of a cube farm. Although this research is primarily focused on team cohesion and how it effects software development, this research has briefly touched on how using an IQ measurement alone to estimate an individual's productivity has proven to be a fallacy in the current team work environment. Additionally, with regards to the cube farm vs. the open work area, this researcher would be interested in new research on how team cohesion works within the cube farm vs. the open work area. With regard to Team Cohesion (3)'s cited research suggestion, Marks, Mattieu, and Zacarro (2001) discuss a dynamic model of team effectiveness; a taxonomic structure where processes are nested within transition and action phases. Specifically, team processes "members' interdependent acts that convert inputs to outcomes through cognitive, verbal, and behavioral activities directed toward organizing taskwork to achieve collective goals" (2001, p 357). Marks, et al., view cohesion as an emergent state rather than a process because processes describe the nature of team member interaction where as emergent states do not and have mutable qualities. To clarify this explanation an example is provided where "teams with low cohesion (an emergent state) may be less willing to manage existing conflict (the process), which, in turn, may create additional conflict that lowers cohesion levels even further" (2001, 358). However, Marks, et al., do not discuss how or when they measure team cohesion. Although, this researcher found the research interesting and did find further evidence that dynamic variables must be measured periodically, this researcher does not believe the suggested research provides any further benefit or specificity on how to measure the dynamic phenomena of team cohesion.

Round 1-Question 7 responses.

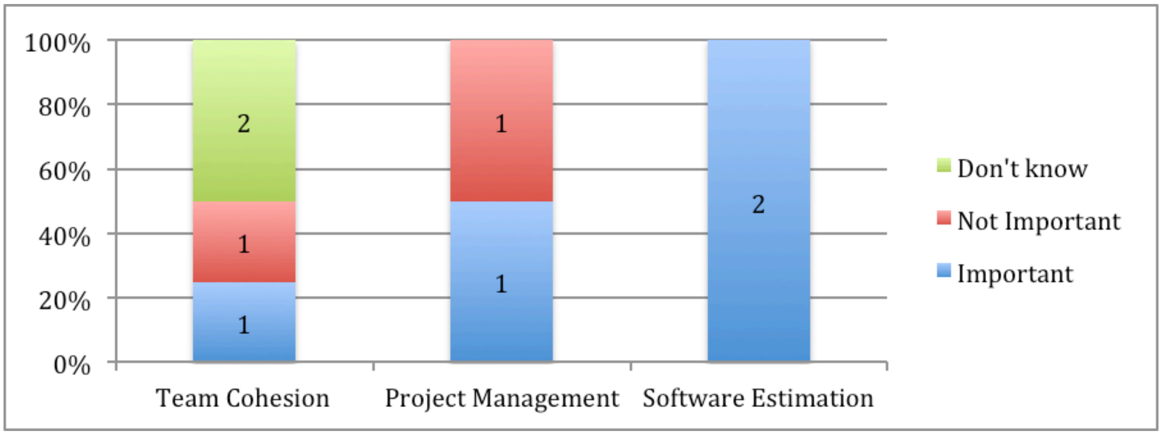

Figure 11. Round 1-Question 7 responses distributed by expertise

Table 14. Round 1-Question 7 responses by expertise

Round 1 Question 7	In your expert guidance how important is task cohesion as a factor of success in software development? Why or Why not?
Team Cohesion (1)	"This is an important question – I believe that cohesion is an indicator of whether the team members like each other and like the task. If they do – they have the propensity to do well together. But, there isn't much you can do about cohesion. I'd rather look at relevant constructs that account for more variance in project effectiveness – for example, getting a clear, shared vision. Making sure members understand each other's perspectives, needs, and areas of expertise – to me these issues/constructs are more closely linked to project effectiveness. I do believe cohesion matters in that it sets a team up well for doing these things."
Team Cohesion (2)	"According to doctoral dissertations of several of my doctoral students, shared vision is the single greatest variable in predicting team effectiveness. I also predict that a team's shared identity is a major factor. Team cohesiveness, in any of the ways it has been measured since the mid-1960's, would follow as a consequence of these. I remember George Ferris summarizing the team cohesiveness literature in 1969 (which I used in some of my classes at Harvard in those years) was: "Teams that win feel good, but teams that feel good don't always win."
Team Cohesion (3)	"I think there are other constructs of importance--mutual support, information exchange, and others."
Team Cohesion (4)	"I assume you mean team cohesion. I think it is CRITICAL! If the people aren't working together, the pieces they build won't work together either."
Project Management (1)	"Task cohesion? Not important. A group of people wanting to work together to achieve a shared goal? Extremely important."

Round 1 Question 7	In your expert guidance how important is task cohesion as a factor of success in software development? Why or Why not?
Project Management (2)	“I think it’s very [important], but I think it’s impossible to estimate. In general, I think COCOMO II.2000 is impossible to use effectively because it contains so many subjective inputs. The TEAM adjustment is not the exception to the rule, no matter how it’s defined. Although I don’t like COCOMO II.2000 for estimating, I think it does provide many useful insights into the magnitude of various factors on software project outcomes. It’s probably useful to try to establish the magnitude of the effect of ECIv2.0 and GEQ on software project outcomes even if they aren’t useful for estimation purposes. They could be very useful for project managers to understand the leverage that fostering team cohesion can have.”
Software Estimation (1)	“Very important. Task cohesion is a function of several variables. First, cohesion does not take place in a non-collaborative environment. Cubicles guarantee a lack of cohesion. Second, traditional management does not foster leadership and the motivation that fuels cohesion. Agile methods are derided frequently by organizations that are based on traditional approaches to development, and productivity has remained stagnant in those organizations. The greatest driver in capability is motivation, which requires leadership and a collaborative environment. One of the greatest, and least documented, organizations in the history of engineering is the infamous Lockheed Skunk Works that produced the P80 jet fighter (first one) in only 247 days. The main features of the team were motivation, collaboration, and an open environment.”
Software Estimation (2)	“Very important; as defined in Boehm et al., it is an economy of scale parameter.”

The researcher was expecting more from the responses of the panelists than what was provided. However, the researcher does recognize the question could have been better clarified; specifically, the question should have been presented in the following manner: Team cohesion is defined as having two subcategories, socially based and task based. Socially based cohesion “refers to a positive attitude toward developing and maintaining interpersonal relationships within a team” and task based cohesion refers to a “task oriented belief about achieving the team’s goals through commitment to the team approach (Rainey & Schweickert, 1988)” (Salas, Burke, Fowlkes, & Priest, 2004, p. 438). With regard to measuring team cohesion, in your expert opinion, do you believe measuring task cohesion alone would provide a better measure of team cohesion or measuring both social and task cohesion for a better measure of team cohesion? Why or Why not? This researcher believes this previous question would have produced the desired results. Perhaps another survey can be accomplished in the future where questions are further clarified.

The subcategories of team cohesion were discussed within this research and to aid the reader, the points made earlier will be repeated here. Team cohesion has two subcategories: (a) socially based and (b) task based. Socially based concerns the development and maintenance of interpersonal relationships within a team and task based cohesion is concerned with the achievement of team goals through commitment to the team (Salas, Burke, Fowlkes, & Priest, 2004). Task cohesion was determined to be the better predictor of work-group performance (Carless & De Paola, 2000) and cohesion is developed as a result of the task at hand and not social interaction among team members (Knouse, 2007).

Team Cohesion (1) and Team Cohesion (2) responses did not specify the importance of task cohesion. This researcher chose to air on the side of caution in interpreting their response as don't know, however, their responses could easily be interpreted as stating task cohesion is not important. Team Cohesion (1), Team Cohesion (2), and Team Cohesion (3) argue shared vision and information exchange would be the better predictors of team productivity rather than team cohesion. Certainly if the project manager does not provide a vision and gain the commitment of the team to that vision there will be great difficulty in reaching the desired end goal. In fact, Team Cohesion (1) argues, "there isn't much you can do about cohesion," however, this research is not interested in modifying cohesion but in ascertaining the state of cohesiveness within the team; that is, how strong is the professional desire of the team member toward the team to succeed in meeting the end goal. Some individuals are more tolerant than others and are able to guide the less experienced team members toward the shared vision of successfully achieving the end goal of a project on-time, on-target, and on- budget. This researcher agrees that shared vision and information exchange are important and a wise project manager will continue to present and remind the team of the shared vision throughout the project lifecycle in order to keep them focused. But how the individuals respect one another, as a professional with regard to achieving the end result of a shared vision is the measurement of interest. If team members do not fully collaborate in a productive manner, e.g., withhold opinions, experience, guidance, or knowledge the project will take that much longer to complete.

Team Cohesion (4), Project Manager (Two), Software Estimation (1), and Software Estimation (2) all believe team cohesion is very important. Team Cohesion (4) responds to team cohesion believing the researcher mislabeled the question, thus the need for a more clarified question. This researcher believes most of the responses are toward the team cohesion and not the team cohesion sub-category task cohesion based upon the responses of Team Cohesion (4), Project Manager (1), and Team Cohesion (2). Project Manager (1) does not believe task cohesion is important but his argument essentially defines what task cohesion is and Team Cohesion (2) specifies team cohesion in his response rather than task cohesion. However, the importance of team cohesion can be gleaned from these responses.

Project Manager (2) and Software Estimation (1) provided interesting supporting arguments. Project Manager (2) does not believe team cohesion can be measured and does not trust COCOMO II.2000 as a software estimation tool due to so many subjective measures. However, he does suggest the use of the ECIv2.0 and GEQ to aid the project manager toward understanding how important fostering team cohesion can improve the productivity of the team. Software Estimation (1) argues that it is the lack of leadership's presentation of shared vision and motivation that leads to less team cohesion along with an enclosed work environment such as a cube farm which is inline with many of the other panelist's assessments.

Of key importance for the reader to remember is that this researcher is attempting to discover how software projects are failing in such large numbers and believes the estimation tools play a part in these failures; specifically, the subjective measure of team cohesion in the COCOMO II.2000 model. This research is to gain expert opinion on whether further research is needed to ascertain if the COCOMO II.2000 TEAM measure accurately portrays team cohesion as it is known today. Additionally, this researcher has presented research that shows how the level of team cohesion affects the productivity of a software development team. Furthermore, the goal of the Round 1 questions was to elicit expert opinion whether this researcher's premise, that the COCOMO II.2000 TEAM measurement is inadequate and warrants future research. Most of the expert panelist's agree team cohesion is important to the successful estimation of a software development project, that COCOMO II.2000 TEAM needs modification, and that future research is warranted.

Round 2 Responses

The Round 2 questions were to determine if either the team cohesion competencies of the ECIv2.0 (Rapisarda, 2002b) or the GEQ adopted for software engineers (Wellington, Briggs, & Girard, 2005) would be better methods to measure the COCOMO II.2000 TEAM variable and when and how often team cohesion should be measured. Each response to the questions in Round 2 is presented with a 100% stacked column representation of the panelists' responses, a table that organizes the panelists responses by expertise, and a analysis of the panelists' responses. Tables 15 – 18 present the panelists' responses and Figures 11-14 provide the 100% stacked column charts of the panelists' responses.

Round 2-Question 1 responses.

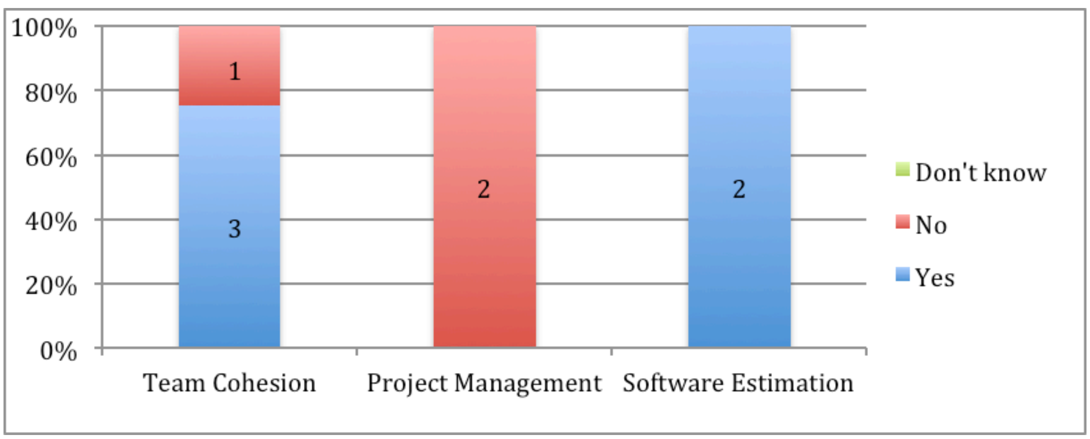

Figure 12. Round 2-Question 1 responses by expertise

Table 15. Round 2-Question 1 responses by expertise

Round 2 Question 1	**The ECI model was used in previous research, which determined nine emotional intelligence competencies strongly correlated with team cohesion (Rapisarda, 2002b). The survey questions associated with these nine competencies are attached to this e-mail. Based upon your expertise would these questions create a more effective method in determining the team cohesion value for the COCOMO II.2000 TEAM variable? Why or Why not?**
Team Cohesion (1)	"No. These items measure emotional intelligence competencies – they are correlated with the cohesion scale in Rapisarda's paper because teams with members that scored high on these items (i.e., frequently demonstrated these competencies) had teams that were more cohesive. In my opinion, team cohesion needs to be measured with items that ask about the team."
Team Cohesion (2)	"Yes, because they are based on inductive research to develop the items and deductive research on team performance, [by] Rapisarda and others since her thesis."
Team Cohesion (3)	"If there are theoretical reasons and the data is there, I don't see why not."
Team Cohesion (4)	"I think a lot of these questions would relate to how well the team functions. I am curious about how you would combine the data from a bunch of individuals. Does the team function as well as the best? As poorly as the worst? Or as the average? I'm not sure. I like that there are questions about mechanical things like how conflicts are handled, frustration management, and adaptability as those things can undermine or support a team regardless of their technical skills. This set of questions is pretty long, but I think they have included many characteristics of individuals that affect cohesion. I also like this one because it could be used to give each team member feedback and help detect what kind of skills people need to work on."
Project Management (1)	"No. The questions seem to be disregarding the lives of the people on the team. Maybe I don't 'hang out' with my team because I go home to my wife and kids as they are more important to me than any team or project or stuff. The questions don't consider any of these real-life situations."
Project Management (2)	"No. I think the ECI set of questions is too elaborate. I don't see estimators or project managers being willing to actually use this set of questions."
Software Estimation (1)	"One would expect the competencies to correlate to team cohesion and in a successful team environment all or most competencies should apply to all team members."
Software Estimation (2)	"For the development team, yes; for the "TEAM" of all stakeholders, no. The group of all stakeholders can and usually do have inherent conflicts and many of the survey questions are not applicable to a group."

Of interest within this first question in Round 2 is that there wasn't any indecision. Team Cohesion (2), Team Cohesion (3), Team Cohesion (4), and Software Estimation (1) all agree the ECIv2.0 questions correlated with team cohesion as stated in

Rapisarda's (2000b) research could be used in place of the COCOMO II.2000 subjective questions. Team Cohesion (4) presented concerns on how the measures of individual team members would be aggregated to a single team cohesion measure. The ECIv2.0 manual has a method of aggregating the individual scores into a team score. Of additional concern is the length of the survey. Future research would need to be accomplished in determining the best questions to determine the COCOMO II.2000 TEAM variable measurement methodology.

Team Cohesion (1), Project Management (1), Project Management (2), and Software Estimation (2) do not believe the ECIv2.0 would be a good substitute for the current COCOMO II.2000 TEAM questions. In fact, Team Cohesion (1) does not believe the ECIv2.0 questions would actually measure team cohesion and Project Management (2) believes the ECIv2.0 questions are too elaborate. Presumably, he believes they are too many and possibly they are not focused enough for team cohesion measurement. Project Management (2) and Software Estimation (2) do not believe the ECIv2.0 questions should be used at all because they are intrusive or because the questions are not designed for group measure and conflict within the stakeholder group is inevitable.

Both the arguments for and against have good points. Previous research has already associated some of the ECIv2.0 questions with team cohesion. Does that mean the questions can measure team cohesion? This researcher is not sure. Future research would have to be done to address that concern. Additionally, the ECIv2.0 questions associated with team cohesion are quite a few more than four and may need to be rewritten for a group setting. Again future research would need to be done to determine the best approach for developing and designing the questions.

Overall the researcher is very happy with the responses to this question. More than half of the panelists expressed their argument in detail and the goal of this question was to elicit opinion regarding the possibility of using the team cohesion correlated questions of the ECIv2.0 questionnaire (Rapisarda, 2000b).

Round 2-Question 2 responses.

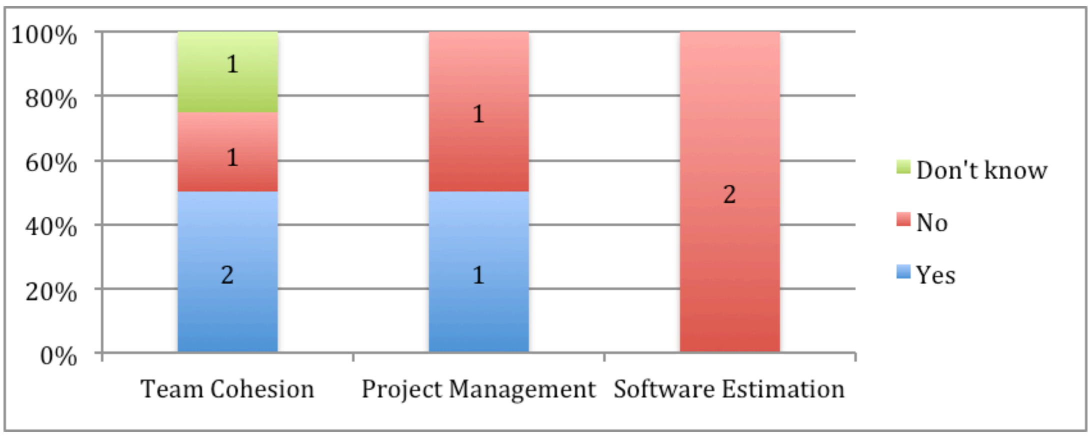

Figure 13. Round 2-Question 2 responses by expertise

Table 16. Round 2-Question 2 responses by expertise

Round 2 Question 2	**Based upon your expertise, would the questions of the GEQ survey modified for software engineers which directly correlate to team cohesion (Wellington, Briggs, & Girard, 2005), which is attached to this e-mail, be a more effective method in determining the team cohesion value for the COCOMO II.2000 TEAM variable? Why or Why not?**
Team Cohesion (1)	"Yes. The items of the GEQ are better measures of team cohesion than the COCOMO II.2000 items. But, I have to add that the GEQ is a long survey. It likely measures more than cohesion. Some of the items seem to focus on task issues – but these may be 'task cohesion' items, for example: Item 14 – 'Our team members have conflicting aspirations for the team's performance.' I have to add that item 16 is not a good item because the meaning is confusing: 'If members of our team have problems on the project, everyone wants to help them so we can get back together again.'"
Team Cohesion (2)	"I don't know the GEQ or the research using it so I cannot comment."

Round 2 Question 2	Based upon your expertise, would the questions of the GEQ survey modified for software engineers which directly correlate to team cohesion (Wellington, Briggs, & Girard, 2005), which is attached to this e-mail, be a more effective method in determining the team cohesion value for the COCOMO II.2000 TEAM variable? Why or Why not?
Team Cohesion (3)	"Sure. Same reason as question #1. The issue does it lead to better team performance."
Team Cohesion (4)	"This captures more touchy-feely aspects of cohesion and may not be a good reflection of cohesion in the work place. I am not sure that people in a workplace expect to hang with their coworkers, so all of the questions about outside of work may not be relevant to their attachment to the team. We confirmed it with students and that may have made those questions more relevant than they would be in real life."
Project Management (1)	"Same answer as #1. No. The questions disregard the lives of the people on the team."
Project Management (2)	"Yes, I think it speaks to a larger subset of what I think contributes to TEAM cohesion than COCOMO II's TEAM variable does, while still consisting of a manageable number of questions."
Software Estimation (1)	"No. Team cohesion applies to the team as a whole. At the root of the team is the development manager, the leader of the team. Team members can be motivated if the manager supports team collectively, which requires direct interaction with the team. If only one member of the team is interviewed/tested for the TEAM parameter, the leader is the key. The team leader is the important part of the team itself. Hersey and Blanchard (Management of Organizational Behavior, 1969) described this phenomenon many years ago, but the concepts still seem to escape software development organizations today. By the way, if the team members are isolated by walls, the Dilbert effect limits the cohesion. This is also true if the manager is physically separated from the team."
Software Estimation (2)	"No, for the 'TEAM' of all stakeholders. The..stakeholders can, and usually do, have inherent conflicts and many of the questions are not applicable to a group."

The majority of panelists did not agree the GEQ questions would make a good substitute for the COCOMO II.2000 TEAM questions nor did they believe the GEQ questions would present a good measure for team cohesion. Those panelists that did agree stated the GEQ questions were a better representation of team cohesion than the COCOMO II.2000 TEAM questions and mentioned the survey was more manageable. Additionally, they stated the questions may measure more than team cohesion but also task cohesion. However, as was discussed earlier task cohesion is a sub-category of team cohesion. Furthermore, Team Cohesion (3) stated the more important issue is "does it lead to better team performance. This research is focused on determining if the COCOMO II.2000 TEAM variable accurately represents team cohesion as we know it today and if there are any suggestions toward improving the questions used to assess the COCOMO II.2000 TEAM variable. However, it is the opinion of this researcher if a project manager is new to the teams he/she could use the GEQ or ECIv2.0 instruments to assess the team cohesion.

Those panelists' who did not agree the GEQ would be a good substitute for the COCOMO II.2000 TEAM questions provided the same reasons as for the previous question regarding the ECIv2.0 instrument. In particular Team Cohesion (4) stated "this captures more touch-feely aspects of cohesion and may not be a good reflection of cohesion in the work place" specifically because the questions ask about relationships outside the workplace. Project Management (1)'s response that the questions disregard the lives of team members is interesting in that this researcher is not sure how to interpret this argument. Presumably, the Project Management (1)'s believes the questions are too intrusive in that they ask team member's personal opinion about the work place. This researcher has worked in both introverted and extroverted environments, that is, an environment where team member's opinions were shunned and an environment where team member's were encouraged to bring conflict into the open and of the two, the most productive environment, was the extroverted environment. Again, this could be another area of future research.

In this researcher's opinion, future research should be conducted where each of the models are substituted for the four questions in the COCOMO II.2000 to see how the differences in estimation would be. Additionally, there may be a need to have different surveys for each of the three phases within the COCOMO II.2000 in order to more accurately portray the software development lifecycle.

Round 2-Question 3 responses.

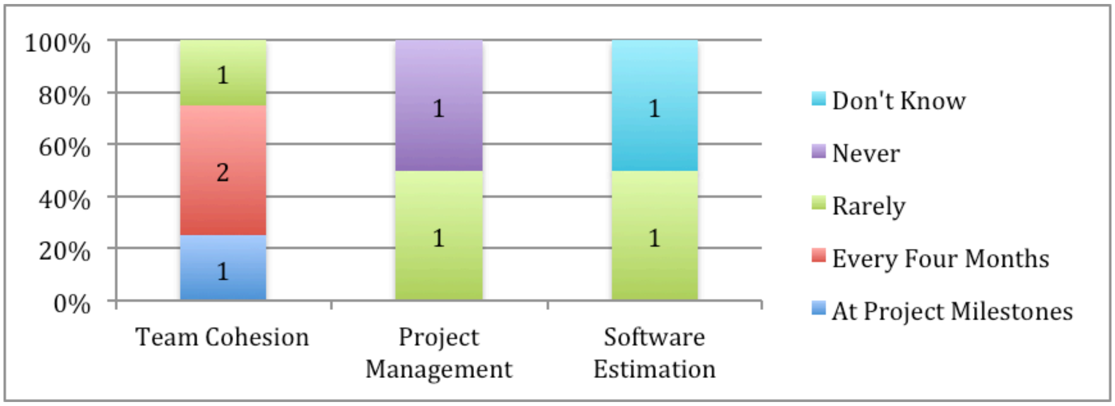

Figure 14. Round 2-Question 3 responses by expertise

Table 17. Round 2-Question 3 responses by expertise

Round 2 Question 3	**Based upon your expertise and the research literature presented in this study how often should a team cohesion survey be administered during the life cycle of a software development**
Team Cohesion (1)	"It depends on how long the team is together and how often they meet while they are together. Ideally, I'd like to administer it every 4 months. Keep in mind; however, I do not believe that cohesion is the best thing for you to measure. Cohesive teams are not always the best performers. I think the rule of thumb (based on research) is that if a team has high cohesion and high task focus (i.e., desire to perform well) than cohesion is good for performance. But, if the team has high cohesion and low task focus (i.e., no desire to perform well), than cohesion is not good for performance. I've been on a team where we like each other so much that we goof off and place the importance of our relationship above that of the task. Also, have you heard of Group Think? Group Think literature suggests that cohesion can result in a suppression of differences of opinion – which can hurt decision-making effectiveness."
Team Cohesion (2)	"At least at the beginning and at the end, but if the team were to exist for more than 6 months, I would administer a survey at major benchmarks or every 12 months, which came first."
Team Cohesion (3)	"Every 3-4 months. Cohesion is dynamic and evolves. The more the better. Better diagnosis of cohesion."

Round 2 Question 3	**Based upon your expertise and the research literature presented in this study how often should a team cohesion survey be administered during the life cycle of a software development**
Team Cohesion (4)	"Rarely-maybe once a year at most. Cohesion takes a long time to develop, so I would not expect it to change much in the short term. This gives a challenge to using it for planning: with the team newly formed at the time you are initially planning, can you predict what their cohesion will be later in the project? Also, the team members may get annoyed if they are asked too often. If the ECI model is deployed by having everyone rate everyone else, gathering the data will be a disruptive activity. Filling out that form for 10-20 people would take a long time and require a significant amount of attention. If it was part of each individual's annual performance review, team members could set goals based on it, work on specific skill weaknesses it detects, and, thereby, improve their ability to be a part of the team. These results would provide a moving measure of cohesion-as each team member's review comes up, that data refines the overall team's cohesion and feeds your planning activities."
Project Management (1)	"Never! Don't issue surveys to your team. If you do, you don't have a team, you have some sort of focus group or some such nonsense."
Project Management (2)	"It depends on the nature of the team, but overall I don't see this as a question of 'how often.' I would probably use the GEQ survey when the team is first put together. I might use it again if the team composition changes significantly, or if the team starts having significant problems. I would not be inclined to make the frequency of the survey tied to calendar time -- I'd be inclined to tie it to events. If the project is going well, I wouldn't re-administer the survey at all. If the project is going poorly, I'd have to be pretty sure the problems were related to team cohesion before I would re-administer the survey."
Software Estimation (1)	"The team cohesiveness is established at or near the beginning of the development. That is a fortunate circumstance for the estimator who must use the TEAM parameter as that basis of development estimates. The team cohesion will not likely change during the development unless the team manager, team structure, or environment are significantly changed. The cohesion is not likely to change if individuals are changed."
Software Estimation (2)	"Yes, for the 'TEAM' of all stakeholders, especially if one is attempting to calculate an estimate to complete the project. Depending on when the survey is administered, there will likely be changes in the group of all stakeholders. Unfortunately, there is no suitable survey for a group of people which is greater than the development team."

As can be seen by Figure 15 the responses to this question were quite varied. Only two agreed the team cohesion survey should be administered every four months and only one agreed the team cohesion survey should rarely be administered. One panelist was quite emphatic that team cohesion surveys should never be administered and one panelist completely dodged the question and his response is the only don't know response.

Team Cohesion (1)'s response is quite interesting and brings to light a very important issue of terminology and measurement standards. Team Cohesion (1) does not believe team cohesion should be the measurement of focus because a team with low task focus and high team cohesion will not be productive and the team cohesion measure will misdirect the estimation, thus, alluding to the importance of both measures of team cohesion and task focus. However, as team cohesion is defined to have two sub-categories of task cohesion and social cohesion, I disagree with Team Cohesion (1)'s assessment. Furthermore, this researcher agrees that both sub-category measures are important and should be used to determine which areas of improvement need to be addressed in creating a successfully collaborative team. With regard to terminology, it is obvious that terminology needs to be standardized when it comes to measuring team cohesion and has been cited as one of the key issues in past research regarding team cohesion.

Team Cohesion (4)'s response of incorporating team cohesion measurement questions in the annual review is very interesting indeed. She states the frequent measures of team cohesion would be disruptive and annoying to the team members due to the amount of questions per survey. By having the team cohesion measures in the annual evaluation it would provide guidance on areas of improvement and strengths. Additionally, the data would be available for a new project manager to review. This researcher likes this idea very much along with Project Management (2)'s suggestion of using the GEQ when significant changes happen to the team or when the team begins showing signs of trouble collaborating. This researcher believes these suggestions are very wise suggestions and if done correctly, could provide the most benefit. However, we need to keep in mind the scenario these teams they are discussing are established teams. When teams are newly formed and will be working together on a project for more than a year, it would be wise to measure the team cohesion at 3-4 month intervals during the first year and then once a year occurring with the annual evaluation. Indeed, some team members participate on more than one team at a time. In this researcher's opinion, the most judicious project manager would assess the team and measure accordingly; that is, if the team is new, measure frequently; if the team is established measure annually or when problems arise. Additionally, Team Cohesion (4)'s statement: "with the team newly formed at the time you are initially planning, can you predict what their cohesion will be later in the project?" This is a very good question. The COCOMO II.2000 TEAM methodology does allow for a very low team cohesion measure and this researcher would use that value to estimate the development because it is better to over estimate time than under estimate. Additionally, the team cohesion measure could be reassessed at intervals and the estimate further refined.

Software Estimation (1)'s response directly goes against what this research has presented and argued; specifically, Software Estimation (1) states that team cohesion will not change unless the work environment changes, however, this research has presented that team cohesion is known to change over time. Presumably, Software Estimation (1) may be discussing a well-established team and, therefore, their team's cohesion may not change dramatically during the project. However, most software development teams today are brought together for the duration of the software product and then move on to other teams. Therefore, this researcher disagrees with Software Estimation (1)'s argument of when to measure software teams.

Software Estimation (2) did not answer the question presented but did make an interesting statement "there is no suitable survey for a group of people which is greater than the development team." If this statement is accurate than it has affirmed the TEAM variable does not accurately assess the team cohesion of the software development team as defined within the COCOMO II.2000 manual. This statement also confirms the need for this research and future research to find a suitable survey that will measure the software development team's cohesion. Again, the software development team is inclusive of all stakeholders within the realm of the production of the software product; e.g., customers, developers, team leads, testers, etc.

Overall this researcher is very satisfied with the panelist's responses as they have been the most articulated responses since the beginning. Furthermore, the panelist's responses are fertile with avenues of future research regarding team cohesion and the dynamics of the software development team behaviors and processes and the effects each has on the other.

Round 2-Question 4 responses.

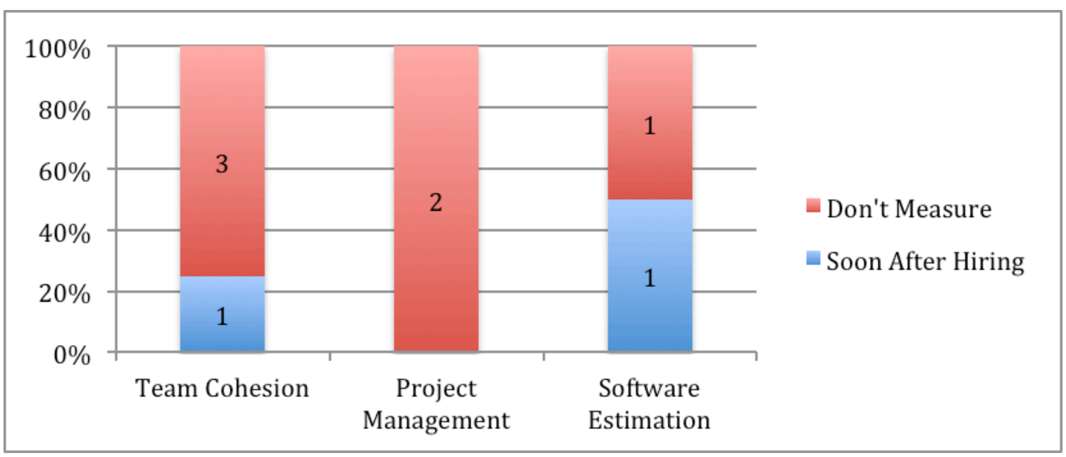

Figure 15. Round 2-question 4 responses by expertise

Table 18. Round 2-Question 4 responses by expertise

Round 2 Question 4	**When a team member exits and a replacement is brought on board, based upon your expertise, when should team cohesion be measured? When the team member exits? Soon after the team member exits? After the replacement is hired but before they report for work? Soon after the replacement is hired? Why or Why not?**
Team Cohesion (1)	"I would not worry about replacement members. I think that cohesion should be measured every 4 months (3 times per year). Team norms (i.e., socially acceptable patterns of behavior within the team) will influence the level of cohesion in the team -- and Sherif's research on norms (done in the 1950s) shows that norms are difficult to change and that they remain the same even when a large number of team members come and go. The team leader is a different story. The team leader has a HUGE influence on team norms and team cohesion."
Team Cohesion (2)	"The team cohesiveness should be based on the team and not on composition of specific people. So the addition or deletion of members is not relevant in timing of surveys."
Team Cohesion (3)	"Soon after member starts performing and thereafter."
Team Cohesion (4)	"I wouldn't react to someone leaving or joining the team. I don't expect cohesion to change much when one person leaves. Cohesion reflects many individual relationships within the team and the loss of one member probably won't change the other relationships quickly. Similarly, adding a new member to the team will take a while to affect the overall cohesion. Every team I have been on has had people coming or going regularly. If you measured as a result of those events, you would always be measuring. Also, reacting to those kinds of events shows the team that you are worried that the events are undermining their abilities."
Project Management (1)	"I don't 'measure' team cohesion. I observe it every day."
Project Management (2)	"I would not re-measure just for replacing one team member unless problems arose after the new team member was added. Even then, I would probably rely more on just talking with team members than re-administering the survey. As with Q3, I don't think amount of time that's passed is the criteria to use for administering the survey. I would choose to (re)administer the survey based more on events or project condition than on time passing."
Software Estimation (1)	"As I pointed out in the answer to Question 3, the key to the team is the leader. The individual developers are frequently joining/leaving the team as the development process moves forward. One of the leader's responsibilities is to keep the team motivated and working together. It is possible to add a disruptive influence to the team which could break up the team; thus, it is important to screen the potential new members to assure the disruption does not occur and affect the team. If the disruption starts to affect team performance, if the manager does not see the problem, the team will make the disruption known."
Software Estimation (2)	"If the survey is only for the development team, then after the replacement is hired, but not too soon. For the group of all stakeholders, the stakeholders may change but the concept of 'hiring a replacement' does not make any sense."

The majority of the panelists do not believe team cohesion should be reassessed when a team member departs or new team member begins. They argue team cohesion should not change with the exit on entry of team members; furthermore, team cohesion should only be reassessed out of the norm when behavioral problems within the team occur.

Team Cohesion (1) and Software Estimation (1) both agree the leader has significant influence over the team's cohesion, however, Software Estimation (1) suggests screening potential new team members to ensure they will be a good fit for the team; e.g., will not cause any disruptions or delay productivity in any way. Furthermore, Software Estimation (1) states the team will notify the manager when disruption begins. This researcher thinks the work environment would determine if the team members would notify the manager; if the work environment is conducive to open communication that would most likely be a true statement, however, if the environment is not open to communication, at best the manager might be indirectly notified via the team members waiting on the problem team member's work so they can continue their own.

Additionally, Team Cohesion (1)'s statement of the team leader's influence on the team regarding team norms and team cohesion is also very interesting. It raises questions like how much does team cohesion change when a new team leader enters the picture? How much influence does the team leader have over team cohesion, task cohesion, and social cohesion? However, the focus on this research is to determine if TEAM is accurately representing team cohesion as we know it today, and this researcher would speculate that team cohesion would change when a new team leader is brought in.

Of interest is Project Management (1)'s statement that he does not measure team cohesion but observes it every day. This researcher would conclude the observation would lead to a determination, which would lead to some action or inaction. Therefore, the observation is a measurement, although a subjective one and subjective measurements of team cohesion have been determined to be an inaccurate means of measuring team cohesion.

Team Cohesion (4)'s argument against measuring when team members leave is an interesting point regarding how the implementation of measuring team cohesion affect the behavior of the team? Again, great question for future research.

Software Estimation (2)'s response continues to isolate the development team from the stakeholders, whereas, this research has made it clear the development team is inclusive of all stakeholders, which is in alignment with the definition of the COCOMO II.2000 model.

Round 3 Responses

This only and final question for Round 3 and this Delphi study is an open-comment question and a visual display is not practical. The responses for this final question are words of wisdom for future research of team cohesion and COCOMO II.2000 TEAM. The analysis for these responses is presented in Table 19

Round 3-Question 1 responses.

Table 19. Round 3-Question 1 responses by expertise

Round 3 Question 1	**In your expert guidance, what research should be accomplished next to improve software estimation? How can the use of emotional intelligence assessments assist in improving software estimation?**
Team Cohesion (1)	"My own research shows that team leaders who score higher on measures of emotional intelligence (EI) produce teams that are: (a) emotionally competent teams, and (b) higher performing than teams with leaders who are lower in EI…When team members don't work well together, they don't share and integrate what they know and think and team performance is easily reduced."
Team Cohesion (2)	"The [direct values] (DVs) should be team performance..[and]Team engagement as a parametric estimate of commitment…The [indirect values] (IVs) should include individual characteristics that make people team resources, team composition, and such. The team processes and quality of the team relationships should be the mediating variables between the IVs and DVs. I would use as control variables age, gender and how long the team has been in existence."
Team Cohesion (3)	"More understanding on what matters in software estimation. What team-level factors (if any) matter. A more precise delineation of what influences software estimation. I am not convinced that EI helps, data will determine that."
Team Cohesion (4)	"I have to admit that this question has required the most thought. I was struck by the range of answers to the questions in part 2. For some of the questions, we disagreed pretty significantly which could give pointers to interesting questions for future research. Absent that, I would like to think that there is some magic method that will improve our ability to estimate, but I am afraid that our poor history of estimation is more a poor history of being able to know what things we will have to work on. And I don't believe we are suddenly going to get better at predicting the long term set of things we should build. While that sounds very unhelpful, I think the answers to your questions have shown a strong belief in the idea that team cohesion affects productivity. Therefore, I think the next interesting question is sort of the converse of the question you are asking. Instead of 'Can measuring team cohesion help (the estimation part) of our process?' I think the next question to ask is 'How does our process affect team cohesion?'"
Project Management (1)	"Talk to people who estimate software well. These people will be hard to find as they don't garner much attention. As for Emotional Intelligence, I would ask project managers to assess the emotional intelligence of their teams. Then, work with the team to improve their EI."
Project Management (2)	"[The] research focus on emotional intelligence assessments is interesting and potentially useful. I think the main applicability is probably not in estimation, however. This is not unique to your research; I believe the same is true of [COCOMO] itself. There are too many factors that require too many subjective assessments for the model to be reliable for forward-looking estimates. Having said that, I think the [COCOMO] model is a useful model of the factors that affect software projects. Your work on emotional intelligence adds depth to the team cohesion area of that model. I just wouldn't use it for estimation purposes. I would use it to motivate project managers to strive for high team cohesion, and your work elaborate[s] many of the factors that managers should consider."
Software Estimation (1)	"I am not sure how the use of emotional intelligence assessments can help in improving software estimation unless the assessments are applied to the project manager. In that case the attributes of the team cannot be directly assessed, but if the manager does not do well in the assessment, the team would not do well either. I would look into the behaviors described by Blanchard and Hersey. I hope that is at least a little help."

Round 3 Question 1	**In your expert guidance, what research should be accomplished next to improve software estimation? How can the use of emotional intelligence assessments assist in improving software estimation?**
Software Estimation (2)	"I can think of two areas related to TEAM...I like the [task cohesion] aspect more than the [team cohesion] concept. [Suggest you] determine, via Delphi and [future] data analysis..the relative value of development team task [cohesion] compared to the all stakeholders (group) task [cohesion]..[and] the impact of domain on the TEAM scale factor range."

Team Cohesion (1) suggests measuring the EI of the project manager would be the best way to use emotional intelligence in improving software estimation because her own research has presented leaders with high EI measures produce teams that are emotionally competent and higher performing. What would be very interesting is research on how team cohesion and the EI measures of project managers would correlate.

Team Cohesion (2) provides a structure of future team cohesion research stating the direct values should be team performance and engagement as a parametric estimation of commitment; the indirect values would include desired characteristics of team composition and members; the team processes and qualities of team relationships would be the mediating variables; and, finally, the control variables would be age, gender, and time the team member has been with the team. Team Cohesion (2) has definitely provided a blue print for future research in team cohesion.

Team Cohesion (3) suggests "a more precise delineation of what influences software estimation." Additionally, Team Cohesion (3) stated he is not convinced that emotional intelligence provides any benefit towards measuring software estimation, however, future research could determine if emotional intelligence measures of the development team used to determine team cohesion would provide any benefit when compared to software estimation done without emotional intelligence measures.

Team Cohesion (4) is not very optimistic about the future of software estimation improvement via team cohesion measures and suggests determining how the software estimation processes affect team cohesion. This research has presented the importance of team cohesion to productivity and the COCOMO II.2000 model recognized it as a variable of concern. This researcher believes estimation practices will improve and it is the purpose of this research to inform project managers to reconsider the estimation tools they are using and to look at all the dynamic variables that come into play within the software development process. We all know an estimate is a rough guess based upon the information to date, however, COCOMO II.2000's TEAM variable needs modification in order to truly measure team cohesion, as we know it today. Therefore, it is possible that other variables need to be reviewed or new ones added to improve the estimation process. Additionally, project managers and executives should not bank on the estimate as fact and should not be surprised if the estimate falls short.

Project Manager (1) suggests gathering wisdom and experience from successful project estimators and agrees the project manager should measure the emotional intelligence of the team to find areas where the team needs improvement. A Delphi study of 'good' software estimators would be a wonderful research project. Another future research project would be to compare the data gathered from the Delphi study to the current software estimation models.

Project Management (2) is in complete agreement with Project Management (One)'s response. Project Management (2) stated although he doesn't believe emotional intelligence would be useful in measuring team cohesion, it would be useful to the project manager to assess the state of his/her team in determining possible areas of improvement.

Project Management (2) also stated this research and its suggestion of emotional intelligence to improve team cohesion has provided more depth to the team cohesion issue.

Software Estimation (1) is in agreement with the Project Management assessments. He believes the only way emotional intelligence would improve software estimation is if it is used to measure the project manager's emotional intelligence. Additionally, he further suggests looking at the behaviors listed within the research of Hersey and Blanchard (1969) who are the authors of the situational leadership theory. Presumably, these behaviors could be compared to the four questions used to assess team cohesion in the COCOMO II.2000 TEAM scale factor.

Software Estimation (2) is very interested in the task cohesion sub-category of team cohesion and suggests future research be accomplished that compares the task cohesion of all stakeholders compared to the task cohesion of the development team alone. As stated earlier in this research, I believe each phase of the software life cycle may need a different survey that is customized toward the behaviors for each phase. Overall, the panelists provided wonderful words of wisdom and a fertile garden of ideas for future research.

CHAPTER 5. DISCUSSION, IMPLICATIONS, RECOMMENDATIONS

The goal of this study is to provide evidence of how one software estimation tool may be providing the project manager erroneous estimation results. Therefore, faulty estimation tools may be an undiscovered factor in the high failure rates of delivering software products on time, on budget, and on target (Standish Group, 1999, 2001, 2002). The objective of this study is to investigate the effectiveness of the team cohesion measure of a well-known software estimation tool, COCOMO II.2000, used by project managers to determine cost and time to completion for a software product. Specifically, this research investigates the method used to determine the COCOMO II.2000 TEAM special factor that represents the team cohesion of all the project stakeholders and is one of five exponential special factors within the estimation model. The TEAM variable is to represent the team cohesion of all stakeholders within the development of the software product, which includes the customer groups and the programming team groups. The customer group is inclusive of Subject Matter Experts (SMEs) and any representatives the customer provides to assist the programming team toward a full understanding of the requirements and business logic of the product to be completed. The programming group is inclusive of the Project Manager(s), Task Lead(s), and individual programmers. Henceforth, the *development team* must be understood as inclusive of both customer and programming groups. As the research results have been detailed in Chapter Four the interpretation of the research results are presented in this chapter.

Findings

The Round 1 survey contained seven questions designed to ascertain the effectiveness of the method used to determine the COCOMO II.2000 TEAM value, which represents the team cohesion of the development team. The questions and responses are presented in Chapter Four. As to the interpretation and analysis of the panelists' overall responses to the Round 1 survey questions, the panelists overwhelmingly agreed the COCOMO II.2000 TEAM value does not effectively represent team cohesion, as it is known today. Specifically, six of the eight panelists agreed the COCOMO II.2000 TEAM method does not effectively measure team cohesion (Figure 9). Additionally, five of the eight panelists agreed the subjective opinion of the project manager is not an effective means to measure a team's cohesion (Figure 8). Three of the eight panelists agreed the COCOMO II.2000 TEAM method does not consider the change of team cohesion over time while the other five did not know (Figure 10).

When asked for the best way to introduce the dynamic phenomena of team cohesion changing over the life cycle of the project, five of the eight panelists agreed the team should be measured during the life cycle of the project (Figure 11). One panelist argued that measuring the change of team cohesion over the life cycle of the project is impossible and used one of this researcher's citations (Demarco & Lister, 1999) to support his argument. Although Demarco and Lister stated, "You can't make teams jell" (1999, p. 132) they didn't say it is impossible to tell exactly what makes a team jell. Demarco and Lister describe characteristics of a jelled team as having low turnover, a strong sense of identity, a sense of eliteness, joint ownership of the product, and the obvious enjoyment people take in their work (p. 127). They further describe a jelled team as a group of people who have a common definition of success, a team spirit, and momentum; "a group of people so strongly knit that the whole is greater than the sum of the parts" (p.123). Therefore, a jelled team can be ascertained by the characteristics of the team's interaction with each other in accomplishing the tasks set to meet the ultimate goal. This research has provided evidence via past research (Wellington, Briggs, & Gerard, 2005; Rapisarda, 2002b) that team cohesion does change over time and that team cohesion can be measured by asking each team member questions which describe Demarco's and Lister's jelled team characteristics.

Additionally, two panelists sited specific research as suggestions on how and when to measure team cohesion; specifically *A Temporally Based Framework and Taxonomy of Team Processes* by Marks, Mathieu, and Zacarro (2001) and *A Theory of Team Coaching* by Hackman and Wagemen (2005).

In Marks, Mathieu, and Zacarro (2001) *A Temporally Based Framework and Taxonomy of Team Processes*, the authors recognize the temporal influences on teams and believe the dynamic states of a team such as team cohesion have been buried within the static teamwork or process label and need to be isolated and measured as emergent states which describe "cognitive, motivational, and affective states of teams, as opposed to the nature of their member interaction" (p. 357). Additionally, Marks, et al. created a conceptual framework and taxonomy which considers the importance of "how time impacts teamwork and that various teamwork processes are more and less likely to occur at different points in the performance cycle" (p.372). This researcher believes the research of Marks, et al. would synchronize with the research regarding the isolation of task cohesion from team cohesion (Carless & DePaola, 2000, Knouse, 2007). As the task cohesion research so convincingly stated it is not so much how the team members interact socially as it is how the team members interact professionally; that is, can each team member respect the professional expertise of the others to assist the team in meeting the ultimate goal of completing the project (Knouse, 2007, p51)?

Furthermore, in Hackman and Wagemen (2005) *A Theory of Team Coaching*, the authors discuss how the dynamics of a team change over time and how to best provide a team coaching strategy to align with the beginning, midpoint, and end of a team's life cycle. Although, this research presents additional theories of what makes up a team, this researcher does not believe Hackman's and Wagemen's research (2005) adds any more depth to this study but does support this research. Additionally, this researcher believes the need for refinement and standardization for group dynamic measures is desperately needed, as phenomenology and definitions seem to overlap in previous research. This researcher believes we have proof of concept but need to bring all researchers' together to determine the best pathway forward for measuring team cohesion.

The Round 2 survey questions were designed to determine when team cohesion should be measured and to investigate the use of two other team cohesion models to replace the current COCOMO II.2000 TEAM method. The two models presented to the panelists for consideration are the correlated EI competencies of the ECIv2.0 associated with team cohesion (Rapisarda, 2002b) and the GEQ modified for software engineers (Wellington, Briggs, & Girard, 2005).

Five of the eight panelists (Figure 14) agreed the EI competencies associated with team cohesion (Rapisarda, 2002b) questions would provide a better method of measuring team cohesion within the COCOMO II.2000 model, however, only three of the eight panelists (Figure 15) agreed the GEQ modified for software engineers (Wellington, Briggs, & Girard, 2005) would provide a better method of measuring team cohesion within the COCOMO II.2000 model. Noted concerns were the length of each survey, the need for all team members to participate, or if only one is to be measured it should be the project lead or project manager, and, finally, concerns of how the scoring of the surveys would be culminated into a group score.

The ECIv2.0 is a 360-degree survey, which means that each team member must take the survey to provide an individual measure. Then each member must take the survey to answer the questions regarding at least four other team members. These scores are then aggregated into a group score and theoretically; the group score of the correlated competencies (Rapisarda, 2002b) would provide the team cohesion measure.

The GEQ is a team survey where only each member of the team takes the survey and the average of all team members' scores represents the team cohesion measure for the group. Most panelists preferred the GEQ questionnaire to the ECIv2.0 because the questions asked specifically how each member envisioned and participated within the team. However, each survey would take more time than the current method in that each team member would need to be involved rather than just the project manager.

Additionally, the fact that team cohesion changes over time implies the need for frequent measure. The question of when should a team should be measured was presented with five possible answers; don't know, never, rarely, every four months, and at project milestones. Six of the panelists agreed team cohesion should be measured during the life cycle of the project. One of the eight panelists agreed the team should be measured at the project milestones while two of the eight panelists agreed team members should be measured every four months. Three of the eight panelists stated team members should rarely be measured and one of the eight panelists stated team members should never be measured (Figure 16). Measuring the team cohesion of the development team at project milestones is in alignment with Tuckman's small group development model stages of forming, storming, norming, performing, and adjourning (Tuckman & Jensen, 1977).

The Round 3 survey question asked the panelists what research should be accomplished to improve software estimation and how can emotional intelligence assessments assist in improving software estimation? All panelists provided very interesting insights and avenues of future research toward improving software estimation, team cohesion, task cohesion, and how all three interact within the software development process.

Research has been presented in this study that higher team EI measures are associated with better performing teams and are directly associated with team cohesion (Druskat, Sala, & Mount, 2006). Four of the eight panelists suggested administering the EI survey to the project manager alone rather than the team, arguing the project manager sets the stage for a successful or unsuccessful team. However, a project manager with a high EI score would successfully ascertain any team cohesion issues and, therefore, successfully guide the team toward resolving any issue before it became problematic. This researcher argues that in today's project management, many project managers do not interact daily with the team that is to be estimated. Additionally, project managers new to their teams and who must provide estimates of development before getting to know the team would find recent team cohesion measures very useful. The benefit of having a team cohesion measure assists a project manager toward ascertaining if adjustments need to be made. Additionally, a project manager, who does not interact daily with a team, will have a better understanding of how to guide task leads toward improving team cohesion.

Another panelist suggests using the converse approach of researching how the software development process effects team cohesion in order to discover a better team cohesion model rather than researching how team cohesion affects the process. I find this suggestion fascinating and very useful. Many times it is thinking outside the box that brings light to a better definition of what is in the box. Additionally, Potok and Vouk (1999) presented research regarding the influence team behavior has on the project parameters and found that correlated teams work best with short term development schedules such as the Agile development methodology. This research has described software development as very challenging and Sawyer stated software development is an "impressively complex socio-technical activity" where limited and/or strained interaction between team members will reduce team cohesion (Sawyer, 2004, p. 95). However, in an article titled "Process or Behavior: Which is the Risk and Which is to be Managed," another researcher found "it is behavior and not the set of procedures, which is the risky factor" (Navare, 2003, p. 6). Additionally, in an article titled "The Art, Science, and Engineering of Software Development," McConnell (1998) stated what makes software engineering so different from other engineering is that "software is so labor intensive that a significant amount of engineering energy must be focused on *project* goals in addition to *product* goals—on the means to the ends as well as the ends themselves" (p. 119).

Measuring different types of engineering teams and different types of software engineering methodologies; e.g. Agile, Lean, CMMI, etc. would provide a broad overview of how process changes team cohesion. However, the challenge would be to isolate the process from the team behavior.

Along this same viewpoint, Team Cohesion (Three) panelist stated "more understanding on what matters in software estimation…[specifically] a more precise delineation of what influences software estimation." As has been mentioned in this research regarding the software life cycle there are many variables within the process; some of which are uncontrollable but need to be added within a risk variable. For example, if the development team works in an area that is prone to natural disasters; e.g., hurricanes, earthquakes, etc., this needs to be considered as a risk. Another problem within the software development life cycle is when corporate executives or project managers enforce an impossible delivery deadline (Jones, 2006). These areas of concern are very valid and very real within software development projects; however, I have not seen them mentioned in any risk variables within the COCOMO II.2000 estimation model.

Two other panelists provided specific suggestions. One panelist provided suggestions for using emotional intelligence assessments to assist in improving software estimation and another panelist provided specific suggestions for research to improve software estimation within the COCOMO II.2000 TEAM method.

Team Cohesion (Two) panelist suggested using team performance and team engagement as a parametric estimate of commitment as direct values and team composition and characteristics that define the team members as team resources as the indirect values. He further suggests using team processes and team relationship quality as mediating variables between the direct and indirect values and suggests using age, gender, and time the team has been in existence as control variables. As this panelist has participated in much EI research and is considered an EI expert I believe this specific suggestion is a definite area for future research.

Software Estimation (Two) panelist suggested comparing the task cohesion of the development team and then the entire group of stakeholders. Apparently, this panelist does not understand the COCOMO II.2000 TEAM variable is to represent the measure of all stakeholders and may have misunderstand this researcher's use of development team within this research. Within this research the development team is defined as all stakeholders involved in the development of the software product. However, comparing the task cohesion and team cohesion of all the team members would be interesting to see how much of the social aspect weighs in toward estimation versus the task cohesion perspective of professionally collaborating with a team to reach a certain goal.

Research Implications

First and foremost, the panelists have agreed the COCOMO II.2000 TEAM method does not provide an adequate measure of team cohesion, and, therefore, it can be concluded that the software estimation measure of the COCOMO II.2000 Post Architecture model is faulty. However, the reader must consider the COCOMO II.2000 model was designed almost twenty years prior to the most recent team cohesion research and, therefore, in order to accurately measure team cohesion today the COCOMO II.2000 TEAM method should be adjusted accordingly. This research also argues that although the project manager is responsible for the project, the project manager's tools, unbeknownst to him/her, may not be providing effective estimation measures, thus, attributing to the high software project failure rate.

The COCOMO II.2000 model must take into consideration the dynamic of team cohesion changing over time. Further research must be completed to determine the best method of introducing another variable to represent the change of team cohesion over time. The panelists' preferred the GEQ modified for software engineers team cohesion survey as a method to be used to determine COCOMO II.2000 TEAM measure. Further research must be accomplished to investigate how the GEQ will work within the current COCOMO II.2000 model.

Second, the panelists have agreed that the COCOMO II.2000 TEAM method must be updated to meet the current research definitions of team cohesion, specifically, the subjective measure of the project manager is not enough and the whole team should be involved in determining the team's cohesion measure. Additionally, the COCOMO II.2000 TEAM method must take into consideration the change of the team cohesion measure over the life cycle of the project in order to effectively measure team cohesion. Further research is needed to redefine the COCOMO II.2000 TEAM method.

Finally, five of the eight panelists agreed task cohesion, a facet of team cohesion, is a very important factor toward successful software development. Task cohesion, found to be the "better predictor of work-group performance" (Carless & DePaola, 2000, p. 85), is defined as the "task oriented belief about achieving the team's goals through commitment to the team approach (Rainey & Schweickert, 1988)" (Salas, Burke, Fowlkes, & Priest, 2004, p. 438). The panelists agreed task cohesion is a very important measure, which needs further research in isolating it from team cohesion to use the task cohesion measure as the development teams' cohesion measure rather than team cohesion as a whole.

Overall, the implication for this research is the COCOMO II.2000 software estimation model has a weak team cohesion variable, which causes an exponential error in the software estimation findings. There is an adjustment variable within COCOMO II.2000 that allows each customer to customize the COCOMO II.2000 software estimation tool, which could mitigate the error of the TEAM variable, however, a good model should represent its variables as accurately as possible. In this case, it is clear this panel of project manager, team cohesion, and software estimation experts have provided enough doubt as to the accuracy of the TEAM measure that future research is needed to improve the COCOMO II.2000 TEAM model. Additionally, this research has provided evidence for project managers to reconsider their software estimation tools and gather metrics for each project in order to compare estimation results with historical results.

Research Limitations

The number of panelists for the study would be the primary limitation of this research. I believe more panelists within the same expertise criteria of project management, team cohesion, and software estimation expertise would provide a more definitive range of interpretations. Additionally, some of the panelists expressed concerns of inability to adequately respond to certain questions due to inexperience in a specific area; e.g. not familiar with the COCOMO II.2000 software estimation model or ECIv2.0 or GEQ models. I believe another Delphi study with more panelists who are within the areas of expertise mentioned in this Delphi study but are more familiar with software estimation process would provide a more thorough and definitive study.

Finally, one of the primary goals of this Delphi study is to gain at most convergence and at least consensus among the panelists. Although this study did not reach convergence, a limitation to this study, it did reach consensus on many of the questions asked of the panel.

Another limitation to this study is that only one software estimation model was investigated, COCOMO II.2000. Other software estimation models such as SEER-SEM, COSYSMO 2005, and REVIC should be investigated for team cohesion variables, the effectiveness of the team cohesion variable, if it exists, and if not, introducing a team cohesion variable, and, finally, using task cohesion methods to measure team cohesion within these models.

Finally, the researcher's worldview is that of a project manager, a senior software engineer, and a scientist. The researcher has spent the last twenty-five years developing software for the DoD and holds a Bachelor of Science in Physics and a Master of Science in Management Information Systems. However, the researcher has been studying EI and its importance in creating ideal work teams for the past fifteen years and has encountered first hand projects that have failed due to poor project management. Therefore, the researcher is very motivated toward discovering how the software estimation practice can be improved and in investigating how the effects of emotional intelligence, team cohesion, and task cohesion can assist in improving the measurement of software estimation, which may bring an unintended bias to the research.

Future Research

Modeling human behavior is a daunting task and some of its research history has been presented. All industrial leaders want to know what the magic formula is for productive teams. As many management gurus have stated respect is first and foremost and then follows trust. Additionally, when it comes to working with geeks, a.k.a. software engineers, a good project manager knows that many geeks fresh from college have under developed social skills and the project manager will have the ability to mold young software engineers toward becoming successful collaborative professionals. Furthermore, colleges have the tendency to breed competitiveness and isolationism when creating software engineers, therefore, it has been suggested that colleges' should require curriculum that develops a professional collaborative social skillset within a team environment.

Although this study has determined the inaccuracy of the team cohesion measure within the COCOMO II.2000 estimation model, further research must be accomplished toward discovering the importance of team cohesion to the software development process. Furthermore, isolating the team cohesion measure to task cohesion will be very beneficial to the software estimation industry. Clearly, future research toward how best to use EI toward improving team cohesion and how this knowledge can benefit the project manager in better estimating software development projects is needed.

This researcher would like to see the initial quantitative research with 250 software development teams completed. I believe the benefits of this research, if done correctly, will provide definitive measures toward improving the team cohesion measure. Further research to define the task cohesion methodology of measuring team cohesion would provide further advances of software engineering team cohesion measurement. Definitive analysis of the substitution of the ECIv2.0 and GEQ questionnaires used within the COCOMO II.2000 estimation model would also provide further defining results toward measuring team cohesion. As mentioned previously, investigation of team cohesion measures in other software estimation models such as SEER-SEM, COSYSMO 2005, and REVIC would shed further light on how team cohesion measures affect the accuracy of software estimation.

Conclusion

This research is just the beginning of much more research that needs to be accomplished to improve team cohesion estimation modeling with regard to software engineering. Although many traditional project managers shun the importance of soft skills within the software development process, any time a group of individuals come together to complete a task, team cohesion and its importance to success must be considered if a project manager wants to successfully deliver a product on time, on target, and on budget.

EI measurement can be of great benefit to an organization as a whole when hiring software engineers and of great benefit to the project manager when a tiger team needs to be created quickly. Furthermore, a project manager who is new to a team could use EI and team cohesion measurement to ascertain the collaborative ability of his/her team quickly. Team cohesion measurement can minimize risk to project failure. Research needs to be accomplished toward creating a minimal survey that can accurately measure team cohesion and EI.

Although emotional intelligence (EI) was not the direct focus of this research it should be considered as a tool in creating effective software development teams and has been proven to provide cost saving benefits to organizations in areas of team development, conflict resolution, and lowering turn over rates thus leading to less training costs and less HR administrative activities (Blattner & Bacigalupo, 2007; Druskat & Wolff, 1999; Jordan & Troth, 2004; Rapisarda, 2002a).

Finally, this researcher wishes to thank the panelists for their participation in this research. Without their participation this research could not have been accomplished and I wish each of them to know how much I deeply appreciate the time they each took from their busy schedules to indulge a doctoral candidate's investigation toward knowledge advancement in the area of team cohesion and software estimation models.

REFERENCES

Agarwal, R., Kumar, M., Yogesh, Mallick S., Bharadraj, R., Anantwar, D. (2001). Estimating software projects. *ACM SigSoft Software Engineering Notes, 26* (4), 60-67.

Ashkanasy, N. M., & Daus, C. S. (2005). Rumors of the death of emotional intelligence. *Journal of Organizational Behavior , 26*, 441-452.

Bar-On, R. (1988). *The development of an operational concept of psychological well-being.* Unpublished doctoral dissertation, Rhodes University, South Africa.

Bar-On, R. (1997). *BarOn Emotional Quotient Inventory (EQ-i): Technical Manual.* Toronto, Canada: Multi-Health Systems.

Bar-On, R. (2000). Emotional and social intelligence: Insights from the emotional quotient inventory. In R. Bar-On, & J. D. Parker (Eds.), *The handbook of emotional intelligence* (pp. 363-388). San Francisco: Josey-Bass.

Bar-On, R., & Parker, J. D. (2000). *The handbook of emotional intelligence.* (R. Bar-On, & J. D. Parker, Eds.) San Francisco, CA: Jossey-Bass.

Bar-On, R., Handley, R., & Fund, S. (2006). The impact of emotional intelligence on performance. In V. U. Druskat, F. Sala, & G. Mount (Eds.), *Linking emotional intelligence and performance at work: Current research evidence*. Mahwah, NJ: Eribaum.

Barsade, S. G. (2002). The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly , 47*, 644-675.

Barsade, S., Brief, A. P., & Spataro, S. E. (2003). The affective revolution in organizational behavior: The emergence of a paradigm. In J. Greenberg (Ed.), *Organizational behavior: The state of the science* (pp. 3-52). Mahwah, NJ: Eribaum.

Barth, S. (2001). 3-D Chess: Boosting team productivity through emotional intelligence. *Harvard Management Update , 6* (12), 3-5.

Benediktsson, O., & Dalcher, D. (2004). New insights into effort estimation for incremental software development projects. *Project Management Journal, 35* (2), 5-12.

Bernheisel, W. A. (1997). *Calibration and validation of the COCOMO II.1997.0 Cost/schedule estimating model to the space and missile systems center database.* Air Force Institute of Technology Air University Air Education and Training Command, Graduate School of Logistics and Acquisition Management. DTIC.

Blattner, J., & Bacigalupo, A. (2007). Using emotional intelligence to develop executive leadership and team and organizational development. *Consulting Psychology Journal: Practice and Research, 59* (3), 209-219.

Boehm, B. W. (1991). Software risk management: principles and practices. *IEEE Software , 8* (1), 32-41.

Boehm, B. W. (2000). *Software cost estimation with COCOMO II: an overview of the book*. Retrieved from http://ebookbrowse.com/overview-pdf-d24832203.

Boehm, B. W. (2006). A view of 20^{th} and 21^{st} century software engineering. *28th ACM/IEEE International Conference on Software Engineering (ICSE 2006), pp.12–29; Shanghai, China, May 2006*.

Boehm, B., & Fairley, R. (2000). Software estimation perspectives. *IEEE Software , 17* (6), 22-26.

Boehm, B., Abts, C., & Chulani, S. (2000). Software development cost estimation approaches-a survey. *Annals of Software Engineering , 10* (1-4), 177-205.

Boehm, B., Abts, C., Winsor-Brown, A., Chulani, S., Clark, B., Horowitz, E., … Steece, B. (2000). *Software cost estimation with COCOMO II.* Upper Saddle River, NJ: Prentice Hall.

Bollen, K. A., & Hoyle, R. H. (1990). Perceived cohesion: a conceptual and empirical examination. *Social Forces , 69* (2), 479-504.

Boyatzis, R. E., & Sala, F. (2004). The emotional competence inventory (ECI). In G. Geher (Ed.), *Measuring Emotional Intelligence* (pp. 147-180). Nova Science Publishers.

Boyatzis, R. E., Goleman, D., & Rhee, K. S. (2000). Clustering competence in emotional intelligence: insights from the emotional competence inventory. In R. Bar-On, & J. D. Parker (Eds.), *The handbook of emotional intelligence* (pp. 343-362). San Francisco: Jossey-Bass.

Brooks, F. (1995). *The mythicial man-month* (Anniversary Edition). Boston: Addison-Wesley.

Brown, Bernice B. (1968). DELPHI Process: A methodology used for the elicitation of opinions of experts, RAND, California, P-3925, September 1968.

Bushyacharu, S. (1996). Conflict and conflict management in joint ventures: The Thai partner perspectives. *Asian Institue of Technology Research Study* (April), No. SM-96-50.

Byrne, J. C. (2003). *The role of emotional intelligence in predicting leadership and related work behavior.* Hoboken: Stevens Institute of Technology, Technology Management.

Carless, S. A., & De Paola, C. (2000). The measurement of cohesion in work teams. *Small Group Research , 31* (1), 71-88.

Carron, A. V. (1982). Cohesiveness in sport groups: implications and considerations. *Journal of Sport Psychology , 4*, 123-138.

Carron, A. V., Brawley, L. R., & Widmeyer, N. W. (2002). *The group environment questionnaire test manual.* Morgantown, West Virginia, USA: Fitness Information Technology, Inc.

Carron, A. V., Brawley, L. R., & Widmeyer, N. W. (1998). The measurement of cohesiveness in sport groups. In J. L. Duda (Ed.), *Advances in Sport and Exercise Psychology Measurement* (pp. 213-226). Morgantown, West Virginia, USA: Fitness Information Technology.

Chai, K.-S. (2003). The art, science and engineering of software development in an embedded system. *The 9th Asia-Pacific Conference on Communications (APCC 2003). 1*, pp. 232-237. IEEE.

Charette, R. N. (2005). Why software fails. *IEEE Spectrum* , 42-49.

Cherniss, C. (2000). Emotional intelligence: what it is and why it matters. *Annual Meeting of the Society for Industrial and Organizational Psychology* (p. 14). New Orleans: www.eiconsortium.org.

Cherniss, C. (2001). *The business case for emotional intelligence.* Retrieved 12 15, 2008, from Emotional Intelligence Consortium: http://www.eiconsortium.org/pdf/business_case_for_ei.pdf

Cherniss, C., & Goleman, D. (Eds.). (2001). *The emotionally intelligent workplace.* San Francisco: Jossey-Bass.

Chiang, I. R., & Mookerjee, V. S. (2004). Improving software team productivity. *Communications of the ACM , 47* (5), 89-93.

Clark, B., Devnani-Chulani, S., & Boehm, B. (1998). Calibrating the COCOMO II.2000 post-architecture model. *Proceedings of the 1998 International Conference on Software Engineering* (pp. 477-480). Kyoto, Japan: IEEE.

Connell, J., & Travaglione, T. (2004). Emotional intelligence: a competitive advantage in times of change. *Strategic Change , 13*, 55-59.

Conte, J. M. (2005). A review and critique of emotional intelligence measures. *Journal of Organizational behavior , 26*, 433-440.

Dawson, R. (2000). Twenty dirty trick to train software engineers. *Proceedings of the 22nd International Conference on Software Engineering(ICSE)*, pp. 209-219. ACM Special Interest Group on Software Engineering (SIGSOFT). doi: 10.1145/337180.337204.

Demarco, T. (1997). *The deadline: a novel about project management.* New York: Dorset House.

Demarco, T. (1995). *Why does software cost so much.* New York: Dorset House Publishing.

Demarco, T., & Lister, T. (1999). *Peopleware: productive projects and teams* (2nd Edition ed.). New York: Dorset Publishing.

Demarco, T., & Lister, T. (2003). *Waltzing with bears.* New York: Dorset Publishing.

Druskat, V., & Wolff, S. (1999). The link between emotions and team effectiveness: how teams engage members and build effective task processes. *Academy of Management Proceedings.*

Druskat, V., & Wolff, S. (2001). Building emotional intelligence of groups. *Harvard Business Review , 79* (3), 80-90.

Druskat, V., & Druskat, P. (2006). Applying emotional intelligence in project working. In S. Pryke, & H. Smith, *The management of complex projects a relationship approach* (pp. 78-96). London: Wiley-Blackwell Publishing.

Druskat, V., Sala, F., & Mount, G. (Eds.). (2006). *Linking emotional intelligence and performance at work.* Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

Duke Corporate Education. (2005). *Building effective teams.* Chicago: Dearborn Trade Publishing.

Dulewicz, V., & Higgs, M. (1999). Can emotional intelligence be measured and developed? *Leadership & Organization Development Journal, 20* (5), pp.242-52.

Dulewicz, V., & Higgs, M. (2000). *Emotional intelligence questionnaire, managerial 360-degree version, manual and users' guide*. ASE/NFER-Nelson, Windsor.

Dulewicz, V., & Higgs, M. (2004). Can emotional intelligence be developed. *International Journal of Human Resource Management , 15* (1), 95-111.

Elfenbein, H. A. (2006). Team emotional intelligence: what it can mean and how it can affect performance. In V. U. Druskat, F. Sala, & G. Mount (Eds.), *Linking* *emotional intelligence and performance at work* (pp. 165-184). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Publishers.

Emmerling, R. J. (2008). Toward an applied science of emotional intelligence in the global workplace: key Issues and challenges. In R. J. Emmerling, Shanwal, V. K., & M. K. Mandal (Eds.), *Emotional intelligence theoretical and cultural perspectives* (pp. 69-88). New York, New York, USA: Nova Science Publishers.

Eskerod, P., & Blichfeldt, B. S. (2005). Managing team entrees and withdrawals during the project life cycle. *International Journal of Project Management , 23*, 495-503.

Feist, G. J., & Barron, F. (1996). Emotional intelligence and academic intelligence in career and life success. *Annual Convention of the American Psychological Society.* San Francisco.

Festinger, L. (1950). Informal social communication. *Psychological Review , 57*, 271-281.

Feyerherm, A. E., & Rice, C. L. (2002). Emotional intelligence and team performance: the good, the bad and the ugly. *The International Journal of Organizational Analysis , 10* (4), 343-362.

Fischer, Russell. (1978). The Delphi method: a description, review, and criticism. The Journal of Academic Librarianship, 4(2), 64-70.

Furton, M. T. (2003). Discovering the true cause of failure in custom software development projects. *Computer & Internet Lawyer , 20* (5), 1-3.

Glass, R. (2003). *Facts and fallacies of software engineering.* Boston: Addison-Wesley.

Glass, R. L. (2005). IT failure rates-70% or 10-15%? *IEEE Software , 22* (3), 112+110.

Glass, R. L. (2006). The Standish report: does it really describe a software crisis? *Communications of the ACM , 49* (8), 15-16.

Goleman, D. (1995). *Emotional intelligence: why it can matter more than IQ.* New York: Bantam.

Goleman, D. (1997). *Emotional intelligence.* New York, New York: Bantam Books.

Goleman, D. (1998). *Working with emotional intelligence.* New York: Bantam Books.

Goleman, D. (2000). *Working with emotional intelligence.* New York: Bantam.

Goleman, D. (2001). Emotional intelligence: issues in paradigm building. In C. Cherniss, & D. Goleman (Eds.), *The emotionally intelligent workplace* (pp. 13-26). San Francisco: Jossey-Bass.

Goleman, D. (2004). What makes a leader? *Harvard Business Review , 82* (1), 82-91.

Goleman, D. (2006). *Emotional intelligence* (10th Anniversary reissue ed.). New York: Bantam Books.

Goleman, D. (2007). *Social intelligence: the revolutionary new science of human relationships.* New York: Bantam.

Goleman, D., Boyatzis, R., & McKee, A. (2004). *Primal leadership.* Boston: Harvard Business School.

Goleman, D., Boyatzis, R., & McKee, A. (2001). *Primal leadership: realizing the power of emotional intelligence.* Boston: Harvard Business School Press.

Golembiewski, R. T. (1962). *The small group.* Chicago: University of Chicago Press.

Gorla, N., & Wah Lam, Y. (2004). Who should work with whom: building effective software project teams. *Communications of the ACM , 47* (6), 79-82.

Grenier, A. (2004). Emotional intelligence as a field of research-a comparative study between scientific theory and practice. *Master Thesis, University of Toronto, Ottawa, Canada* . Toronto, Ontario, Canada: National Library of Canada (ISBN 0-612-91422-4).

Groth, R. (2004). Is the software industry's productivity declining? *IEEE Software , 21* (6), 92-94.

Gully, S. M., Devine, D. J., & Whitney, D. J. (1995). A meta-analysis of cohesion and performance: effects of level of analysis and task interdependence. *Small Group Research , 26*, 497-520.

Gustafson, D. A. (2002). *Theory and problems of software engineering.* New York: McGraw-Hill.

Guzzo, R. A., & Dickson, M. W. (1996). Teams in organizations: recent research on performance effectiveness. *Annual Review of Psychology , 47*, 307-340.

Hackman, J. R., & Wagemen, R. (2005). A theory of team coaching. *Academy of Management Review, 30* (2), 269-287.

Hassan, A., & Holt, R. (2003). The chaos of software development. *Proceedings of the Sixth International Workshop on Principles of Software Evolution (IWPSE'03)* (pp. 84-94). IEEE Computer Society.

Hay, & McBer. (1996). *Generic competency dictionary.* Boston: Hay Group.

Hazzan, O., & Tomayko, J. (2005). Reflection and abstraction in learning software engineering's human aspects. *Computer , 38* (6), 39-45.

Heldman, K. (2004). *PMP Project Management Professional study guide* (2 ed.). Alameda, CA: Sybex.

Herrnstein, R. J., & Murray, C. (1994). *The bell curve.* New York: Free Press Paperbacks Book published by Simon & Schuster.

Herrnstein, R. J., & Murray, C. (1996). *The bell curve: intelligence and class structure in american life.* New York: Free Press Paperbacks.

Hersey, P., & Blanchard, K. H. (1969). Life cycle theory of leadership. *Training & Development Journal*, 23(5), 26-34.

Higgs, M. (2005). Influence of team composition and task complexity on team performance. *Team Performance Management, 11* (7/8), 227-250.

Higgs, M. (2003). How can we make sense of leadership in the 21st century? *Leadership and Organization Development Journal , 24* (5), 273-284.

Higgs, M. (2001). Is there a relationship between the Myers-Briggs Type Indicator and emotional intelligence? *Journal of Managerial Psychology , 16* (7), 509-533.

Higgs, M., & Dulewicz, S. (2000). Emotional intelligence, leadership and organisational culture. Linkage Emotional Intelligence Conference.

Higgs, M., & Dulewicz, S. (1999). Making sense of emotional intelligence. NFER Nelson.

Hoegl, M., Weinkauf, K., & Gemuenden, H. G. (2004). Interteam corrdination, project commitment, and teamwork in multiteam R&D projects: a longitudinal study. *Organization Science , 15* (1), 38-55.

Hogan, J. M., & Thomas, R. (2005). Developing the software engineering team. In A. Young, & D. Tolhurst (Ed.), *Australasian computing education conference. 42*, pp. 203-210. Newcastle: Australian Computer Society.

Howey, R. (2002). Understanding software technology. *Knowledge, Technology, & Policy , 15* (3), 70-81.

Humphrey, W. S. (2005). Why big software projects fail: The 12 key questions. *CrossTalk: The Journal of Defense Software Engineering , 18* (3), 25-29.

IEEE Computer Society. (2004). *Guide to the software engineering body of knowledge.* (A. Abran, J. W. Moore, P. Bourque, R. Dupuis, & L. L. Tripp, Eds.) Los Alamitos, CA, USA: IEEE Computer Society.

IEEE Computer Society. (1990). *IEEE Standard Glossary of Software Engineering Terminology.* New York: IEEE.

Jeffery, R., & Scott, L. (2002). Has twenty-five years of empiridcal software engineering made a difference? *Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02)* (pp. 539-546). Australia: IEEE Computer Society.

Jones, C. (2006, June). *Social and technical reasons for software project failures.* Retrieved 11 30, 2008, from CrossTalk-The Journal of Defense Computing: http://www.stsc.hill.af.mil/crosstalk/2006/06/0606Jones.html

Jordan, P. J., & Ashkanasy, N. M. (2006). Emotional intelligence, emotional self awareness, and team effectiveness. In V. U. Druskat, F. Sala, & G. Mount (Eds.), *Linking emotional intelligence and performance at work* (pp. 145-164). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Publishers.

Jordan, P. J., & Troth, A. C. (2004). Managing emotions during team problem solving: emotional intelligence and conflict resolution. *Human Performance, 17* (2), 195-218.

Jorgensen, M., & Molokken, K. (2006). *How large are software cost overruns? A review of the 1994 Chaos report.* Retrieved April 14, 2007, from Simula Research Laboratory: http://www.simula.no/research/engineering/publications/Jorgensen.2006.4/downloadPdfFile

Kang, H.-R., Yang, H.-D., & Rowley, C. (2006). Factors in team effectiveness: cognitive and demographic similarities of software development team members. *Human Relations , 59* (12), 1681-1710.

Karn, J., & Cowling, A. (2005). A study into the effect of disruptions on performance of software engineering teams. *Empircal Software Engineering* .

Kemper, C. L. (1999). EQ vs. IQ. *Communication World , 16* (9), 15-19.

Klein, H. J., & Mulvey, P. W. (1995). Two investigations of the relationships among group goals, goal commitment, cohesion, and performance. *Organizational Behavior and Human Decision Processes , 61*, 44-53.

Knouse, S. (2006). Task cohesion: a mechanism for bringing together diverse teams. International Journal of Management, 23(3), 588 – 596.

Knouse, S. (2007). Building task cohesion to bring teams together. Quality Progress, 40(3), 49 – 53.

Laird, L. M. (2006). The limitations of estimation. *IT Pro* , 40-45.

Leban, W. (2004). Linking emotional intelligence abilities and transformational leadership styles. *Leadership And Organization Development Journal , 25* (7), 554-564.

Leban, W. V. (2003). The relationship between leader behavior and emotional intelligence of the project manager and the success of complex projects. *Doctoral dissertation, Benedictine University, 2003* . Chicago, IL, USA: Proquest Information and Learning Company UMI 3092853.

Linstone, H.A. & Turoff, M. (2002). Introduction [to chapter I]. In H.A. Linstone & M. Turoff (Eds.), The Delphi method. Techniques and applications (pp. 3-12). Retrieved March 6, 2011, from http://is.njit.edu/pubs/delphibook/.

Lott, A. J., & Lott, B. E. (1965). Group cohesiveness as interpersonal attraction: A review of relationships with antecedents and consequent variables. *Psychological Bulletin , 64*, 259-309.

Lum, K., Bramble, M., Hihn, J., Hackney, J., Khorrami, M., & Monson, E. (2003). *Handbook for software cost estimation.* Jet Propulsion Laboratory. Pasadena: National Aeronautics and Space Administration (NASA).

Lum, K., Hihn, J., & Menzies, T. (2006). Studies in software cost model behavior: do we really understand cost model performance?. Jet Propulsion Laboratory. Pasadena: National Aeronautics and Space Administration (NASA). Retrieved from http://hdl.handle.net/2014/41450 on 8 May, 2011.

Marks, M. A., Mattieu, J. E, and Zacarro, S. J. (2001), A temporally based framework and taxonomy of team processes. Academy of Management Review, 26 (3), 356-376.

Matthews, G. Z., & Roberts, R. (2007). *The science of emotional intelligence.* Oxford: Oxford University Press.

Matthews, G., Roberts, R. D., & Zeidner, M. (2004). Seven myths about emotional intelligence. *Psychological Inquiry , 15* (3), 179-196.

Matthews, G., Zeidner, M., & Roberts, R. (2004). *Emotional intelligence: science & myth.* Cambridge, MA: MIT Press.

Matthews, G., Zeidner, M., & Roberts, R. (2007). *The science of emotional intelligence.* (G. Matthews, M. Zeidner, & R. Roberts, Eds.) New York: Oxford University Press, Inc.

Maxwell, J. C. (2001). *The 17 indisputable laws of teamwork.* Nashville: Nelson, Inc.

Mayer, J. D., Caruso, D. R., & Salovey, P. (1999). Emotional intelligence meets traditional standards for an intelligence. *Intelligence , 27*, 267-298.

Mayer, J. D., Salovey, P., & Caruso, D. (2000a). Models of emotional intelligence. In R. Sternberg (Ed.), *Handbook of intelligence.* Cambridge University Press.

Mayer, J. D., Salovey, P., & Caruso, D. R. (2000b). Emotional intelligence as zeitgeist, as personality, and as a mental ability. In R. &. Bar-On, & R. &. Bar-On (Ed.), *The handbook of emotional intelligence* (pp. 92-117). San Francisco: Jossey-Bass.

Mayer, J. D., Salovey, P., & Caruso, D. R. (2004). Emotional intelligence: theory, findings, and implications. *Psychological Inquiry , 15* (3), 197-215.

McConnell, S. (2006). *Software estimation: demystifying the black art.* Redmond, WA: Microsoft Press.

McConnell, S. (2004). Professional software development Boston: Pearson Education Inc.

McConnell, S. (1998). The art, science, and engineering of software development. *IEEE Software , 15* (1), 118-120.

McDermid, J., & Rook, P. (1993). Software development process models. In J. A. McDermid, *Software engineer's reference book.* (pp. 26-28). Boca Raton: CRC Press.

McDonald, S., & Edwards, H. (2007). Who should test whom: examining the use and abuse of personality tests in software engineering. *Communications of the ACM , 50* (1), 67-71.

Mills, H. (1999). The management of software engineering part 1: principles of software engineering. *IBM Systems Journal , 38* (2&3), 289-295.

Molokken, K., & Jorgensen, M. (2003). A review of surveys on software effort estimation. *Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03)*, (p. 223).

Morgan, J. (2005). Why the software industry needs a good ghostbuster. *Communications of the ACM , 48* (8), 129-133.

Mudrack, P. E. (1989). Defining group cohesiveness: a legacy of confusion? *Small Group Behavior , 20* (1), 37-49.

Mullen, B., & Copper, C. (1994). The relationship between group cohesiveness and performance: An Integration. *Psychological Bulletin , 115*, 210-227.

Murdoch, S. (2007). IQ: A smart history of a failed idea [Kindle DX version]. Hoboken, NJ:John Wiley & Sons, Inc. Retrieved from Amazon.com.

Murray, J. P. (2001). Recognizing the responsibility of a failed information technology project as a shared failure. *Information Systems Management , 18* (2), 25-29.

Musilek, P., Pedrycz, W., Sun, N., & Succi, G. (2002). On the sensitivity of COCOMO II.2000 software cost estimation model. *Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS'02)* (pp. 13-20). Edmonton, Alberta, Canada: IEEE.

Myers, I.B., & McCaulley, M. (1998). *MBTI manual: a guide to the development and use of the Myers Briggs Type Indicator* (3 ed.). Palo Alto, CA, USA: Consulting Psychologists Press.

Nasir, M. (2006). A survey of software estimation techniques and project planning practices. *Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing* (pp. 305-310). Las Vegas, NV: IEEE Computer Society.

Navare, J. (2003). Process or behavior: which is the risk and which is to be managed? *Managerial Finance , 29* (5/6), 6-19.

Neisser, U., Boodoo, G., Bouchard, T. J., Boykin, A. W., Bordy, N., Ceci, S.J., . . . Urbina, S. (1996). Intelligence: knowns and unknowns. *American Psychologist, 51* (2), 77-101.

Nemeth, C. J., & Staw, B. (1989). The tradeoffs of social control and innovation in small groups and organizations. *Advances in Experimental Social Psychology , 22*, 175-210.

Nguyen, V., Steece, B., & Boehm, B. (2008). *A constrained regressiontTechnique for COCOMO calibration.* Retrieved Nov 12, 2008, from Center for Systems and Software Engineering Technical Report: http://sunset.usc.edu/csse/TECHRPTS/2008/usc-csse-2008-806/usc-csse-2008-806.pdf

Oriogun, P. K. (1999). A survey of Boehm's work on the spiral models and COCOMO II-towards software development process quality improvement. *Software Quality Journal , 8*, 53-62.

Okoli, C. & Pawlowski, S. D. (2004). The delphi method as a research tool: an example, design consideratoins and applications. *Information & Management , 42*, 15-29.

Paskevich, D. M. (1995). Conceptual and measurement factors of collective efficacy in its relationship to cohesion and performance outcome. *Unpublished doctoral dissertation, University of Waterloo, Waterloo, Canada .*

Patanakul, P., & Milosevic, D. (2009). The effectiveness in managing a group of mulitple projects: factors of influence and measurement criteria. *International Journal of Project Management , 27*, 216-233.

Peslak, A. R. (2005). Emotions and team projects and processes. *Team Performance Management , 11* (7/8), 251-262.

Peslak, A. R. (2006). The impact of personality on information technology team projects. *Proceedings of the 2006 ACM SIGMIS CPR Conference on Computer Personnel Research: Forty Four Years of Computer Personnel Research: Achievements, Challenges & the Future* (pp. 273-279). Claremont: ACM Press.

Pillis, E., & Furumo, K. (2006). Virtual vs. face-to-face teams: deadbeats, deserters, and other considerations. *Special Interest Group on Management Information Systems-Computer Personnel Research* (pp. 318-320). Claremont, California: ACM Press.

Poore, J. H. (2004). A tale of three disciplines...and a revolution. *Computer , 37* (1), 30-36.

Potok, T.E., Vouk, M.A. (1999) A model of correlated team behavior in a software development environment. IEEE symposium on application-specific systems and software engineering and technology, ASSET 99, Richardson, proceedings

Pour, G., Griss, M., & Lutz, M. (2000). The push to make software engineering respectable. *IEEE Computer , 33* (5), 35-43.

Pressman, R. S. (2004). Development strategies and project management. In A. B. Tucker, *Computer science handbook* (pp. 108.1-108.21). Boca Raton, FL: The Association for Computing Machinery (ACM).

Project Management Institute. (2004). *A guide to the project management body of knowledge* (3rd ed.).

Rainey, D.W., & Schweickert, G.J. (1988). An exploratory study of team cohesion before and after a spring trip. Sport Psychologist, 2(4), 314-317.

Rapisarda, B. A. (2002a). The impact of emotional intelligence on work team cohesiveness and performance. *The International Journal of Organizational Analysis , 10* (4), 363-379.

Rapisarda, B. A. (2002b). The impact of emotional intelligence on work team cohesiveness and performance. *Doctoral dissertation, Case Western Reserve University* . Cleveland, Ohio, USA: ProQuest Information and Learning UMI 3066076.

Rifkin, S. (2001). What makes measuring software so hard? *IEEE Software , 18* (3), 41-45.

Robson, C. (2002). Real world research. Carlton,Victoria, Australia: Blackwell Publishing.

Rubinstein, D. (2007, Mar 1). *Standish group report: there's less development chaos today.* Retrieved Apr 5, 2007, from SD Times: http://www.sdtimes.com/article/story-20070301-01.html

Saeki, M. (1995). Communication, collaboration and cooperation in software development-how should we support group work in software development? *Asia Pacific Software Engineering Conference, 1995* (pp. 12-20). Brisbane, Qld.: IEEE.

Salas, E., Burke, C. Fowlkes, J., & Priest, H. (2004) On measuring teamwork skills. In M. Herson (Series Ed.) & J. C. Thomas, (Vol Ed.), *Comprehensive handbook of psychological assessment: Vol 4. industrial and organizational assessment* (pp. 427 – 442). Hoboken, NJ: John Wiley & Sons, Inc.

Salovey, P., & Mayer, J. D. (1990). Emotional intelligence. *Imagination, Cognition and Personality , 9*, 185-211.

Sandvig, J.S. (2008). An examination of the relationship between emotional intelligence and an individual's perception of team cohesiveness (Doctoral dissertation). Available from ProQuest Dissertations and Theses database. (UMI No. 3311295).

Sawyer, S. (2004). Software Development Teams. *Communications of the ACM , 47* (12), 95-99.

Sawyer, S., & Guinan, P. J. (1998). Software development: processes and performance. *IBM Systems Journal , 37* (4), 552-569.

Schachter, S., Ellertson, J., McBride, D., & Gregory, D. (1951). An experimental study of cohesiveness and productivity. *Human Relations , 4*, 229-238.

Schulze, R., & & Roberts, R. D. (2005). *Emotional intelligence an international handbook.* (R. Schulze, & R. D. & Roberts, Eds.) Cambridge, MA, USA: Hogrefe & Huber Publishers.

Sikes, B., Gulbro, R., & Shonesy, L. (2010). Conflict in work teams: problems and solutions. *Proceedings of the Academy of Organizational Culture, Communications and Conflict*, *New Orleans, 2010, 15*(1). pp. 48-52.

Skulmoski, G. J., Hartman, F. T., & Krahn, J. (2007). The Delphi method for graduate research. *Journal of. Information Technology Education*, 6.

Smith, J. L., Bohner, S. A., & McCrickard, D. S. (2005). Project management for the 21st Century: supporting collaborative design through risk analysis. *43rd ACM Southeast Conference (ACMSE '05)*, (pp. 2.300-2.305). Kennesaw, GA.

Somech, A., Desivilya, H. S., & Lidogoster, H. (2008). Team conflict management and team effectiveness: the effects of task interdependence and team identification. *Journal of. Organizational Behavior*, doi: 10.1002/job.

Standish Group. (1999). *Chaos: A recipe for success.* Retrieved 09 12, 2004, from Standish Group Sample Research: http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf

Standish Group. (2001). *Extreme chaos.* Retrieved 09 12, 2004, from Standish Group Sample Research: http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf

Standish Group. (2002). *Show me the money: return on inspection.* Retrieved from Standish Group Sample Research.

Sternberg, R. (1996). *Successful intelligence.* New York: Simon & Schuster.

Stevens, K. T. (1998). The effects of roles and personality characteristcs on software development team effectiveness. Doctoral dissertation, Virginia Polytechnic Institute and State University.

Stewart, J. (2001). Is the Delphi technique a qualitative method? *Medical Education, 35*(10), 922-923.

Summers, I., Coffelt, T., & Horton, R. E. (1988). Work-group cohesion. *Psychological Reports , 63*, 627-636.

Sunindijo, R. Y., & Hadikusumo, B. H. (2005). Benefits of emotional intelligencee to project management: a study of leadership and conflict resolution style of project managers in thailand. In A. C. Sidwell (Ed.), *The Queensland University of Technology (QUT) Research Week International Conference.* Brisbane, Australia.

Tellioglu, H., & Wagner, I. (1999). Software cultures: exploring cultural practices in managing heterogeneity within system design. *Communications of the ACM , 42* (12), 71-77.

Thompson, J. B., & Reed, K. (2005). The mark of a discipline. *IEEE Software , 22* (6), 96-97.

Thorndike, E. L. (1920). Intelligence and its use. *Harper's Magazine , 140*, 227-235.

Tiwana, A., & Keil, M. (2004). One-minute risk assessment tool. *Communications of the ACM , 47* (11), 73-77.

Tomayko, J., & Hazzan, O. (2004). *Human aspects of software engineering.* Hingham, MA: Charles River Media, Inc.

Tuleya, L. G. (Ed.). (2007). *Thesaurus of psychological index terms* (11 ed.). American Psychological Association (APA).

Tuckman, B. W., and M. A. Jensen. (1977). Stages of small-group development revisited. Group and Organization Studies 2 (4), 419-27.

Tullet, A. D. (1996). The thinking style of the managers of multiple projects: implications for problem solving when managing change. *International Journal of Project Management , 14*(5), 281-287.

Twaites, G., & Sibilla, M. (2002). Software engineering in a SEI Level-5 organization. *International Journal of Reliability, Quality and Safety Engineering , 9* (4), 347{365.

Valerdi, R., & Boehm, B., (2010). COSYSMO: A systems engineering cost model. *Genie Logiciel, 92*, 2-6.

Van Vliet, H. (2008). *Software engineering principles and practices* (3rd ed.). Chichester, West Sussex, England: John Wiley & Sons Ltd.

Wallace, L., & Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM , 47* (4), 68-73.

Wech, B. A., Mossholder, K. W., Steel, R. P., & Bennett, N. (1998). Does work group cohesiveness affect individuals' performance and organizational commitment? A cross-level examination. *Small Group Research , 29* (4), 472-494.

Wellington, C. A., Briggs, T., & Girard, C. D. (2005). Examining team cohesion as an effect of software engineering methodology. *ACM SIGSOFT Software Engineering Notes , 30* (4), 1-5.

Wolff, S. B. (2005). *Emotional Competence Inventory (ECI) technical manual.* Hay Group McClelland Center for Research and Innovation. Retrieved from http://www.eiconsortium.org/pdf/ECI_2_0_Technical_Manual_v2.pdf

Wolff, S. B., Druskat, V. U., Koman, E. S., & Messer, T. E. (2006). The link between group emotional competence and group effectiveness. In V. U. Druskat, F. Sala, & G. Mount (Eds.), *Linking emotional intelligence and performance at work* (pp. 223-242). Mahwah, New Jersey, USA: Lawrence Erlbaum Associates, Publishers.

Worchel, S., Cooper, J., & Goethals, G. R. (1991). *Understanding social psychology* (5th ed.). Pacific Grove: Brooks/Cole.

Xia, W., & Lee, G. (2004). Grasping the complexity of IS development projects. *Communications of the ACM* , *47* (5), 69-74.

Zuse, H. (1998). *A framework of software measurement.* Berlin, Germany: Walter de Gruyter & Co.

APPENDIX A. E-MAIL FOR FIRST ROUND

Dear Panel Members,

Thank you all again for agreeing to participate as members of this Delphi panel to determine the most effective stratagem to improve the team cohesion measure within the COCOMO II.2000 estimation model.

INSTRUCTIONS:
The process for this Delphi study will include three rounds of e-mail questionnaires. The second and final e-mail will have a summary of previous responses attached for review and comment. Responses are due to the researcher two weeks after each e-mail is received. Upon receipt of all responses the researcher will take two weeks to create a summary of the responses and send the next e-mail with the summary attached. This process will continue for a total of three e-mail rounds. The panelists will return responses two weeks after receiving each e-mail and the researcher will return a summary and additional questions two weeks after receiving the responses. The third and final e-mail will provide panelists a final opportunity for parting comments.

A copy of the first two chapters of this study is attached to this first e-mail study. The literature review presents the foundational information for this study. The goal of this study is to determine the effectiveness of the COCOMO II.2000 TEAM variable. COCOMO II.2000 is a software estimation model and the TEAM variable represents the team cohesion measure of the software development team that will complete the project under estimation review. As stated in the literature review the TEAM measure is assessed by the project manager's subjective answers to four simple questions. Again, the goal of this study is to determine if the TEAM variable and its survey model (the four questions) are effective in light of the literature review. Additionally, two other team cohesion models will be reviewed for effectiveness. Copies of the cited articles can be viewed at the following website: www.bascresearch.com. If you are unable to view a copy of any of the articles please let the researcher know.

Please provide thorough and succinct answers to each question and forward responses within 14 days after receipt of the e-mail. Timely responses will keep this process within a minimal time frame. If for whatever reason you are unable to meet the response time please e-mail the researcher immediately so arrangements can be made.

Again, thank you for your participation in this study!

QUESTIONS:

1. Do you believe this research has provided enough evidence to warrant the investigation of the effectiveness of the COCOMO II.2000 TEAM variable? Why or Why not?

2. In light of the team cohesion literature review within this research do you believe the COCOMO II.2000 TEAM variable effectively measures team cohesion? Why or Why not?

3. Do you believe the project manager's subjective assessment of the software development team is an effective method to measure team cohesion? Why or Why not?

4. Do you believe the four survey questions used to determine the COCOMO II.2000 TEAM variable's value effectively assesses a software development team's team cohesion measurement? Why or Why not?

5. Based upon your expertise and the literature review of this study does the COCOMO II.2000 TEAM variable take into consideration how team cohesion changes over the life cycle of the project? Why or Why not?

6. What suggestions, if any, would be the best way to introduce the dynamic phenomena of team cohesion changing over the life-cycle of the project into the current COCOMO II.2000 model?

7. In your expert guidance how important is task cohesion as a factor of success in software development? Why or Why not?

Thank you for taking the time to participate in this study!

Respectfully,

Betty A. Carden
PhD Candidate, IT Management
School of Business and Technology
Capella University

APPENDIX B. E-MAIL FOR SECOND ROUND

Dear Panel Members,

Thank you all again for agreeing to participate as members of this Delphi panel to determine the most effective stratagem to improve the team cohesion measure within the COCOMO II.2000 estimation model. Copies of the cited articles can be viewed at the following website: www.bascresearch.com.

INSTRUCTIONS:
This is the second of three e-mails. Attached is a list of survey questions taken from the ECIv2.0 and the GEQ surveys. The GEQ is a team cohesion instrument and the ECIv2.0 questions have been directly associated with team cohesion (Rapisarda, 2000b). Please provide thorough and succinct answers to each question in this e-mail and forward responses within 14 days after receipt of this e-mail.

QUESTIONS:
1. The ECI model was used in previous research which determined nine emotional intelligence competencies strongly correlated with team cohesion (Rapisarda, 2002b). The survey questions associated with these nine competencies are attached to this e-mail. In your opinion would these questions create a more effective method in determining the team cohesion value for the COCOMO II.2000 TEAM variable? Why or Why not?

2. In your opinion, would the questions of the GEQ survey modified for software engineers which directly correlate to team cohesion (Wellington, Briggs, & Girard, 2005), which is attached to this e-mail, be a more effective method in determining the team cohesion value for the COCOMO II.2000 TEAM variable? Why or Why not?

3. Based upon your expertise and the research literature presented in this study how often should a team cohesion survey be administered during the life cycle of a software development project? Why?

4. When a team member exits and a replacement is brought on board when should team cohesion be measured? When the team member exits? Soon after the team member exits? After the replacement is hired but before they report for work? Soon after the replacement is hired? Why or Why not?

Again, thank for taking the time to participate in this study!

Respectfully,

Betty A. Carden
PhD Candidate, IT Management
School of Business and Technology
Capella University

APPENDIX C. E-MAIL FOR THIRD AND FINAL ROUND

Dear Panel Members,

Thank you all again for agreeing to participate as members of this Delphi panel to determine the most effective stratagem to improve the team cohesion measure within the COCOMO II.2000 estimation model. Copies of the cited articles can be viewed at the following website: www.bascresearch.com.

INSTRUCTIONS:
This is the third and final e-mail for this study. Please provide thorough and succinct answers to each question and any parting comments in this e-mail and forward responses within 14 days after receipt of this e-mail.

QUESTIONS:

1. In your expert guidance, what research should be accomplished next to improve software estimation? How can the use of emotional intelligence assessments assist in improving software estimation?

2. This is the final e-mail. You have received a summary of the results to date of the panel. Do you have any further comments?

Again, thank for taking the time to participate in this study!

Respectfully,

Betty A. Carden
PhD Candidate, IT Management
School of Business and Technology
Capella University

APPENDIX D. OVERVIEW OF RESEARCH PERTAINING TO THE VALIDITY OF THE GEQ

Dependent Variable	Type of Validity	Result	Reference
GEQ Items	Content	+	Carron et al. (1985, phase 2)
Sport Cohesion Questionnaire	Concurrent	+	Brawley et al. (1987, Study 1)[a]
Team Climate Inventory	Concurrent	+	Brawley et al. (1987, Study 1)[b]
Bass Inventory	Concurrent	+	Brawley et al. (1987, Study 1)[c]
Adherence Behavior in Exercise Groups			
Drop out	Predictive	+	Carron et al. (1988, Study 1)
Drop out	Predictive	+	Spink & Carron (1993)
Attendance	Predictive	+	Spink & Carron (1994, Study 1)
Attendance	Predictive	+	Spink & Carron (1994, Study 2)
Absenteeism	Predictive	+	Spink & Carron (1992)
Lateness	Predictive	+	Spink & Carron (1992)
Early Exit	Predictive	–	Spink & Carron (1993)
Adherence Behavior in Sport Groups			
Absent/Late	Predictive	+	Carron et al. (1988, Study 2)
Resistance to Disruption			
Sport	Predictive	+	Brawley, Carron, & Widmeyer (1998, Study 1)
Sport	Predictive	+	Brawley et al. (1988, Study 2)[d]
Exercise	Predictive	+	Brawley et al. (1988, Study 2)
Attributions	Predictive	+	Brawley et al. (1987, Study 3)
Social Loafing	Predictive	+	Naylor & Brawley (1992)
	Predictive	+	McKnight, Williams, & Widmeyer (1991)
Group Size			
Exercise	Predictive	+	Carron & Spink (1995, Study 1)
Exercise	Predictive	+	Carron & Spink (1995, Study 2)
Exercise	Predictive	+	Carron & Spink (1995, Study 3)
Sport	Predictive	+	Widmeyer, Brawley, & Carron (1990, Study 1)
Leadership	Predictive	+	Westre & Weiss (1991)
	Predictive	+	Widmeyer & Williams (1991)
Team Building			
Sport	Predictive	+	McClure & Foster (1991)
Sport	Predictive	+	Prapavessis, Carron, & Spink (1996)
Exercise	Predictive	+	Carron & Spink (1993)
Exercise	Predictive	+	Spink & Carron (1993)
Role Involvement			
Clarity	Predictive	+	Dawe & Carron (1990)
Clarity	Predictive	+	Grand & Carron (1982)
Acceptance	Predictive	+	Dawe & Carron (1990)
Acceptance	Predictive	+	Grand & Carron (1982)
Collective Efficacy	Predictive	+	Paskevich (1995)
Communication	Predictive	–	Huntley & Carron (1992)
	Predictive	+	Widmeyer & Williams (1991)
	Predictive	+	Widmeyer et al. (1993)
Coordination	Predictive	–	Huntley & Carron (1992)
	Predictive	–	Widmeyer et al. (1993)

Dependent Variable	Type of Validity	Result	Reference
Team sport vs. Individual sport	Predictive	+	Brawley et al. (1987, Study 2)
Duration of membership	Predictive	–	Brawley et al. (1987, Study 2)
Group Environment Questionnaire			
	Factorial	+	Carron et al. (1985, Phase 4)
	Factorial	–	Schutz et al. (1994)
	Factorial	–	Kozub (1993)
	Factorial	+	Li & Harmer (1996)

Note. The + sign indicates support from the research for the conceptual model: the – sign indicates a lack of support.(a) Six of eight analyses supported predictions, (b) Nine of 12 analyses supported predictions, (c) Twenty-eight of 32 analyses supported predictions, and (d) One of two analyses supported predictions. From G.E.Q. *The Group Environment Questionnaire Test Manual* (pp.33-34) by A. V. Carron, L. R. Brawley, & N. W. Widmeyer, 2002, Morgantown, WV: Copyright 2002 by Fitness Information Technology, Inc. Reprinted with permission.