Performing Collaborative, Distributed Systems Engineering (CDSE): Lessons Learned from CDSE Enterprises

by

Darlene Ann Utter

B.S. Aerospace Engineering and Information Technology, Massachusetts Institute of Technology (2004)

Submitted to the Engineering Systems Division in partial fulfillment of the requirements for the degree of

Master of Science in Engineering Systems

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
FEB 28 2007
LIBRARIES

© Massachusetts Institute of Technology 2007. All rights reserved.

BARKER

Author ....
Engineering Systems Division
January 19, 2007

Certified by ....
Dr. Deborah Nightingale
Professor of the Practice, Aeronautics & Astronautics and Engineering Systems
Thesis Supervisor

Certified by ....
Dr. Donna H. Rhodes
Senior Lecturer, Engineering Systems Division
Thesis Supervisor

Accepted by ....
Dr. Richard de Neufville
Professor of Engineering Systems, Chair ESD Education Committee

Performing Collaborative, Distributed Systems Engineering (CDSE): Lessons Learned from CDSE Enterprises

by

Darlene Ann Utter

Submitted to the Engineering Systems Division
on January 19, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Engineering Systems

Abstract

The United States aerospace and defense budgets are shrinking, resources are scarce and requirements are more demanding; aerospace and defense enterprises are expected to deliver a more capable product in less time and with fewer resources. To achieve this tough mission, the enterprises that comprise the United States aerospace and defense industries must form strategic partnerships and collaborations to utilize their respective resources, knowledge, and expertise to meet their customers' needs. Collaboration, be it between competing companies or within different divisions of the same company, is necessary for the survival of each company and the defense industry. In the past, United States aerospace and defense company relationships consisted mostly of a prime contractor, with sub-contractors providing a specific hardware or software subsystem, as specified by the prime contractor. Today, aerospace and defense company relationships are moving more toward that of "partners" where the previous supplier or sub-contractor for hardware or software subsystems is now sharing in the overall system design and engineering efforts. Since the partner companies and intra-company divisions are still geographically distributed throughout the United States, it is necessary for the aerospace and defense contractors to perform collaborative, distributed systems engineering (CDSE) over several geographical locations.

Previous research has demonstrated that the design practices of distributed design teams differ from those of traditional, co-located teams. However, many companies today are performing CDSE using systems engineering (SE) processes and methods developed for traditional SE environments and are therefore encountering many issues. Successful SE practices are difficult to carry-out when performed by a traditional, co-located enterprise. The addition of geographic distribution and cross-company or intra-company collaboration in SE presents a myriad of social and technological challenges that necessitate new and different SE methods for success.

Best practices for CDSE are currently unknown (or undocumented). In an attempt to benchmark the current state of CDSE practices in industry, this research presents the collection of CDSE lessons learned and success factors gathered from two case studies carried out at two United States aerospace and defense companies. The case studies examine many different factors that pertain to the companies' current CDSE efforts, including collaboration scenarios; collaboration tools; knowledge and decision management; SE practices and processes; SE process improvements; SE culture; SE project management, SE organization; and SE collaboration benefits and motivation. Since the research for successful CDSE practices is in its infancy, this research also outlines key areas for future CDSE research.

Thesis Supervisor: Dr. Deborah Nightingale
Title: Professor of the Practice, Aeronautics & Astronautics and Engineering Systems

Thesis Supervisor: Dr. Donna H. Rhodes
Title: Senior Lecturer, Engineering Systems Division

Acknowledgments

Since I've always wanted to be a real, live rocket scientist. I have always been fond of the expression, "Shoot for the moon, even if you miss, you will land among the stars." I consider myself so very lucky to be surrounded by so many stars! I don't even know how to begin expressing my gratitude to all those who helped me start my graduate school journey, and those who have helped me along the way...

My sincerest thanks to my research advisor, mentor, role model, and friend, Dr. Donna Rhodes. Her advice, assistance, patience, creativity, kindness, and thoughtfulness have truly been an inspiration and my guide for the past 18 months. I hope to work with her again soon and will miss her immensely!

My academic advisor, Professor Debbie Nightingale, is another role model and mentor I will sorely miss. I can't thank her enough for providing me with a wonderful group of people to work with, and for always offering her advice and kind words. Her passion for her work and her perseverance are inspiring.

Without the support of my managers and supervisors at Raytheon, my graduate work at MIT would not be possible. Sincerest thanks to: Dan Dechant, Mike Earley, Rich Ricca, and Mark D' Valentine for their assistance in obtaining my scholarship and their sponsorship and support of my graduate degree. I would also like to thank Bob Guerin and Trish Semeter, my new managers, for their continued support. Also, a special thanks to Nina Caissie and Beth Wilson for their additional assistance.

This thesis would not be possible without the enthusiasm and support of the my company sponsors and thesis reviewers. I pledged to keep all sponsors and participants anonymous, so I cannot list out their names here...but I cannot thank them all enough for their time, patience, assistance, and willingness to share some of their experiences and resources with me. I greatly enjoyed getting to meet all of my interviewees and every interview I went on. I learned so much from all of you. I want to specifically thank the central contact persons and their assistants at both Company A and Company B - your help in particular was invaluable to gathering the information in these studies.

A very special thanks to all of the faculty and staff of the LAI, as well as my labmates and fellow LAI classmates.

To my co-workers, supervisors, and amazing friends at Raytheon: Bob Crane, Steve Labitt, and Gennaro Ledonne. I don't even know where to begin thanking each one of you for your assistance and support in obtaining my scholarship and also for supporting me while at school. You guys have been such great mentors - thanks for giving me a chance! I would also like to especially thank Jim Barry and Bob Busch for their assistance in obtaining my scholarship. And of course, a very special thanks to Susan Muldoon, who has helped me with so many things at Raytheon and who has been a wonderful listener and friend!

To my (and Larry's) good friends... Caroline Lamb, Kathleen McCoy, Kathleen Rubritz, Justin McBride, and Joe and April Cattell. You have all been so helpful and supportive these past several months, and I can't thank you enough!

Larry #1 (as requested): To Larry Hawe (II), for his many hours of help with LaTeX and BibTeX (even though he MADE me use them). I really do appreciate all your help, even though you made me write this thank you.

To all of my future in-laws. I can't thank you enough for your support. Thank you Larry, Karla, Julie, Tony, Grandma and Grandpa Langford (you guys were excellent painters!), Jamie, and Alisha. To Grandma Stowe and Auntie Dolores, I am so happy you both decided to stay with us for the month of November. I can't even begin to thank you both for your help and support while I was writing most of my thesis. I am so excited to be a part of your family.

To my extended family...Grandma Barbara, Grandpa Luddy, Carolyn, Grandma Liz, Grandma Nancy, all my aunts and uncles, cousins, and Chris and Kevin - thank you for your love and support all these years!

To my immediate family - You have all "supported" me in so many ways through the past several years...and rough years they have been. To my brothers Brian and Craig, thank you for your many prank phone calls laughs while I was so busy. To my "twin" sister Kristi-lyn, you are beautiful, talented, kind, and amazing - I can't even begin to thank you for all your support. And of course to my parents, James and Debra. You are both such amazing and loving people, who have given me so much so unselfishly. I can't even begin to thank you for all the sacrifices you made to send me to MIT, and all your encouragement and support while I have been here for most of the past 6 years. You are all truly amazing people and I love you so very very much.

Larry #2: Finally, to my fiance Larry Hawe, the man I plan on spending the rest of my life with. I am speechless (and that is a very difficult feat to accomplish)! Wow. I can't even find the words to tell you how grateful I am for all your love and support this past 18 months and the past 5+ years. You are truly amazing - your kindness, sense of humor, diligence, and intelligence overwhelm me sometimes. I am so so so very proud of you and all the hard work you have done in our house. There are so many things I look forward to with you - our wedding, children, and traveling the world! So long as we have each other - we can do anything. I love you so much.

This thesis is dedicated to the "loves" of my life: my family, Larry, and in loving memory of Christina Hawe-Mullan.

Contents

List of Figures

List of Tables

List of Acronyms

AF - Air Force
AITS - Action Item Tracking System
AP - Application Protocol
CAD - Computer Aided Drafting
CBA - Collaborative Based Acquisition
CDR - Critical Design Review
CDSE - Collaborative, Distributed Systems Engineering
CEE - Collaborative Engineering Environment
CM - Configuration Managed
CMMI - Capability Maturity Model Integration
COTS - Commercial Off the Shelf
COUHES - Committee on the Use of Humans as Experimental Subjects
CPT - Cross Product Team
CSCW - Computer Supported Collaborative Work
CST - Central Standard Time
DARPA - Defense Advanced Research Projects Agency
DSE - Decision Support Environment
EMT - Engineering Model Tool
ESD - Engineering Systems Division
EST - Eastern Standard Time
EVMS - Earned Value Management System
FDMT - Formal Data Management Tool
GCSS - Group Communication Support Systems
GDSS - Group Decision Support Systems
IDMA - Interoperability Data Management and Analysis
IDMT - Informal Data Management Tool
IEE - Integrated Engineering Environment IMS - Integrated Master Schedule
INCOSE - International Council on Systems Engineering
IPT - Integrated Product Team
ISO - International Organization for Standardization
IT - Information Technology
ITAR - International Traffic in Arms Regulations
KM - Knowledge Management
KMPI - Knowledge Management Performance Index

LCD - Liquid Crystal Display
MILSTD - Military Standard
MIT - Massachusetts Institute of Technology
MST - Mountain Standard Time
PAW - Product Architecture Workshop
PDR - Preliminary Design Review
PGP - Pretty Good Privacy
PM - Program Manager/Program Management
PST - Pacific Standard Time
PTS - Problem Tracking System
RCRB - Requirements Change Review Board
RDP - Requirements Development Plan
SBA - Simulation Based Acquisition
SCCB - System Configuration Control Board
SE - Systems Engineering
SECM - Systems Engineering Collaboration Maturity
SEDRES - Systems Engineering Data Representation and Exchange Standardization Project
SEMP - Systems Engineering Management Plan
SME - Subject Matter Experts
SRDA - System Requirements, Design and Analysis
SoS - Systems of Systems
SOW - Statement of Work
SWE - Software Engineering
TD - Technical Directors
TP-4 - Technical Panel on Systems Engineering for Defense Modernization
TPDS - Toyota Product Development System
TTCP - Technical Cooperation Program
UK - United Kingdom
US - United States
VPN - Virtual Private Network
WET - Western European Time
WTRT - Web-based Trouble Reporting Tool

Chapter 1

Introduction and Overview

The United States aerospace and defense budgets are shrinking, resources are scarce and requirements are more demanding; aerospace and defense enterprises are expected to deliver a more capable product in less time and with fewer resources. To achieve this tough mission, the enterprises that comprise the United States aerospace and defense industries must form strategic partnerships and collaborations to utilize their respective resources, knowledge, and expertise to meet their customers' needs. Collaboration, be it between competing companies or within different divisions of the same company, is necessary for the survival of each company and the defense industry. In the past, United States aerospace and defense company relationships consisted mostly of a prime contractor, with sub-contractors providing a specific hardware or software subsystem, as specified by the prime contractor. Today, aerospace and defense company relationships are moving more toward that of "partners" where the previous supplier or sub-contractor for hardware or software subsystems is now sharing in the overall system design and engineering efforts. Since the partner companies and intra-company divisions are still geographically distributed throughout the United States, it is necessary for the aerospace and defense contractors to perform collaborative, distributed systems engineering (CDSE) over several geographical locations.

This chapter introduces the work described in this thesis, including the CDSE problem definition, problem motivation, and the research objectives and questions. Additionally, this chapter provides a research and thesis overview.

1.1 Problem Definition: CDSE Motivational Example

Consider the following hypothetical meeting that is based on my personal experiences and that of my colleagues as systems engineers at a large United States aerospace and defense company performing CDSE:

> Yesterday, the systems engineering (SE) lead for Product Z sent an email to 25 systems engineers on her team (the "Z-team") announcing a two hour classified meeting at 11AM today to discuss the interface design issues with two of Product Z's subsystems (subsystem A and B). The "Z-team" is a CDSE team comprised of approximately fifty systems engineers at two geographic locations in the United States: Boston, Massachusetts and Houston, Texas. There are approximately fifteen systems engineers from Company X in Boston working on subsystem A, and ten systems engineers from Company Y in Houston working on subsystem B. The following paragraphs describe how the meeting today proceeded.
>
> At 10:45AM, the SE lead entered the classified "Collaborative Design" conference room in Boston to begin preparing for her meeting. First, she set up the unclassified conference call phone by entering the "800 number" for external conference calls (since the less expensive local conference call number can only be used for internal company conference calls) and the "host" code. Once the phone conference was set up, she logged-on to the classified conference room computer, only to find that the classified network was down. She logged off and re-started the computer, meanwhile checking all of the hardware connections. Since the meeting material is classified, the only way to share the classified information is by using a

collaborative tool that allows both locations to view the classified material simultaneously over an encrypted network. Since rebooting the computer did not fix the problem, she ended the conference call she just set up, and placed a call to the information technology (IT) help desk in Los Angeles, CA. Since it is only 7:55AM in California, and the help-desk does not open until 8AM Pacific Standard Time (PST), she was out of luck for at least another five minutes.

At 11:00AM, her Boston team members began to show up, as she was on hold with the IT help-desk (Of course, when the help-desk opens at 8AM, it is inundated with a back-up of calls.). Her team members chatted aimlessly about the Red Sox while waiting for the meeting to get started. Since the meeting conversation was about the interface between subsystem A and subsystem B, the meeting could not begin until the Texas team was present. At 11:30AM, after the help-desk in California called the local IT help desk in Massachusetts and had them reset the local classified network, the classified conference room computer was ready, and the conference call with Texas was re-dialed. However, there was no one at the other end of the conference call in TX. The SE lead in Massachusetts again hung-up the conference call, and called the subsystem B SE lead in Texas directly. As the subsystem B SE lead pointed out, the SE lead did not specify in her email 11AM *Eastern Standard Time* (EST), and the Texas team thought the meeting was to begin at 11AM *Central Standard Time* (CST).

Finally, at noon, EST all team members are present at their respective locations and the meeting began. Until now, both subsystems have been undergoing requirements definition and preliminary design in isolation at their respective locations, each team with a difference expectation of the mutual interface. Recently questions and issues have arisen about the nature, format and content of both the hardware and software interfaces between subsystems A and B. Therefore, this meeting was called by the SE lead to discuss the necessary mutual interface specifications between subsystems A and B. The interfaces must be decided and agreed upon by both design teams before requirements definition and preliminary design work on each subsystem can proceed. Once the interfaces are agreed upon, changes may need to be made to either subsystem A or B to accommodate the mutual interface.

The meeting began with a discussion of critical parameter "ABC." A subsystem A team member said that he was expecting "ABC" to be sent in a processed form to subsystem A, from subsystem B over the software interface every "n" seconds. A subsystem B team member replied that this request was impossible for subsystem B. A thirty minute debate ensued before the SE Lead, after having listened to all the perspectives and having noticed a disconnect, asked the Texas team to define parameter "ABC." As it turned out, Company X and Company Y had different acronym interpretations for the term "ABC", which slightly altered its definition. Ten minutes later, the two teams had an agreed upon definition of ABC (which neither team wrote down), and had begun to discuss the actual requirement, specifically the classified frequency "n" with which "ABC" was sent.

Since the teleconference was being held on an unclassified phone line, the two teams could not discuss "n" outright; they instead turned to "cryptic" messages to discuss an agreeable value for "n". A team member from subsystem B said, "If you go to slide 7 of the SE lead's presentation on the classified network, see that number in the middle of the page? Take that number, multiply by 35 and add 7.2. We can get 'ABC' to subsystem A with 'that number' frequency. Will that work?" The team members from subsystem A chose what they thought was the middle number from the list of numbers on slide 7, and performed the calculations for "n". They replied that "No, that number won't work." Another discussion broke out over whether they had the right number from slide 7, whether both teams did the calculations correctly, and why that proposed number (that no one really knew for sure) wouldn't work. After several minutes of arguing, the SE Lead tabled the discussion for the next meeting, and said she would send a classified email to both teams after the meeting with the subsystem A team's interpretation of "n."

With only 10 minutes of the meeting remaining, in an effort to understand what type of data subsystem B needs from subsystem A, the subsystem A team lead asked the subsystem B team to explain how subsystem B plans to process the data it receives from subsystem A. The subsystem B lead replied that the data processing design of subsystem B is Company Y proprietary, and that he could not share that information. A subsystem A team member got angry and reacted by saying, "How are we to design the subsystem A interface data types if we don't know what subsystem B is going to do with the data?" Another argument ensued.

At 1:05PM EST, someone pounded on the classified "Collaborative Design" conference room door - another meeting was scheduled for the conference room. The CDSE meeting ended with both teams unsatisfied and no progress toward an interface specification made, other than the verbal definition of parameter "ABC" which both teams had been working with and thought they knew for months, and neither team wrote down to share with their colleagues.

The hypothetical meeting example demonstrates several of the key issues plaguing the emerging practice of CDSE, such as, management and coordination, collaboration tools and environments, technical SE and design, social and cultural interpretations and misunderstandings, collaborative decision making, information technology, and knowledge management. Due to IT issues, mismanagement, and miscommunications, what was supposed to be a two hour meeting to reach consensus on design interfaces, was instead only an hour-long argument, riddled with technical, tool, and social CDSE issues. Although hypothetical, all of the individual scenarios described in the meeting have actually occurred.

1.2 Motivation

Currently, we lack established methods or practices to carry out CDSE successfully. As SE is one of the "up front" program activities, it affects all of the later product development processes. Any experienced systems engineer or program manager can tell you that if the SE activities are not performed correctly, it can lead to major overall program cost overruns and/or schedule slips down the road. Therefore, successful implementation of the SE activities is critical. With that in mind, there are three additional motivating reasons for performing research to establish successful CDSE practices. First, today's dynamic global competitive environment necessitates distributed SE collaboration. Second, increasingly limited resources, including the availability of experienced systems engineers and tighter aerospace and defense budgets, leads to the need for remote collaboration. Third, there are additional likely benefits of performing CDSE. The following subsections will describe the three motivations for CDSE in more detail.

1.2.1 Motivation 1: Dynamic Global Competitive Environment and System Complexity Necessitates Collaboration

As a result of the dynamic global environment and new integrated enterprises, SE and design practices are currently being carried out by large United States aerospace and defense companies collaboratively over several geographic regions through the use of information technology systems and virtual tools. Fagerson and Olson interviewed several companies regarding their motivation for increased cooperation and integration.[18] When questioned about their motivation, the interviewed companies replied that customer demands for shorter lead times, implementation of new technology, better overall solutions and interfaces, better quality, lower costs, exchange of knowledge, and optimized value chains were their reasons. Today's aerospace and defense products are very complex, and the collaborating companies required to build these complex products are located throughout the United States. Therefore the need for distributed collaboration exists, and companies are collaborating, both internally and externally, to meet those needs. However, the methods for successful collaboration are in their earliest form. Preliminary research has demonstrated that the design process of distributed design teams differs from that of traditional face-to-face groups. However, current research does not identify the critical methods and factors (both social and technical) that enable teams to successfully handle the complexity introduced by the distributed team structure.[23]

As evidenced by the hypothetical CDSE meeting described in Section 1.1, performing CDSE has significantly complicated the already complicated SE process. However, companies are performing CDSE as a necessity to remain competitive in today's dynamic global environment. Therefore, these companies must learn the critical social and technical methods and practices to successfully perform CDSE, while remaining responsive to the customer and within cost and schedule constraints. This research defines successful CDSE practices used by industry and identifies the lessons learned by CDSE participants to assist aerospace and defense contractors to perform CDSE successfully. The topic of CDSE is important to all aerospace and defense contractors, as at some point in the past, or at some point in the future, they are likely to work as partners with their competitors.

1.2.2 Motivation 2: Increasingly Limited Resources

Good, experienced systems engineers are in short supply and high demand in the global aerospace and defense industry, although research shows this situation to be improving with the development of SE educational programs in the United States.[1] [4] [11] [40] [5] It is widely known that the aerospace and defense industry lacks experienced, senior systems engineers and that development of such an engineer is difficult.[12] In the past, when a large aerospace or defense contract was awarded, key SE personnel would be temporarily relocated to the geographic site where the contract was awarded. This temporary relocation would generally result in the company paying to fly the engineer to the site every Monday, paying for hotel accommodations for the engineer for four nights, paying for all the engineer's meals for five days, and then paying to fly the engineer back to his/her home town on Friday evening. This was not only a difficult arrangement for the engineer and his/her family, but was incredibly expensive for the company.

With such fierce competition in the United States aerospace and defense industry and shrinking defense budgets, the aerospace and defense contractors are trying to deliver more products for less cost to satisfy the customer and maintain a competitive edge. This move toward improved productivity and less waste is evidenced by the lean and six sigma initiatives that have sprung up in most aerospace and defense companies throughout the 1990's.[31] It is very difficult in this environment for a company to spend so much money on the wasteful travel of systems engineers within a company.

If successful CDSE practices are identified and can be successfully performed, the expensive and inconvenient temporary transfer of systems engineers to other locations can be halted. Engineers could remain in their current geographic locations and senior, experienced systems engineers can be utilized across a company or multiple companies, all through the use of effective management and collaboration resources.

1.2.3 Motivation 3: Additional Benefits

Not only do we need to collaborate for competitive advantage and to distribute limited resources, we can likely also reap additional benefits that are not possible without CDSE. Possible benefits include: a longer work day (utilizing multiple time zones); cheaper labor (by employing SE's in geographical areas with a lower cost of living); shared expertise and knowledge (within and across companies); and shared risks (within and across companies). These benefits and other possible benefits of CDSE are further discussed in Section 2.3.

1.3 Research Objectives and Questions

There are three main objectives of this research. First, to define successful social and technical CDSE practices by examining how large United States aerospace and defense companies are performing CDSE, and the lessons they have learned in the process. Successful CDSE involves many factors, all of which are addressed by this research in an attempt to understand all facets necessary to implement successful CDSE. These factors are:

1. Use IT and Collaboration Tools
2. Schedule and Conduct Meetings
3. Communicate
4. Train Engineers
5. Overcome Social and Cultural Differences
6. Make Decisions
7. Adapt the Product
8. Overcome Issues and Barriers
9. Determine and Measure the CDSE Benefits

10. Manage Knowledge and Data

11. Coordinate Processes

The second objective is to identify key CDSE issues encountered, barriers and how they were overcome, and practices that were tried and failed in an effort to assist companies who are starting to perform CDSE and prevent them from repeating the same mistakes.

The last objective, as this research is one of the first of its kind on the topic of CDSE, is to identify key topics for future research. This research is exploratory and aims to use the current state of United States aerospace and defense company CDSE practices to identify shortfalls, problem areas, or themes where additional work is necessary or could prove beneficial to CDSE practices.

The three CDSE research questions are summarized in Figure 1-1.

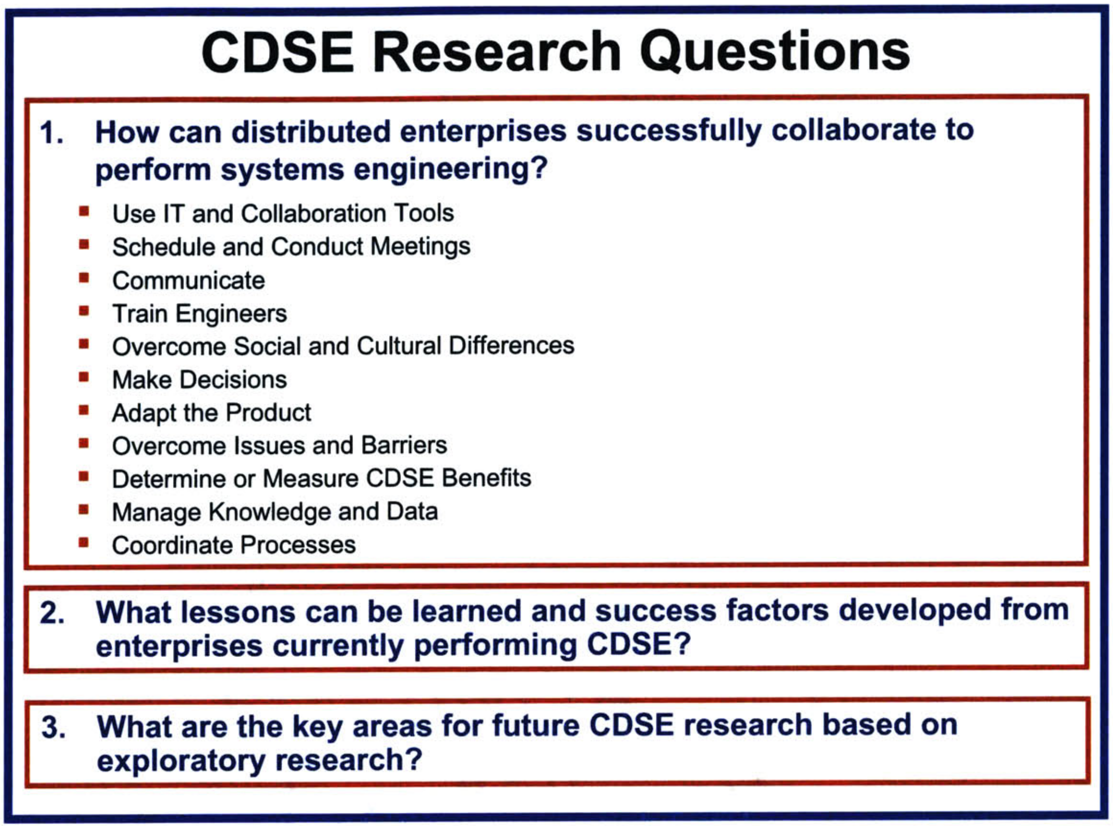


Figure 1-1: Summary of the CDSE research questions addressed by this thesis.

Figure 1-2 summarizes the technical and social factors associated with CDSE.

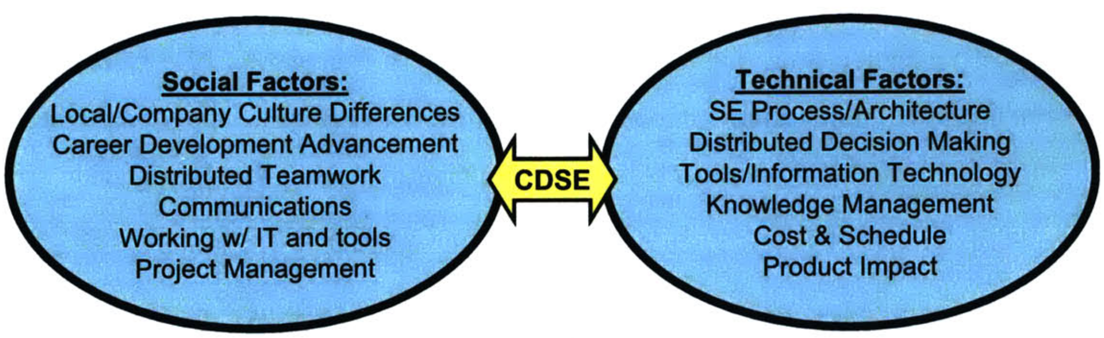


Figure 1-2: Summary of the technical and social factors affecting CDSE.

1.4 Research Overview

To accomplish the objectives outlined in Section 1.3, in-depth case studies were carried out at two United States aerospace and defense companies. The case studies examined many different factors that pertain to the companies' current CDSE efforts through the use of semi-structured interviews with SE personnel, including systems engineers, SE management, and SE support staff. Chapter 3 describes the research methods that were employed in these case case studies in detail. Chapter 4 and Chapter 5 document the findings from each of the two case studies. Chapter 6 details the combined analysis of the two case studies.

1.5 Thesis Overview

This thesis is organized into the following chapters:

1. **Introduction and Overview:** This chapter introduces the problem of CDSE and provides the motivation and objectives of this research.

2. **CDSE - Definitions, Background, and Influencing Factors:** This chapter presents several of the key terms necessary to understand the CDSE discussions and background material. The chapter includes a review of relevant literature and a discussion about the emergence of CDSE, CDSE motivations and proposed benefits, proposed frameworks, previous related works, CDSE influencing factors (including knowledge management, decision making, social and cultural effects, technological impacts, and communication methods), information technology, collaboration tools, and proposed CDSE success criteria and barriers.

3. **Research Methods:** This chapter describes the research methods used to obtain company data, construct interview questions, execute interviews, record data, and perform data analysis. In addition, this chapter summarizes the limitations on the findings discussed in this thesis.

4. **Case Study 1 - Company A:** Included in this chapter is a Company A and

Program A overview; a description of the current SE collaboration situation; a summary of the tools currently in use by Company A; an overview of Company A's knowledge, data, and decision making practices; a summary of the current collaborative SE practices and processes; the social and cultural experiences of interviewees; and a summary of the CDSE benefits and motivations as related by the Company A interviewees.

5. **Case Study 2 - Company B:** This chapter includes a Company B and Program B overview; a description of the current SE collaboration situation; a summary of the tools currently in use by Company B; an overview of Company B's knowledge, data, and decision making practices; a summary of the current collaborative SE practices and processes; the social and cultural experiences of interviewees; and a summary of the CDSE benefits and motivations as related by the Company B interviewees.

6. **CDSE Case Study Analysis:** This chapter provides a comparative analysis of the two case studies and summarizes the major issues and success factors for each of the major topics investigated in the case studies.

7. **CDSE Research - Recommended and Proposed Future Work:** This chapter summarizes many of the obvious and emergent areas of future work on the topic of CDSE, including research aimed at addressing the limitations of the case studies described herein and the recommended future research topics requested by interviewees.

8. **Summary and Conclusions:** This chapter provides a brief summary of the research presented in this thesis and summarizes the major findings and conclusions.

Chapter 2

CDSE: Definitions, Background, and Influencing Factors

The topic of SE alone, with the many different disciplines, perspectives and processes that comprise it, has a plethora of literature available to research. Combined with the topics of collaboration and distributed teamwork, the research is overwhelming: confusing acronyms, ill-defined terms, and a large number of influencing factors. This chapter therefore begins by presenting several of the key terms necessary to understand the CDSE discussions and background material presented in this chapter and throughout this research. The chapter goes on to include a review of relevant literature and discussions on the emergence of CDSE, CDSE motivations and proposed benefits, proposed frameworks, previous related works, CDSE influencing factors (including knowledge management, decision making, social and cultural effects, technological impacts, and communication methods), information technology, collaboration tools, and proposed CDSE success criteria and barriers.

Reader beware, the influencing factors and topics discussed within this chapter are not comprehensive of all the possible literature available on CDSE and all of its related elements. Doing so would be impossible, as CDSE is affected by many, many areas of research.

The content included in this chapter touches upon many of those CDSE-related fields and sets the stage for the information presented in this research. Figure 2-1 graphically depicts the relationships between many of the CDSE influencing factors.

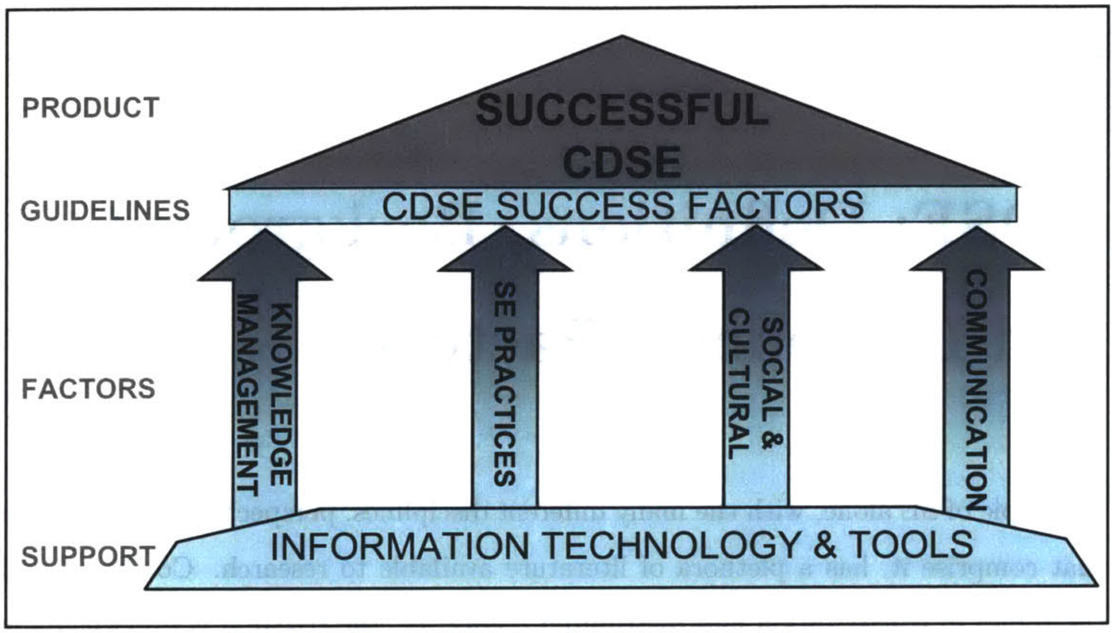


Figure 2-1: Summary of CDSE influencing factors and relationships.

2.1 Definitions: Laying the CDSE Groundwork

Unfortunately, many of the terms used in CDSE have multiple meanings and applications, depending on the field. The following terms appear frequently in this document and the CDSE literature. For each term, a brief description and definition is provided:

Collaborative/Cooperative Teams Most simply put, these are individuals or groups of individuals who are working together toward the same common goal, whether that goal be creating a final product, creating a tool, performing analysis, etc.[28] The collaborating or cooperating individuals or teams may be part of the same organization, or from many different organizations. They may be co-located or located in many different locations. Often, collaborations may be legal arrangements or contracted arrangements, in which case there is a defined team structure and team leadership. In any collaboration or cooperation, there is a great deal of coordination activities, as Harvey and Koubek explain: "There are many steps that lead to any collaboration between multiple organizations, such as negotiating and developing the collaboration agreement, administering the collaboration project, and actually forming the group to complete the collaborative task."[23] Note that throughout this research and in the literature, the terms collaborative and cooperative are used interchangeably.

Distributed/Dispersed Teams Distributed or dispersed teams are composed of people in the same organization or on the same project who work in different geographic locations. The distributed team can be comprised of people who work for the same company and are geographically separated and/or people who work from different geographic locations AND different companies.[28] Often, individuals that work on distributed teams also work on collaborative teams. Keep in mind, a 1977 study found that fellow collaborators more than 50 feet away are not "co-located," and thus even with co-workers in the same building, interactions take on those similar to a distributed team.[28] Note that throughout this research and in the literature, the terms distributed and dispersed are used interchangeably.

Systems Engineering (SE) There are many theories and definitions available for the practice and discipline of SE. There are unfortunately many disputes, even amongst experienced systems engineers, as to the correct definition of the activities involved in SE. The following following definitions are some of the most common:

From MILSTD 499A SE is " A logical sequence of activities and decisions that transforms operational need into a description of system performance parameters and a preferred system configuration."[13]

From the International Council on Systems Engineering (INCOSE) "Systems

Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem: Operations, Cost & Schedule, Performance, Training & Support, Test, Disposal, Manufacturing. SE integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs."[25]

Harris proposes that "Systems Engineering is a socio-technical activity it involves people using technology. In this respect both the SE process and the systems it creates are similar." [21] Harris further describes that "systems engineering can be viewed as occurring in three, broad phases: 1. Concept generation, arriving at a statement of high level User Requirements. 2. System design, involving the establishment of top level System Requirements. 3. Detailed requirement breakdown, supporting design and creation of system elements and their interfacing."[22]

Shenhar reviewed the SE literature and summarized the following common SE activities:

- "The identification of an operational need with an opportunity to create a system to respond to this need."[36]
- "Setting the exact system and functional requirements to ensure the best fit to customer needs."[36]
- "Dividing and allocating the functional requirements into different subfunctions and modes of operation."[36]
- "Choosing a system concept that will best fit these requirements."[36]

- "Designing the system architecture, based on the chosen concept."[36]
- "Dividing the system into separate subsystems and components to ensure overall optimization, least interfaces, and least mutual effects of the various subsystems."[36]
- "Optimizing the specifications of the various subsystems through simulation, analysis, and trade studies."[36]
- "Managing the interaction with various engineering groups who perform the subsystems design while integrating various people and disciplines."[36]
- "Performing the integration of the various subsystems into a total system." [36]
- "Evaluating the performance and qualifications of the final system through simulation and testing activities."[36]
- "Demonstrating the operating system to customers and convincing them that it responds to their needs."[36]

These descriptions are all very different yet very similar, and are all correct, depending on the context and application. **Note that for the research described herein, the reference to "systems engineering" or "systems engineering activities" refers most closely to the summary provided by Shenhar, and is inclusive of system integration and qualification activities.**

Systems Engineer Simply put, a systems engineer practices SE. Given that SE is a very broad discipline, what exactly do systems engineers do? Many have heard the expression "Jack of all trades, master of none." In many ways, that idiom describes a systems engineer well. Systems engineers tend to have a broad range of knowledge in many different disciplines and applications, but rarely have very specific knowledge in any one area. Experienced systems engineers, however, may be a "jack of all trades, master of many." It is important to note that the older population of systems engineers usually were "discipline" engineers for a long time, before moving to the "newer" discipline of SE. Shenhar summarizes the many roles of a systems engineer:

- "Need identifer and system marketeer: To be the link bond between customer needs and system idea."[36]
- "Architect and chief conceptual designer: To be the lead person in envisioning the systems concept, and to create the link between the customers requirements, the systems requirements and the systems configuration."[36]
- "Integrator: To see the entire picture and how each part is contributing to the performance and feasibility of the system as a whole. Further, the systems engineer must coordinate the efforts of the various disciplines and professions involved such that the result is an overall optimal system."[36]
- "Analyst and data processor: To collect data from various sources and analyze them as a basis for decision making."[36]
- "Problem solver and decision maker: The process of systems engineering involves numerous trade-off decisions and the resolution of conflicts at different interface points. These conflicts are primarily professional rather than personal, and reflect the different views, interests, and biases of the many participants in creating the system."[36]
- "Manager and administrator: In addition to being a technical leader, the systems engineer must be a manager and administrator. He or she must work with people, organize their work, motivate them, communicate with them, and deal with their needs."[36]

Due to the complex nature of the products in development in today's United States aerospace and defense industry, systems engineers are a vital component of the product development team.

Traditional SE Environment Intra-company and inter-company collaborations to develop complex aerospace and defense products have been the norm in the United States aerospace and defense industry for a long time. However, in the past, a great deal of money, travel, and time went into co-locating development teams within the same place. Harris summarizes, "The opinion widely held in the systems engineering community is that the best way to achieve close teamwork is to co-locate the team members. If they all work together in an environment where social contact is possible and easy access leads to easy clarification of issues, the result is a high level of collaboration."[22] Today, however, technology has enabled collaborations of all types, including SE, to be carried out via information technology.

Collaborative Distributed Systems Engineering (CDSE) Combining the definitions of the previous terms, CDSE is collaborations between individuals or teams from within the same company or from different companies, performing SE activities from geographically distributed locations. The CDSE teams have a common goal, which is to develop a final product that meets or exceeds the customers' expectations.

Virtual Team CDSE teams are a type of virtual team. Lipnack and Stamps define a virtual team broadly as "a group of people who work interdependently with a shared purpose across space, time and organizational boundaries using technology."[28]

Computer Supported Collaborative Work (CSCW) To support virtual teams a new field has emerged to provide these teams with the tools they need to do their job. Achalakul, Sirinaovakul, and Nuttaworakul summarize CSCW as "the concept that can be used to create a computerbased, distributed, virtual workplace, where researchers/analysts can meet and interact with one another via the virtual agents or objects. This concept focuses on putting interactive, dynamic representations of data and people into virtual landscapes and offers the powerful mechanisms for navigation, exploration and communication."[2] This field is important to CDSE, as the concepts developed and tested for collaborative tools are key to supporting the distributed development of complex systems.

Successful This term is used often in this research, as it is necessary to distinguish between CDSE and "successful" CDSE. Many different forms of CDSE take place throughout the aerospace and defense industry. However, if CDSE is not carried out "successfully," the products and services delivered are not successful - they may not meet customers' expectations, they may be over-budget, or they may be delivered late. Realistically, as CDSE practices were improving and lessons were being learned about this new way of doing business, the CDSE teams may have been over-budget or behind schedule.

Ultimately, success in this context means satisfying or exceeding the customer's expectations and being as close to "on schedule" and "within budget" as possible. "Successful" additionally means not wasting resources, such as personnel, time, or money, by performing work in a less optimal manner, when better alternatives exist. This definition of success is vague, and has been intentionally defined so to be applicable to all ranges of aerospace and defense product or service development.

2.2 How has Distributed Collaboration Emerged?

Armed with the necessary descriptions and definitions for CDSE, it is also important to understand how distributed collaboration emerged. In general, the emergence of CDSE has happened slowly over time and mirrors the improvements in information technology over the last two decades. As information technology has improved, it has enabled engineers to take advantage of the capability to communicate using near-real-time networks, tools, and applications. Also, as information technology has improved, so has processing and computing power, enabling complex analysis and simulations an take place.[17] Harvey and Koubek summarize that "with the emergence of information technology and the convergence of computer networking and telecommunication technologies, it is no longer a requirement for people or cooperative companies to be located in the same place to communicate. Instead, people or companies that are geographically dispersed can engage in collaborative arrangements communication networks, increased processing and computing powers.[23]

Not only has distributed collaboration emerged because technology has enabled it to do so, the nature of business has changed over the past few decades, and companies have adapted to remain competitive. Madni, Lin, and Madni explain, "monolithic enterprises that totally own all of the products, services, and channels required to serve a customer are rapidly being replaced by strategic partnerships, virtual enterprises, and integrated value chains. The need to operate in a rapidly changing business and technical environment is driving the need for technology infrastructures and application architectures that are increasingly more flexible, interoperable, extensible, and maintainable."[29]

Through improvements in technology and the advent of such rapid communication and transportation media, companies now have global reach. With such a vast amount of potential customers, sources of raw materials, and markets to sell goods, it is becoming cost prohibitive to co-locate all of the key-stakeholders in the same location. Therefore methods of distributed collaboration have emerged out of necessity. Harris describes, "As collaborative teams are increasingly built from organizations with global operations, or built from groups of globally dispersed, collaborating organizations, the costs of bringing teams together in one place for longterm projects becomes prohibitive, to say nothing of the social disruption involved."[22]

These factors - improvements in information technology, transformations in company organizations, and the cost-prohibitive nature of co-located global teams - have allowed collaborative distribution to emerge into what it is today. Now that distributed collaboration has emerged, there are several motivations for utilizing collaborative distributed teams and several proposed benefits to their utilization.

2.3 CDSE Motivations and Proposed Benefits

Despite the complexities that arise when teams are teams are collaborating through IT from distributed locations, there are many motivations and possible benefits that outweigh the potential pitfalls. SE is already highly complex, why complicate SE with distributed collaboration?

2.3.1 Why Collaborate?

Throughout the literature on distributed collaboration, many motivations were stated for distributed collaboration during the design process, a typical SE activity. The following quotations from the literature summarize those motivations:

- **Flexibility:** Hammand, Harvey and Koubek believe that collaborative, distributed engineering will "increase the flexibility of design and production processes by pooling strengths with multiple organizations on a product-by-product basis to create distributed collaborative corporations."[20]

- **Improvements to the Entire Design Process:** Fagerstrom and Olsson interviewed many suppliers in the mechanical industry to determine their reasons for collaborating, they report: "The driving forces behind increased cooperation and integration between main and subsuppliers are, according to the interviewed companies, the following: demands for shorter lead times, implementation of new technology, better overall solutions and interfaces, better quality, lower costs, exchange of knowledge, and an optimized value chain."[18]

- **Improved System Design from Utilization of Global Expertise:** Harris explains, "The purpose of pulling together a team involving a range of contributing organizations is to use the specific domain expertise of these organizations to contribute to the system design. One of the benefits of dispersed teams is that the best sources of expertise available in the world can be integrated into the team."[22]

- **Pooling of Resources:** Harvey and Koubek state that, "Collaborative arrangements allow companies to share resources and core competencies while also sharing the associated risk and infrastructure costs."[23]

- **Expand Markets:** Harvey and Koubek further explain that, "Collaborative organizations may allow a company to exploit the window of opportunity where they may not be able to respond alone."[23]

With the motivations established for why companies may partake in distributed collaboration, it is important to understand why it is necessary to perform research on CDSE.

2.3.2 Why study CDSE?

The literature available concerning distributed collaborative practices and collaboration tools provided many reasons why additional research on CDSE is profitable. First and foremost, if we do not study CDSE, we risk the possibility for failure. Lipnack and Stamps claim,"Everything that goes wrong with in-the-same-place teams also plagues virtual teams - only worse."[28] Lipnack and Stamps go on to say, "That the one major reason that virtual teams fail, when compared with face-to-face teams, is because they do not change their working environment and processes to accommodate for distributed collaboration."[28] If that isn't enough motivation for the study of CDSE, the following paragraphs describe several additional motivations:

- **We Currently Lack Defined Successful CDSE Practices:** The preliminary literary findings by Hammond, Koubek and Harvey suggests that collaborative, distributed design practices differ from those of traditional face-to-face methods in a systematic fashion. The literature reviewed by the authors reveals that current research does not yet identify both the social and technological factors that will enable teams to perform successfully in a collaborative distributed environment. As the authors summarize, "Once the elements are clearly identified, (a) designers can intervene with appropriate technological support, and (b) management can intervene with appropriate training, protocols, and methods to facilitate virtual work group success." [20] This research is a first-step in achieving the mission outlined by Hammond, Koubek, and Harvey, as it documents the current state of CDSE environments and identifies many key factors and necessary areas for future research in collaborative, distributed design.

- **Future Application for Systems of Systems (SoS):** SoS is a popular buzzword in today's aerospace and defense industries. The definition of a SoS is widely debated, but Maier summarizes that an SoS is "an emergent class of systems that are built from components which are large-scale systems in their own right."[30] This research does not directly define, discuss, or describe SoS. For a detailed discussion of SoS in general, the reader is referred to popular SoS works by Maier (1998) [30] and Sage and Cuppan (2001).[35] There are many applications and examples of SoS, especially in the aerospace and defense sectors. As budgets are shrinking and resources scarce, systems currently in use are being adapted and utilized to take on responsibilities and functionalities they were not designed to do. Therefore SoS, and the SE activities related to SoS's, are an important motivation for CDSE and will likely be a very important topic of future CDSE research. The nature of SoS requires that SE teams work in a collaborative and distributed fashion to achieve SoS project success. As Chen and Clothier explain, there are common environmental, infrastructure and management challenges for SoS design and management, such as: high technological complexity; multiple stakeholder inputs; many constituent systems with independent lifecycles and differing lines of responsibility; system formulation on short notice (relatively speaking) to meet unprecedented needs; engineering data and knowledge sharing across companies/distance; enhanced SE processes and process management; new SE organizational structures; and SE tool suites and collaborative work environments for distributed engineering. These SoS challenges may be facilitated by successful CDSE methods.

- **Be Armed in Advance:** By studying CDSE and all the factors and issues associated with it, we can better understand what can go wrong, and what we can do to make things go more smoothly. Komi-Sirvio and Tihinen studied issues associated with distributed software development. They agree, "Being more aware of possible pitfalls and potential risks, those involved with the development of distributed software projects should therefore be in a better position to successfully plan and execute these projects."[26]

- **CDSE is Customer-Driven and Exemplified:** One of the United States aerospace and defense industries biggest customers, the United States Navy, has already taken great strides in improving collaborative design and SE work environments. The Technical Cooperation Program (TTCP) is an international organization that promotes defence technical information exchange. The responsibility of this organization is to "provide battle force systems that are not just mobile, responsive and effective, but which are also integrated and interoperable when employed as a joint coalition force." A panel of the TTCP called the The Technical Panel on SE for Defense Modernization (TP-4) is tasked with investigating common SE practices and collaborative environments for a state-of-the-art battle force. After studying the current (2002) state of SE common practices, the TP-4 panel concluded that: "Basic collaborative capabilities and an adequate base of higher-level knowledge and information to resolve force level interoperability and integration problems do not presently exist to support the current acquisition process." With these shortcomings in mind, the United States Navy has developed a Collaborative Engineering Environment (CEE) comprised of modern information technology and computer-aided engineering tools for use by the United States Navy acquisition community.[9]

The Naval CEE consists of three major elements: a Decision Support Environment (DSE), to support interaction and the sharing of information; the Integrated Engineering Environment (IEE), which is comprised of government and commercial systems architecting and engineering tools; and the Interoperability Data Management and Analysis (IDMA), which is the link to systems databases and visualization tools. This environment allows the Navy to share a wealth of information to distributed development teams via a common environment. The Navy has provided networking, training, standardized terms and interfaces, as well as classified capabilities to support the distributed SE activities. The customer has recognized the importance of a distributed, collaborative SE environment as vital to their mission success, acquisition process, and long-term capabilities. It is therefore necessary that Navy contractors also invest their interests and resources into the same, and follow the example the customer has set-forth.[9]

- **Use CDSE for Competitive Advantage:** In today's United States aerospace and defense industry, your fellow collaborator one day is your competitor the next. Many aerospace and defense companies have very similar, if not the same, core competencies, although they may take different approaches to developing a product. Being able to successfully collaborate and coordinate to create an integrated final product across geographical boundaries can be considered a competitive advantage.[23]

2.3.3 Proposed Benefits of Distributed Collaboration

In addition to the many motivations described in the previous section, another motivation is the proposed benefits that can be realized when performing distributed collaboration. The following is a list of assembled possible benefits found throughout the review of CDSE-related literature:

- CDSE enables system development to take place independently of the geographic location of contributing individuals or companies.[26]
- CDSE enables the possibility to utilize teams in different time zones to support "round the clock" high-speed development.[26]
- CDSE enables the ability to employ a better skilled/more experienced staff.[26]

- CDSE enables the opportunity to lower development costs (traveling less, either by being closer to raw materials, working in areas with a lower cost of labor, utilizing more experienced engineers, etc.)[26]

- CDSE enables the ability to respond to the customers' needs by working with them locally (A single team member or small group can be sent "on-location" to work with customer while the rest of the team remains at home-site).[26]

- CDSE enables the capability to remedy the situation of unbalanced demand for SE in different geographic locations.[26]

- Systems engineers, especially those from different intellectual and occupational backgrounds, have diverse experiences and knowledge. Therefore, CDSE activities and final products will be more effective, since distributed systems engineers can leverage their different perspectives and access to different information sources.[37]

- CDSE enables "Organizational Flexibility" and the "Increased ability to absorb change."[32]

- "The steps that teams take to cope with their network nature - using collaborative technologies and designing flexible organizations - not only compensates for capabilities lost, but also establish the basis for extraordinary performance." [28]

- CDSE enables shorter cycle times by utilizing parallel processing and improved communications.[28]

- CDSE can "[l]everage organizational learning and the sharing of best practices across collaboration sites."[28]

2.4 What distinguishes SE from SW/HW/Virtual enterprises?

At this point, the reader may be wondering why all of the research available on virtual teams and collaborative software development, etc. is not enough to address CDSE. What sets CDSE apart from just "CD" teaming is the nature of systems engineering. Unlike most "traditional" virtual teams, or collaborative distributed software or hardware development teams, SE begins with no definition. In software or hardware engineering (say for example, mechanical, industrial, civil, radar, aerospace, etc.), the designers or engineers begin with a set of requirements. The engineers set about to design a system that meets the requirements given to them. At the end of their hardware or software design process, the engineers hand-off (to systems engineers) a finished product and typically a test document describing how each requirement was met. In that type (harware or software development) of collaborative environment, it is easier to "divide and conquer" a product - requirements can be separated, smaller subsystems can be made and distributed amongst teams.

On the contrary, SE begins with an idea - something the customer has in mind. (And often times, the customer doesn't always know what he/she has in mind). Through creative thought processes, simulation, analysis, architecting, and experience, the systems engineers derive and flow down the requirements for an entire final product: hardware, software, firmware, and interfaces. In the SE process, it is not easy to determine where or how to best "divide and conquer" an abstraction. Buede explains, "What makes SE unique, especially in contrast with traditional engineering disciplines, is that SE does not build tangible products. Whereas civil engineers might design buildings and electrical engineers might design circuits, systems engineers deal with abstract systems, and rely on other engineering disciplines to design and deliver the tangible products that are the realization of those systems."[6]

In addition, as the "up-front" process, SE activities, especially design, impact the entire rest of the product development and manufacturing cycle. Harvey an Koubek claim that, "A design mistake discovered in manufacturing costs approximately 100 times what it would have cost if it had been found during design."[23] SE sets the stage for the entire product: design, development, manufacturing, maintenance, and product retirement.

Although much of the research available about collaboration tools, communications, and social/cultural interactions in a collaborative distributed environment are relevant to CDSE - there are many unexplored areas, specifically when it comes to collaborative system architectures, development processes, and system design methods, that are newly introduced by this research.

2.5 What are the Proposed Frameworks and Models for Distributed Collaboration?

In order to understand what is possible for CDSE, it is important to understand the existing models and frameworks for distributed collaboration. The following paragraphs summarize previous works related to distributed collaborative models and frameworks.

- **Enterprise Ontology to Support Distributed Collaboration:**
  The complex activities associated with distributed collaboration can be well-represented by an ontology. Madni, Lin, and Madni created IDEON, an enterprise ontology to support integrated planning and execution of enterprise processes. The focus of IDEON is integrating and managing enterprise planning and execution within a collaborative distributed environment with applications such as supply chain management, command and control, collaborative SE, emergency management and crisis planning and execution. As defined by the authors, IDEON is "a unified enterprise ontology that provides a common foundation for designing, reinventing, managing, and controlling collaborative,

distributed enterprises....It consists of: (a) a set of 'business' objects that represent common entities within an enterprise context; (b) relations that link these objects in specific ways to establish business configurations; and (c) business rules that govern the behavior of various business configurations." The authors propose that IDEON will have several potential benefits that directly impact CDSE, such as decision support, analysis tool support, process and organization re-engineering support, thereby providing users with requested information, and promoting a common understanding between system develops and users.[29]

IDEON is based on four high-level design concepts: neutrality (notationally independent), extensibility (readily extensible to other application domains), complementarity (many perspectives of enterprise are needed and used), and interoperability (it is designed to interoperate with other ontologies or enterprise processes). IDEON has four views that capture the key relationships and concepts characterizing an enterprise. The four views are reproduced graphically in Figure 2-2. The views are: 1)Enterprise Context View: the "interactions between and enterprise and its external environment;" 2) Enterprise Organizational View: the "structural view of the enterprise which complements the enterprise context view;" 3) Process View: the "(re)planning-execution-control cycle;" and 4)Resource/Product View: "elaborates on the various types of resources that might be needed to execute a process."The authors successfully applied IDEON to two applications, a process-centric crisis action planning and execution system and an integrated product-process development project.[29]

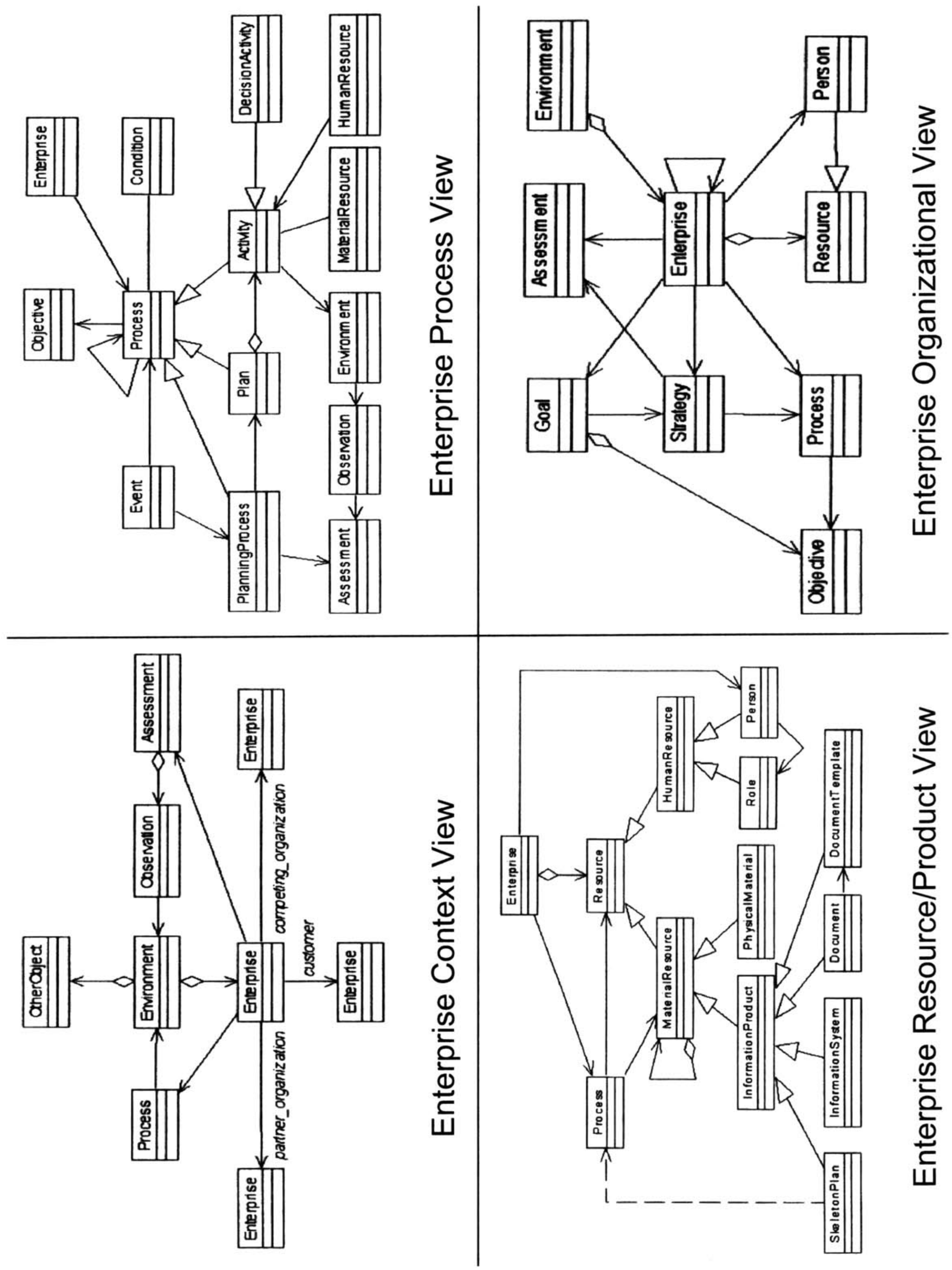


Figure 2-2: Summary of IDEON ontology views summarized from Madni, Lin, and Madni, pp. 41, 42, 43, 44.[29]

- **Systems Engineering Data Representation and Exchange Standardization Project (SEDRES):**
  The SEDRES Project is an important initiative because it helped to bring about world-wide realization of the importance SE and SE standards, especially across companies. The SEDRES project culminated with the creation of an ISO Application Protocol for data exchange between SE tools. Armed with these standards, data and knowledge transfer between collaborative SE tools can be greatly improved.

  The history of the SEDRES Project is very interesting. "The SEDRES Project is sponsored by Europes five major aerospace companies and three universities, and is supported by grants under the European Framework IV ESPRIT Programme, Project No. 20496. Its overall objective is to support the development of engineering systems by teams whose participants may be in different companies and in different countries." The SEDRES project is a European initiative with the mission to standardize data exchange capability between SE tools. As Candy and Harris explain, there are several SE tools commercially available that have the capability to export and import data, allowing for tool data exchange. However, current data exchange methods are very tool-dependent, and there are no standards widely available to expedite and facilitate it. The SEDRES Project team has proposed and evaluated a data model, the concept for which is reproduced in Figure 2-3. The SEDRES Project Missions are:[7]

  "Primary Mission: To provide a first draft Standard for data exchange between systems engineering tools as an enabling technology for an Integrated Project Support Environment, to publicize this standard within the systems engineering community, and to take action to encourage its adoption."[7]

  "Supporting Mission Statements: 1. To improve the systems engineering process. 2. To support Integrated Product Development. 3. To obtain commitment by Systems Engineering tool developers."[7]

Candy and Harris explain that a SE data exchange standard was developed and evaluated by systems engineers in the Project member companies. The evaluation results obtained were compared against the supporting missions, and the SEDRES approach for data exchange has shown promising results for improving SE data exchange between tools. The success of the SEDRES Project has led to the data exchange model being incorporated as an Application Protocol (AP-233) within the ISO 10303 STEP environment.[7],[21]

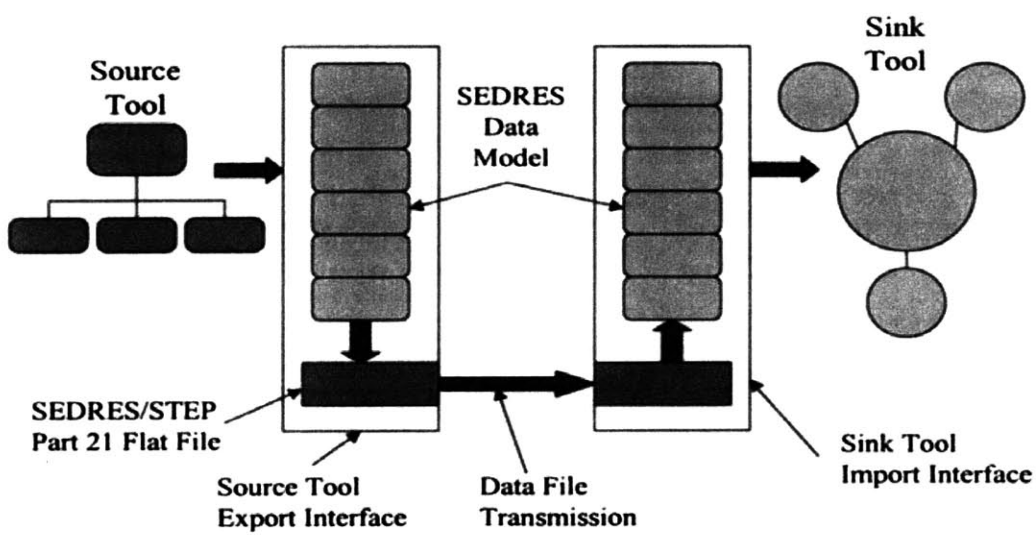


Figure 2-3: Summary of SEDRES data exchange model reproduced from Candy and Harris, p.14.[7]

- **Development of a Virtual Laboratory:**

  Achalakul, Sirinaovakul, and Nuttaworakul, have developed a design framework along with implementation details for a "virtual research laboratory" to support collaborative researchers in dispersed locations. The framework the authors have developed facilitates the dissemination and analysis of information in real-time to multiple stakeholders. The authors claim that their virtual laboratory is: "A solution for building a bridge for accessing, transferring, and manipulating data/objects via the internet." The author's framework is applied to data analysis efforts to tie together different research communities in Europe and Asia. Figure 2-4, reproduced from [2] summarizes the collaborative environment architecture proposed by the authors. The environment allows researchers to exchange text, images, voice, and video and coordinates discussion by means of chat and whiteboard applications. Also in the environment are computing, plotting, spreadsheet, and presentation tools to allow real-time research coordination. To implement their architecture, the authors chose Microsoft component model COM/DCOM. The reader is referred to [2] for a detailed discussion of the environment implementation. Upon completing the implementation of their virtual laboratory, the authors set out to evaluate its quality with a real-time example using ISO 9241's concept of system usability. The example chosen was analysis of a satellite image by researchers in Korea and Thailand. Once the analysis was complete, the authors asked the participants to rank each aspect of the virtual laboratory (effectiveness, efficiency, and satisfactory) on a scale from 1 to 5, where 5 is the highest score. Using these results, the authors scored the system usability at 68%, which they consider satisfactory for the beta version of their laboratory. In summary the framework and implementation scenario developed by the authors enables multiple analysts from dispersed locations to work together via a collaborative environment over the internet as though in the same location.[2]

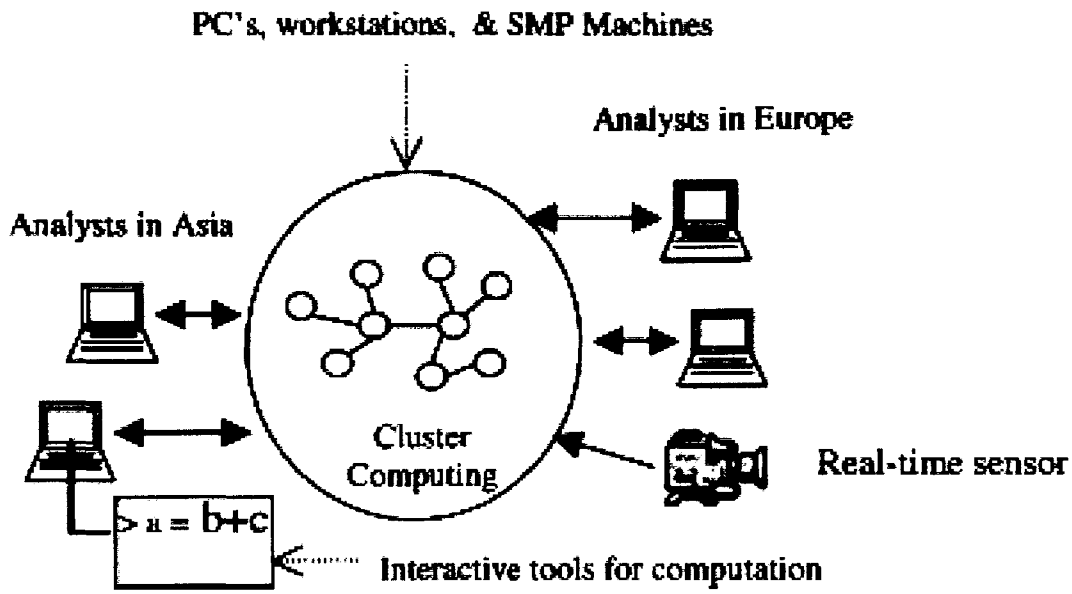


Figure 2-4: Summary of virtual laboratory collaborative environment architecture reproduced from Achalakul, Sirinaovakul, and Nuttaworakul.[2]

- **Framework to Evaluate Collaborative Systems for Distributed Collaboration:**

  The SEDRES project results can greatly improve CDSE tools by improving communications and data exchange and the Virtual Laboratory project serves as the basis for coordinating several applications and interfaces for system research and analysis through a single tool. However, how do we go about evaluating collaborative tools in the first place? Huang developed a framework to evaluate collaborative systems that support distributed collaboration. The framework he derived from reviewing literary works in the fields of information systems, system evaluation, and stakeholder analysis is reproduced in Figure 2-5. Huang's framework consists of five domains: the "context" domain (both internal and external to the collaborative system and organization in which it resides), which is necessary for any evaluation to make meaningful sense; the "content" domain, or what is being evaluated (hardware, software, location, interfaces, etc.); the "stakeholders' perspective" domain, or the views and requirements of system users, customers, developers, or even company shareholders about the collaborative system; the "process" domain, simply put is "how to carry out the evaluation;" and the "success factor" domain, or the lessons learned about the system that become building blocks of success for future collaborative systems.[24] The framework was used to evaluate a multi-site, multi-partner collaborative system. During the trial evaluation, additional areas for future work were identified.

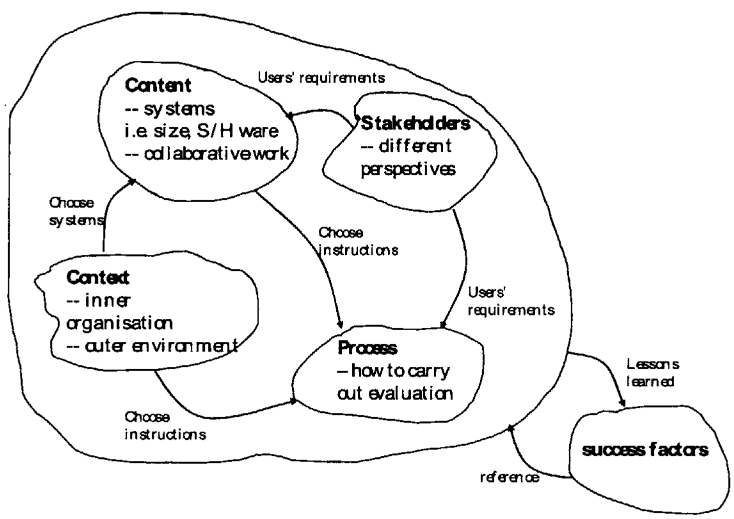


Figure 2-5: Framework for evaluation of collaborative systems reproduced from Huang.[24]

- **Lessons from The Toyota Product Development System (TPDS) Model:** Sometimes tools aren't the focus of collaborative distributed relationships. For years, many leaders in today's manufacturing industries have tried to understand and implement the very successful Toyota Production System and Toyota's principles of lean manufacturing. However, few know of Toyota's equally as successful Product Development System. (Toyota's product development practices achieve approximately four times more value-added productivity than typical American product development systems in the automotive industry.) Although not directly related to distributed collaborative work, there are several interesting and useful insights to be gained by the TPDS methods that may be very helpful in a collaborative distributed environment. Cleveland describes the TPDS as a "knowledge-based" approach, one in which knowledge and technical expertise drive decision making. The key features of the TPDS are summarized in Table 2.1. Also included in Table 2.1 is a "Relation to CDSE" column, discussing why the TPDS model features may be important to CDSE. [8]

Key TPDS Feature	Characteristics	Relation to CDSE
"Functional Managers as Teachers"	- Managers are most technically competent engineers - Teaching and managing by continuously asking why	- Management in CDSE is politically driven, not always best technical engineers
"Clear Emphasis on and Reward for Technical Competence"	- Engineers judged on knowledge and use of technical information - Technical excellence revered	- Often in CDSE environements (and defense in general) awards are given for meeting cost/schedule milestones, as opposed to technical product, and therefore there is misalignment
" 'Pull' Scheduling and Distributive Planning and Control"	- Single master schedule with key dates that are never missed kept by chief engineer - Chied engineer outlines what needs to be done by each team by key dates - No need for a top-down detailed timeline for the program management	- Often in CDSE environements, there are multiple schedules that do not always align. Each sub-team/location is typically responsible for determining what is needed to reach milestone.
"Set-based Concurrent Engineering"	- Sub-system level engineers generate multiple alternative designs/solutions for each design - called "sets" - Each design is evaluated via performance trade-offs - System evolves from combinations of proposed subsystem designs	- Typically in CDSE, design trade-offs and evaluations take place at sub-system level, not at system level.
"Knowledge Capture and Re-use"	- Knowledge and data about "sets" and their respective performance data are stored and easily accessible by all team members - This system greatly reduces need for re-invention - This system encourages re-use	- Knowledge sharing is huge issue for CDSE - Very little re-use and a lot of re-invention are often needed
"Standardization around Checklists and Design Standards"	- There are engineering checklists amd design standards for each sub-system and component	- Standardization of products and processes is key for CDSE
"Visual Management of the Development Process"	- Team rooms have color-coded graphics to make it immediately obvious where a project is not meeting its goals	- A similar system might prove very useful in CDSE collaborative tool environement to relay messages and ensure ALL team members at all locations are aware of critical issues/programmatic problems.

Table 2.1: Summary of TPDS key features adapted from Cleveland.[8]

2.6 CDSE Factors to Consider

As previously mentioned, there are many factors that influence and affect CDSE. This section summarizes several key areas of research from the literature that relate or affect CDSE, including "technical", social, cultural, knowledge management, decision making, communications, and knowledge sharing factors.

2.6.1 Product Factors to Consider in CDSE Environment: Technical, Process, Architectural

A great many people have written a great many things (and not so great things) on the traditional practice of SE and its related activities, such as SE processes, system architecting, interface methods, etc. However, very little research has been done, and can be found in the literature, on SE-specific collaborative distributed work. There have been several studies related to collaborative, distributed manufacturing design and software development. This section summarizes those related studies and in some cases indicates how the findings may pertain to CDSE research.

2.6.1.1 Collaborative Distributed Manufacturing Design

The following two works, based on collaborative, distributed manufacturing design, form the basis for a lot of the methods and factors used to develop the interview questions in this CDSE research. In addition, as this research focuses predominantly on the manufacturing product design process, it also provides a lot of the background material on distributed collaboration practices and collaboration factors and issues that are relevant to the "design" aspect of SE.

Harvey and Koubek reviewed several of the attributes and factors that affect distributed collaboration engineering teams. The goal of their research, similar to the goal of this research, is to use the information collected to guide future research and development of new tools and methods to support distributed collaboration. To understand the attributes and factors affecting distributed collaboration, the authors researched three comprehensive areas of the literature. Upon completion of their literature review, Harvey and Koubek propose a model that they believe can contribute to the understanding of the engineering collaboration process. There model is based on cognitive, social, and environmental attributes that were identified from their literature search. The attributes are grouped into three areas: task characteristics ("complexity and intellectual process phase"), collaborative technology ("communication medium and conversational props"), and group/individual development ("group vocabulary schema, individual cognitive resources, and group task cohesion").[23] The model developed by Harvey and Koubek is reproduced in Figure 2-6. The authors explain that the category of task characteristics: "deals not only with the complexity of the task itself but begins to ask the question as to how the phase of the engineering task may drive the requirement for different tools to support the communication process." The category of collaborative technology includes the elements that affect the communication between distributed engineers, including, conversational props and communication media. The category of group/individual interaction considers both the social and cognitive interactions that affect the collaborative process. This category includes such factors as cognitive workload, vocabulary, and task cohesion.

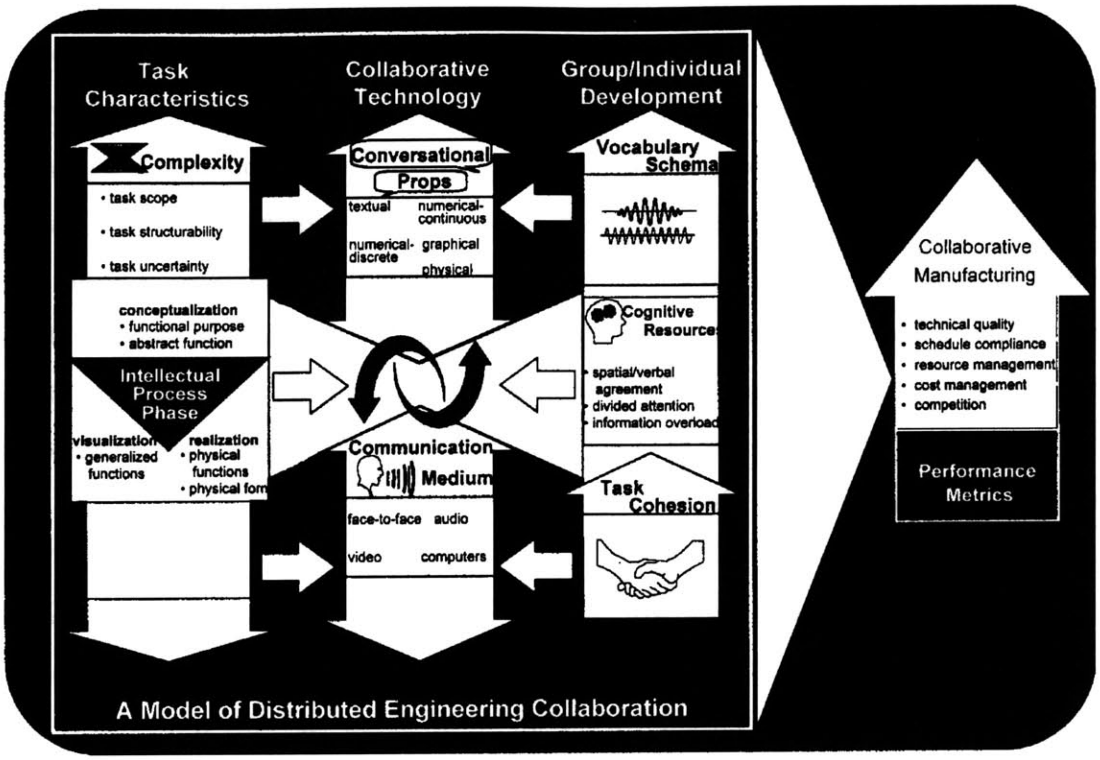


Figure 2-6: Model of distributed engineering collaboration reproduced from Harvey and Koubek.[23]

To substantiate their model, Harvey and Koubek thoroughly reviewed the related literature. The findings from their literature search are summarized in the social and cultural factors review, Section 2.6.3.

Continuing on their previous works, Harvey and Koubek, along with Hammond, examined literature on the design processes and practices that may affect distributed design teams during the design phase of collaborative manufacturing. The authors outline a classic socio-technical framework for the study of distributed engineering collaboration and also propose a model of distributed engineering collaboration based upon their literature findings.

In an effort to better understand their literature search, the authors relied on a sociotechnical theoretical framework, which claims that: "social and technical subsystems of an organization must be optimized jointly for greatest overall performance results." The technical subsystem, consisting of the technology, processes and methods needed to complete design tasks, enables distributed collaboration. The social subsystem, which is comprised of team members, member interaction, team communications, and the relationships between team members, is complex. Often, team members are most comfortable communicating face-to-face and are resistant to interactions with the technology and planning that is necessary for collaborative distribution. The bottom line is that technology alone is not the answer to successful distributed collaboration - the social system must also be considered. [20]

The literature search by Harvey, Koubek and Hammond focuses on two areas: Collaborative Design Teams and Distributed Collaboration. With respect to collaborative design teams, the authors summarize the following major theories and findings:

- There is evidence that suggests that the quality of conclusions, ideas, products, and decisions made by groups often exceeds that of the ablest team member("'N+1 heads are better than N");[20]
- The "groupthink" mentality can often lead to lower quality group decision, as team members may be socially pressured to agree with a lesser decision or are timid to propose their own better solutions;[20] and
- Although there are many different traditional design models for group design, including engineering models, cognitive models, and computational models, the group design process is in general characterized by designers' generating design alternatives, evaluating the alternatives, and selecting the alternative that meets their goals and needs. (The reader is referred to Hammond, Koubek, and Harvey's work for a more detailed discussion of each model type.)[20]

It is important to understand the theories governing collaborative design teams to determine how, where, and why issues may arise when geographic distribution is involved. When geographic distribution is needed, the authors explain that communication greatly impacts distributed collaboration. The authors summarize several key findings from their distributed communication literature search:[20]

- A 1977 study of how communication was affected by telecommunications media found that "...group structure, hierarchy, and interaction patterns (frequency and duration of interaction) that emerge in face-to-face meetings do not occur so readily in distributed communication. It was also clear that people using media richer in nonverbal cues had more favorable impressions of one another and their work."[20]

- "The limitations of communication within a specific medium, or channel capacity, are a function of channel bandwidth, which decreases from face-to-face to video to audio interactions. Face-to-face interactions provide a broad channel as contributors can transmit signals through any of the five senses. However, in video or audio communication, the potential signals are narrowed or eliminated by medium restraints, resulting in decreased efficiency in information transfer."[20]

- "...in the absence of some channels of communication the whole nature of communication is altered."[20]

The authors reviewed several works relative to group dynamics, which are key findings to consider for distributed collaboration. Their findings include:

- Team members feel that the level of participation is generally more equal when a technological interface is used.[20]

- The emergence of a leader is not as prominent in distributed team environments as in well-documented face-to-face environments.[20]

- In general, a technological medium allows timid, less confident team members to feel less intimidated since their teammates are separated by distance (the environment is less personal).[20]

- Distributed team members are more or equally satisfied with their teams when compared with face-to-face teams.[20]

- Group communication in distributed environments involves "Fewer messages, with greater task orientation and less spontaneity..."[20]

- Distributed design teams tend to consider more alternatives with a greater degree of clarity than co-located teams. [20]

- Distributed design teams tend to be more argumentative and tend to have a lower consensus on decisions.[20]

- Fewer messages, with greater task orientation and less spontaneity, were found in distributed communication.[20]

The authors summarized the key literary works that form the basis of their distributed design team communication and decision making conclusions in a table, reproduced in Table 2.2.

Table 2.2: Summary of distributed decision making communications research reproduced from Hammond, Koubek and Harvey, p.47.[20]

Dependent Variables	Researchers	Findings
Time to complete task	Meridith (1997)	Distributed took longer
	Weeks and Chapanis (1976)	Distributed took longer
	Hiltz, Johnson, and Turoff (1986)	Distributed took longer
	Bennison (1988)	Videoconference meetings were shorter with speedier decision making
	Bul and Slvasankaran (1990)*	No difference
Level of participation	Hiltz, Johnson, and Turoff (1986)	Greater equality in distributed
	Williams (1977)	Greater equality in audio than in face-to-face
	Meridith (1997)	Increased participation, less dominance in distributed
	George, Easton, Nunamaker, and Northcroft (1990)*	Greater equality in distributed regardless of task complexity
Emergence of a leader	Hiltz, Johnson, and Turoff (1986)	No leader emerged in distributed
	Williams (1977)	No leader emerged in distributed
	Lim, Raman, and Wei (1990)	More evenly distributed influence in distributed
Satisfaction	Bul and Slvasankaran (1990)*	No difference
	Olaniran (1996)	No difference
	Gallupe, DeSanctis, and Dickson (1988)*	Less satisfied with distributed
	Carmel (1991)*	More satisfied with distributed
Magnitude of communication	Weeks and Chapanis (1976)	Fewer messages and less percentage of time spent communicating in distributed
Type of communication	Bennison (1988)	Less spontaneity, higher degree of task orientation is distributed
	Williams	More argumentative statements in distributed
Number of alternatives	Lewis (1982)*	More alternatives in distributed
	Dennis, Tyran, Vogel, and Nunamaker (1990)	More alternatives in distributed
	Hiltz, Johnson, and Turoff (1986)	More clarification of alternatives in distributed
Consensus	Hiltz, Johnson, and Turoff (1986)	Less agreement in distributed
	Williams (1977)	Less agreement in distributed
	Watson, DeSanctis, and Poole (1988)*	No difference

Note: * signifies as cited in Carey and Kacmar (1997).

Hammond, Koubek and Harvey also reviewed factors traditionally associated with group decision quality, including the magnitude and type of group interactions. Their literature search revealed that the technological medium used for group interactions, in this case teleconferencing communications, influenced the decision quality. They relate: "that subjects perceived information overload when using the teleconferencing communication mode but not in face-to-face. In this case, the increased load of information led to the inability to separate background noise from critical or useful information."[20] The authors further explain, "that mental stress was increased by distributed communication. Such overloads can result in lower quality decisions and mental fatigue for participants."[20]

After reviewing a great deal of literature, the authors propose a model of distributed engineering collaboration founded on the principle that "the design processes of distributed design teams may differ from those of traditional face-to-face groups in some systematic manner."[20] They propose the model outlined in Figure 2-7 to summarize distributed engineering collaboration. Based on their findings, the author's propose that identifying how "interactions and design processes change" in a distributed collaboration environment, as well as how the changes affect the design quality, will enable designer and management interventions, methods, and appropriate training to optimize the the performance of distributed manufacturing enterprises.[20]

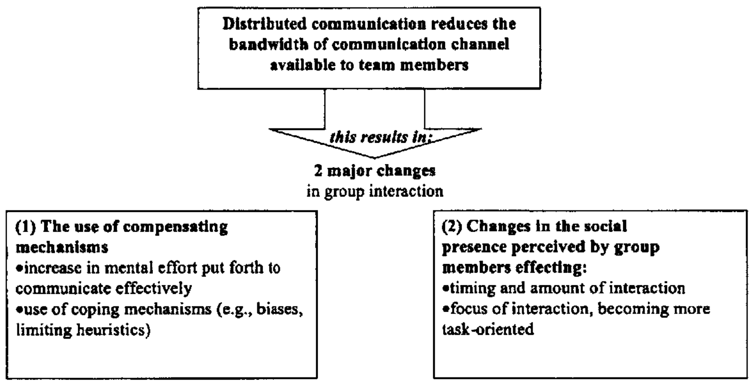


Figure 2-7: Model of distributed engineering collaboration reproduced from Hammond, Koubek and Harvey, p.49.[20]

2.6.1.2 Collaborative Distributed Software Development:

Although some may disagree, several phases of a software engineering (SWE) project are very similar to those of SE projects, including requirements development and flow-down, design, integration and testing. Similar in many ways to the research presented in this thesis, Komi-sirvio and Tihinen surveyed many individuals from countries around the world that participate in distributed software development. The aim of their research was to gain a better understanding of the nature of software engineering in a distributed environment and the problems and issues that have been encountered by distributed SWE participants. Their research purpose was multi-faceted: first, they wanted to rank distributed SWE problem areas by their frequency of occurrence; and second, they wanted to capture the practical experience and lessons learned by those currently performing distributed SWE.[26]

Komi-Sirvio and Tihinen utilized semi-structured questionnaires (sent via mail or e-mail) with both open and closed (multiple choice) questions to survey many participants of distributed software engineering. In all, there were 27 responses from 21 world-wide companies. The survey topics included: characterization of the organization; characterization of the distributed projects; utilization rate of various communication tools; problems and the solutions developed to overcome them; advantages of distribution; and overall satisfaction. In one section, respondents were asked to acknowledge all of the problems they have experienced in their projects from a list of problems identified by the authors in their literature search of distributed software engineering. The respondents were also asked to elaborate on the trouble they encountered, and how the issues were solved (if applicable). Figure 2-8 summarizes the results from that section, adapted from the summary of results in Komi-Sirvio and Tihinen's report. The most interesting and pertinent aspect of this research is that many of the problems themselves are not directly related to software engineering, but instead have to do with simply working in a collaborative, distributed environment (and therefore have application to CDSE). Table 2.3 summarizes the top four problem areas, as discussed by Komi-Sirvio and Tihinen.[26]

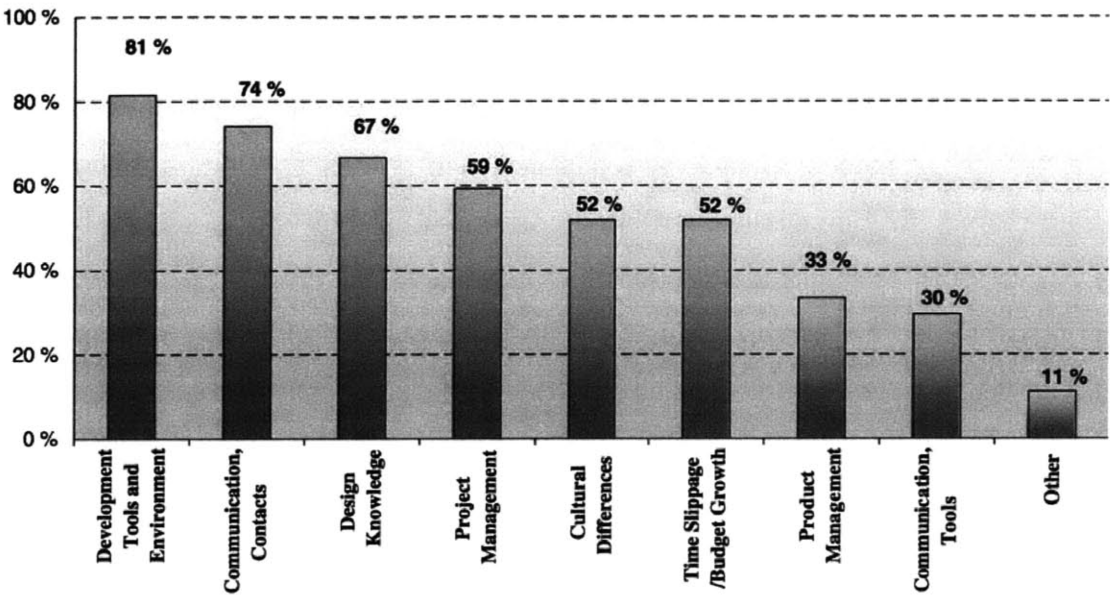


Figure 2-8: Summary of problem areas from distributed SWE survey, reproduced from Komi-Sirvio and Tihinen, p.113.[26]

Problem Area	Problems Encountered	Problem Solutions	Additional Conclusions
Development Tools and Environement	- Network reliability - Network speed - Network usability - Tool compatibility	- Increase site bandwidth - Change development stategy to be asynchronous (each site has own local database, synch with main database 1x per day) - Define and document acceptable tools and versions for the whole project life cycle - Define configuration and version management tools and practices - Get official, explicit approval for the plan from all parties involved - Arrange for the main developer site to take the lead and responsibility for the tool environment and for organizing identical tools for all sites	- Technology and tools may enable distributed SWE, but they are far from perfect. - There is clear need for configuration management tools and processes. - It is critical to establish a compatible SWE environment. - Developers are reluctant to change tools.
Communications, Contacts	- Cultural differences, resulting in misunderstandings - Physical distance, face-to-face meeting frequency decreased with distance - Time zone differences, reduce the opportunity for meetings	- Have informal team-building sessions and face-to-face meetings, especially at the beginning of the project - Decrease the need for contacting other team members by splitting projects into smaller, independent and more manageable units - Appoint a contact person from each site	- Lack of knowledge and misunderstandings has resulted in: redundant work, no work at all, and mistaken assumptions of who is in charge of different stages in the project. - Distance makes it easier to mask possible problems and withdraw from deicion making. - Communication tools are not the problem - if communication is still an issue, we can infer that better communication tools will not fix the problem.
Design Knowledge	- Interpretation of specification, - Understanding of design rationale - Difficult to transfer and share knowledge (especially when leadership was located off-site) - Partner site inompetence - Partner site inability to carry out their development tasks	- Have face-to-face kick-off and technical meetings to discuss design rationale, terminology and application area issues - Divide of work and responsibility into smaller units - Create practical guidelines for developing design documents and using development tools - Have training material provided in electronic form to make it easier for engineers to acquire and use available knowledge	- If architecture design is done at a site different from where the implementation takes place, efforts need be made to ensure that the design rationales are understood and communicated across sites, to verify that they are understood correctly. - Knowledge transfer solely via design documents was regarded as a slow and laborious process - motivating the need for a design document with clear and adequate structure, content and level of detail.
Project Management	- Distributed management may have poorer visibility into problems/issues - Additional unforseen costs for additional up-front planning, travel, communication,etc. - Problem hiding is easier to do in distributed environment - Failure to inform other sites of decisions or changes - Difficulties in getting information requested	- Breaking down project tasks into weekly delivery results is reported to be an efficient way to track the progress of development projects. - Plan development blocks so that they can be independently developed by different sites. (Allowing a local project manager to take over some planning and follow-up activities from the main project manager) - Clearly established rules, definitions of responsibilities, results and timetables along with regular meetings andthe management and control of process are reported to be highly important in a distributed software development environment - Detailed, up-front planning and strict control activities	- Project management challenges are harder to solve in a distributed environment than in a centralized development environment. - For distributed development, significantly more effort is required for up-front planning in order to be able to manage a project successfully.
Cultural Differences	- Different and divergent values cause misunderstandings and dissatisfaction - Misinterpretations of requests and activities - Different commitments to decisions and timetables in diiferent countries	- Enhance communication throughout the project in order to build understanding between the sites - Define and use predefined terminology - Improve language skills and develop and share knowledge concerning cultural issues and customs.	- Training and common sense have an important role in tackling the cultural differences in distributed projects.

Table 2.3: Summary of top four distributed software engineering problem area responses, obtained by Komi-Sirvio and Tihinen.[26]

In another section of their survey, Komi-Sirvio and Tihinen ask respondents to identify the main sources of software errors they have experienced in their projects from a list of possible errors identified by the authors. Respondents were asked to rank the possible error sources on a scale from 1-8, where 1 is the biggest contributor. The results are reproduced in Figure 2-9 and include the responses for errors ranked 1,2, and 3. It is interesting to note that the largest error sources have to do with critical aspects of the SE, including requirements, interfaces, and design.[26]

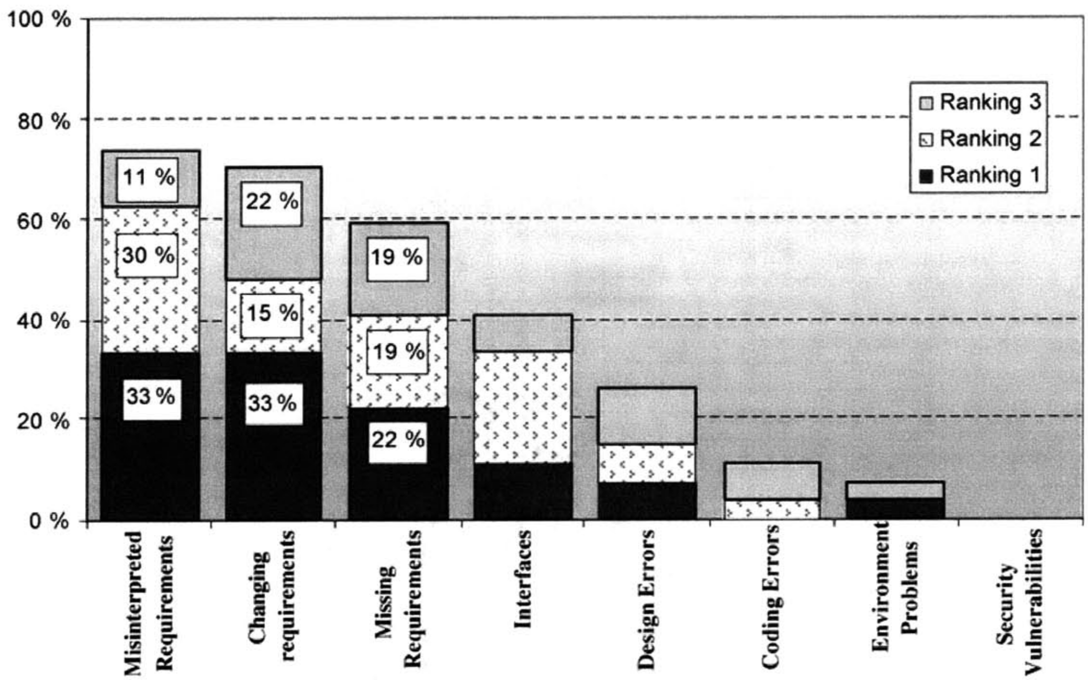


Figure 2-9: Summary of major sources of software error from distributed SWE survey, reproduced from Komi-Sirvio and Tihinen, p.118.[26]

2.6.1.3 Summary

Clearly, based on the relatively few resources available about collaborative distributed engineering processes, there is a great deal of technical, process, architectural, and other CDSE-product related research that is needed to improve CDSE practices.

2.6.2 Collaborative, Distributed Knowledge Management (KM) and Decision Making

Knowledge management and decision making are very difficult in a collaborative distributed environment for many reasons. This subsection summarizes some of the key findings related to collaborative, distributed knowledge management and decision making.

2.6.2.1 Collaborative, Distributed Knowledge Management

Successful KM is critical to the success of CDSE. KM can be described as a: "comprehensive term for providing the right piece of knowledge to the right people at the right time."[18]. There is a great deal of classical research and literature available on KM; however, this research is mostly concerned with KM in the context of a collaborative and/or distributed environment. The reader is referred to the classical works by Nonaka (1994)[33] and Polanyi (1966)[34] for theoretical discussions of knowledge management. Some classic definitions are needed for the discussion of KM in the context of CDSE, including:

- **Knowledge:** The definition of knowledge is widely debated and depends on the context. Here, we use the definition provided by Davenport, et al.[10] Knowledge is "information combined with experience, context, interpretation, and reflection. It is a high-value form of information that is ready to apply to decisions and actions."

- **Explicit Knowledge:** From Nonaka, "Explicit or codified knowledge refers to knowledge that is transmittable in formal, systematic language."[33]

- **Tacit Knowledge:** From Nonaka,"Tacit knowledge has a personal quality, which makes it hard to formalize and communicate. Tacit knowledge is deeply rooted in action, commitment, and involvement in a specific context." [33] Therefore tacit knowledge is difficult to express in linguistic form.

- **Knowledge Management:** The definition of KM, like that of knowledge, is often debated and again depends on the specific context. Fagerstrom and Olsson explain that KM is about: "Exploiting existing knowledge resources, piloting the creation of new knowledge, and integrating new knowledge into exploitable knowledge resources."[18]

Fagerstrom and Olsson discuss KM in a collaborative product development environment consisting of main and sub-supplier relationships, an environment which is very similar to that in the United States defense industry. The authors explain that knowledge is a key strategic factor for a company's future competitiveness. In a CDSE environment like that of the United States aerospace and defense industry, collaborating companies are also competing companies. Successful knowledge sharing and knowledge management practices are key to remaining competitive (protecting company proprietary data) while also working collaboratively with your partners to disclose enough information to have superior products that meet or exceed customer expectations. The authors explain several aspects of knowledge management that are important for collaboration:[18]

- Tacit knowledge embedded in organizational methods and processes and developed from personal experience, is unique and often difficult to imitate.[18]

- The cooperation of both parties is necessary for successful knowledge transfer - knowledge can only be shared if there are two willing parties.[18]

- A shared communication language is essential if knowledge sharing is to take place efficiently.[18]

- People do not tend to absorb large amounts of data or information at once.[18]

- People tend to prefer gaining information first from co-workers. The second choice for information gathering is folders or databases.[18]

- A common understanding of culture and behavior is needed to share knowledge via information technology or knowledge management tools. Computer tools alone only facilitate the exchange of information.

- “Engineers spend as much of 30% of their time searching for and accessing engineering design information.”[18]
- The KM support tools that currently exist for multidisciplinary product design are directed primarily toward the storage and exchange of explicit knowledge concerning processes or projects.[18]

Fagerstrom and Olsson completed a case study of one main supplier and ten of their subsuppliers to better understand the mechanisms that create and distribute knowledge. Through the completion of semi-structured interviews the authors collected a great deal of insight about knowledge transfer in a collaborative product development environment. Key findings relative to CDSE from their study include: [18]

- "Subsuppliers who only have detailed knowledge about the subsystem they will develop and produce seldom become satisfactory members of the main suppliers product development process.” [18]
- Strong product leadership with strong insights into the main suppliers’ process models and quality systems are necessary from the perspective of the subsuppliers in order to become an acceptable member of the main suppliers’ product development team. [18]
- “The main supplier does not always take the subsuppliers knowledge into consideration.” [18]
- “The subsuppliers normally have good knowledge of how to design a system similar to the one they did before. They also know how to make modifications to fit the manufacturing process. However, there are few subsuppliers that could discuss totally new concepts.” [18]
- The knowledge possessed by the subsuppliers was not immediately obvious to the main suppliers, often resulting in communication problems. [18]
- “The most common knowledge spreading among the interviewed companies was spontaneous and informal. It occurred in the everyday work, despite different

computer-based communication tools..." [18]

- Having formal meetings to exchange experiences and transfer knowledge in the cross-functional teams was considered quite positive. Such meetings were employed quarterly in some cases. [18]

- "New knowledge is often generated together with new customers, demands, and problems." An example of this includes the creation of a new, joint production process when additional new demands were made on a subsupplier that resulted in an improved way of working and technical product. [18]

- "Common definitions and terms are important not only for communication between main and subsuppliers, but also for exchanging knowledge." [18]

In discussing their findings, Fagerstrom and Olsson summarize that there is a great deal of tacit knowledge involved in collaborative product development. They propose that a major question resulting from their work, first proposed by Nonoka, is how to transform tacit knowledge into explicit knowledge (knowledge in a usable, transferrable form). The authors make several suggestions to improve the issues uncovered during their case study. One suggestion is to include the subsuppliers in the formulation of the product development process earlier on to better utilize supplier expertise and reduce risk. Another suggestion is to improve the coordination between suppliers - when formal coordination by the main supplier was too slow, subsuppliers informally interact to solve problems, excluding the main supplier from key decisions involving interfaces, etc. The authors suggest that organized formal meetings between suppliers and the main supplier to discuss knowledge and experience have proven to be quite successful and effective. Last, the authors suggest the creation of uniform definitions, processes and models as essential to collaboration success in knowledge management.

Toyota has achieved great success at managing knowledge in their supplier networks, and there are likely several lessons we can learn from them in how they collaborate, teach, and share knowledge with their suppliers. Not only does Toyota excel at sharing explicit knowledge, they have also established an infrastructure and several inter-organization processes that facilitate transferring of Toyota's tacit knowledge. Their trifecta of success includes supplier associations, consulting groups, and voluntary learning teams. The trifecta is summarized in Figure 2-10. The following bullets discusses each feature in greater detail.[15]

- **Supplier Associations:** Foster explicit knowledge sharing by providing a regular forum for Toyota to share information with the suppliers and elicit feedback. General assembly meetings are held bi-monthly and topic committee meetings are held either monthly or bi-monthly.Note, these meetings appear to be similar in nature to CDSE monthly status reports or semi-annual team meetings.[15]

- **Consulting Groups:** Toyota established consulting groups to acquire, store, and diffuse production knowledge. The consultants are a group of highly-experienced senior executives and other production consultants. The consultants are sent to the suppliers, sometimes for lengthy periods of time, to assist companies with implementation of the Toyota Production System, at no charge to the suppliers. All suppliers must share their project results with the other suppliers, as a way to showcase "best practices." Dyer and Hatch, who performed this study of Toyota's suppliers, say that sharing their experiences is "critical because the ability to see a working template dramatically increases the chances that suppliers can successfully replicate that knowledge within their own plants." (And, to protect supplier proprietary information, some areas and topics are off-limits for sharing.) Note that these types of relationships and the showcasing of processes and changes may be critical to the success of CDSE - especially since one CDSE company is typically dictating the methods and processes (not too mention the tools and development environments) that will be used by all CDSE partners or contractors on a project. These consulting groups provide for the transfer of tacit knowledge by experiencing and physically seeing how processes and changes occur.[15]

- **Voluntary Learning:** Teams In some ways similar to the consulting groups, voluntary learning teams are groups of suppliers working together to solve common problems. Small groups of suppliers work together with their shared expertise to address problems at their member plants. Once the problems are solved, the group disperses the lessons learned throughout the entire supplier network. These voluntary learning teams also foster the transfer of tacit knowledge through context-specific learning. Suppliers from other companies literally go into different plants and physically solve a problem - they soon return to their own plant and can implement the same solution from experience. The voluntary learning team method is similar in nature to the process improvement teams used by Company A, which have also achieved great success in knowledge management. (Refer to Section 4.4.4.) This type of team, voluntary learning or process improvement, is proposed for all CDSE teams to solve issues and share knowledge about tools, interfaces, integration, and testing.[15]

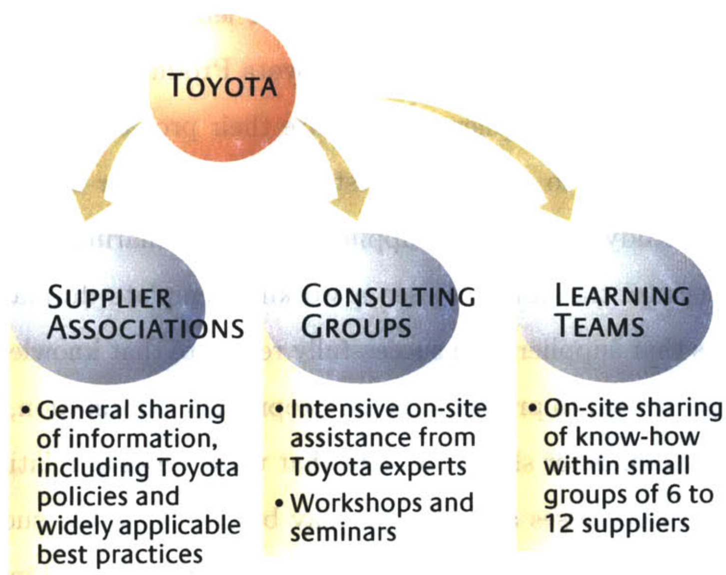


Figure 2-10: Summary of Toyota processes for knowledge sharing and learning, reproduced from Dyer and Hatch.[15]

Obviously, the relationship between Toyota and its suppliers is not identical to the CDSE relationships in the United States aerospace and defense industry, but the methods Toyota uses may be applicable to building relationships between United States defense contractors and of course sharing knowledge (explicit and tacit) and data. In addition to increased knowledge sharing, Dyer and Hatch explain that the Toyota methods have additional benefits when it comes to collaboration relationships. The authors explain that the networks also create a strong identity for the suppliers and when a company receives help, they feel a strong need to reciprocate - both allowing knowledge to flow more freely and building trust between suppliers. However, similar to the United States aerospace and defense industry, the Toyota suppliers are in constant competition, and these networks provide a strong incentive for the supplies to learn and improve quickly. Although United States CDSE companies may be competitors one moment, they are likely to be working as partners the next, and there is still an incentive to share knowledge and help your competitors.[15]

Vizcaino, Piattini, Martinez, and Aranda describe knowledge as "becoming the most important asset of enterprises." Since knowledge is so critical to a company's competitive advantage, the authors suggest using knowledge management to evaluate collaborative applications. In discussing knowledge management, the author's discuss several issues. First, they explain that engineers may be unaware of all of the tacit knowledge they possess, since they obtain it from their daily job experiences without realizing it. Second, they discuss the problem of knowledge transfer. The author's summarize the flows of knowledge into four steps: 1)Socialization: "When tacit knowledge is created from tacit knowledge;" 2) Externalization: "Which requires the expression of tacit knowledge and its translation into comprehensible forms that can be understood by others, for instance, by formalizing it in reports, documents, etc.;" 3) Combination: "When explicit knowledge creates more complex explicit knowledge by combining information that resides in formal sources like documents;" and 4) Internalization: "When explicit knowledge generates tacit knowledge...". These knowledge flows and their relationships to explicit and tacit knowledge are summarized in Figure 2-11, reproduced from Vizcaino, Piattini, Martinez, and Aranda.[42]

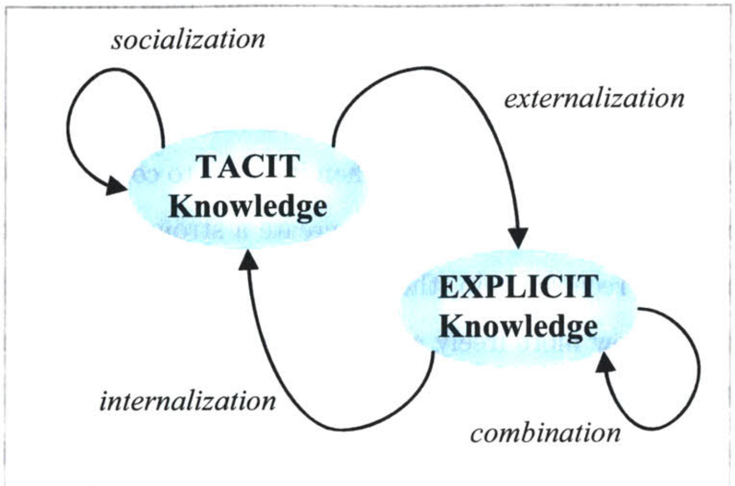


Figure 2-11: Summary of knowledge flows, reproduced from Vizcaino, Piattini, Martinez, and Aranda.[42]

Based on previous works in this field, the authors propose a knowledge management performance index (KMPI) as the metric to use to evaluate collaborative technologies. The KMPI is based on 5 components: knowledge creation, knowledge accumulation, knowledge sharing, knowledge utilization, and knowledge internalization. At the time of writing the authors were planning on using this model to evaluate collaborative systems.[42]

2.6.2.2 Collaborative, Distributed Decision Making

Decision making in a distributed, collaborative environment is difficult. It is often difficult in a traditional environment - but questions over responsibility, who must weigh-in on decisions, issues of budget and schedule, etc. are much more complicated by the addition of distance and multiple players. Hammer and Stanton reviewed a case study at Duke Power, where a more collaborative management style was implemented when Duke Power migrated to a process enterprise. After the collaborative management style was implemented, there was a great deal of confusion about who owned what decisions and what approvals, etc. were needed. To overcome these issues, Duke Power management cooperatively constructed a "decision rights matrix."

This matrix outlined the roles each of the managers would play in each of the major decisions. Included in the matrix was not only which manager would make the actual decision, but also which managers needed to be consulted beforehand, and which managers should be informed after decisions were made. Although this matrix was developed for a completely different purpose, it would likely prove very useful in the currently confusing decision making process (if one even exists) in current CDSE enterprises.[19]

2.6.2.3 Collaborative, Distributed Negotiations

Like knowledge management and decision making, negotiations in a CDSE environment is also a challenge. In a typical "negotiation," concerned parties are together in the same room and use voice discussions along with audio-visual cues (such as whiteboards, drawings, slides, or other documents) marked-up in real-time to reach agreements on issues. In a collaborative distributed environment, voice media is still available (usually via telephone), but the ability to read body language and real-time audio-visual cues are often not. Harris summarizes previous works on distributed negotiations, which concluded the following about important factors influencing negotiations:[22]

- "The ability to see the other person or their environment is not important. If the video channel used is low quality, it can even be detrimental."[22]
- "Good quality voice communication is essential, with no lag, echo, or other distraction."[22]
- "Sketching and drawing mark-up are important, and this must be real-time."[22]
- "The quality of the sketching system must allow for free use of handwriting, and this must occur in real time, reproducing well-formed writing."[22]
- "Conversion of handwriting to computer-generated text, or the ability to key in text, is nowhere near as effective as handwriting."[22]

- "The participants must be able to sketch and write whenever they like - simultaneously if they wish.
- "The work of the participants must be recognizable (e.g., color-coded)."[22]
- "It must be possible to show gesture at least to some extent. It is important to be able to point at parts of the graphic display, and for this pointer position to be seen. Moving the pointer can show a range of gestures."[22]

Many of the above conclusions were implemented in a distributed negotiation environment called ROCOCO, to great success.[22]

2.6.3 Social and Cultural Factors in a CDSE Environment

Lipnack and Stamps summarize that "Successful collaboration is 90 percent people and 10 percent technology."[28] It is evident that the introduction of collaboration technology, distance, different company and locale cultures, and new methods of interaction and communication introduces several new issues to a CDSE environment, not typically experienced in a traditional SE environment.

Vaughn and Fleming presented many social issues and their impacts affecting the distributed engineering team they lead for the development and support of a space-based surveillance system. Their team is distributed across four different states and includes SE, software development, integration and test activities. The authors expanded the work by Patrick Lencioni by summarizing the five dysfunctions of a team and how their distributed team has experienced and dealt with those dysfunctions.[27] A summary description of the five dysfunctions, how they impact the team, why the team experiences them, and how management can overcome them is summarized in Table 2.4. [41]

Dysfunction	What is it?	What impact does it have?	Why do we see it?	How do we deal with it?
Absence of Trust	- Unwillingness to be vulnerable in a group setting - Inability to build a common purpose - Inability to support team goals vs. individual / organizational goals	- Creates an "us vs. them" mentality - Closure of communication pathways - Support organizations "fear" the main organization (not invented here syndrome)	- Wanting to put the best foot forward - hide the negatives - Interfaces are more complex - Cultural differences are harder to see and understand - Assuming everyone has identical goals - Management "down the hall" vs. "across the country" - unequal utilization of team creates lack of trust	- Stay Customer Focused - Promote being a "team player" - Adopt a leadership style that promotes collaboration vs. direction
Fear of Conflict	- Inability to have "passionate debate" over ideas - Creates an "artificial harmony" because conflict stays in the background	- Sometimes the best solutions never make it in front of the group - The adopted solution may not have a high degree of buy-in from the team	- It takes longer to build trust in a virtual environment - Nobody wants to criticize the customer - Nobody wants to have an apparent disagreement in front of the customer	- Leadership must: acknowledge conflict is healthy, involve employees in decision making, look for solutions with high levels of group acceptance - Embrace vertical integration: customer and government are part of team, peer reviews, and problem solving
Lack of Commitment	- Unwillingness to buy into a solution - Perpetuates the "secondary team" aspects of the group instead of the "primary team" goals	- Diminishes the sense of urgency - Impacts morale - Without commitment, the team has no clear mission	- Teams tend to want consensus: without trust, there's no healthy conflict, without conflict, there is no optimal solution, without belief in the solution, there is no commitment	- Leadership must: involve employees in planning, push team consensus, consider all stakeholder inputs, and treat all team members well - Stay performance driven - Be a team player
Avoidance of Accountability	- Unwillingness of team members to adopt a high standard of performance	- Creates resentment among team members with varying standards of performance - Encourages mediocre performance - Necessitates that management be the sole source of discipline (the bad cop)	- Secondary groups within the team cling to their own processes and standards - Team management doesn't always have adequate authority over secondary groups	- Better definition of goals/standards: vertical integration, better communication - Empowerment of "non-management" leadership
Inattention to Results	- Secondary groups tend to focus on their own goals/objectives, rather than those of the Primary group	- Members fail to properly execute the plan - Resentment ensues, because not everyone is working towards the same set of goals	- The team is large enough to have many secondary groups due to geography, different companies and business units, different job functions	- Focus on Execution: get frequent status of goals, performance plans, performance risks - Vertical integration provides more insight into the program as a whole

Table 2.4: Summary of the five dysfunctions of a team and their impact on a collaborative distributed engineering team adapted from Vaughn and Fleming.[41]

Unlike the work of Vaughn and Fleming which was based on practical experience, Harvey and Koubek developed a model for collaborative distributive work based on literature research. Harvey and Koubek extensively reviewed the literature available about collaborative groups, especially cognitive, social and environmental factors. The cognitive attributes relate to how team members individually process information within the collaborative environment. The authors divide the themes of cognitive attributes into several sub-topics: Individual Design Process, Design Representation, and Cognitive Resources. A description of each sub-topic for cognitive attributes is summarized in the following bullets:[23]

- **Individual Design Representation:** The authors summarize that the individual design process is more difficult and complex from simple problem solving because the individual doing the solving does not have a defined initial state. Also, the design process can be summarized as consisting of "goal elaboration, design generation, and design evaluation." [23]

- **Design Representation:** Designers and engineers make pictures, sketches, and drawings to record their ideas share complex concepts, archive design geometry, simulate design, and several other reasons. It is therefore necessary that designs and engineers have a visual to work with. It is important to note that not all drawings and/or sketches have the same purpose or require the same level of detail. It is therefore critical that designers and engineers have available a wide variety of tools to represent their designs.[23]

- **Cognitive Resources:** It is very interesting that the authors note that the individuals chosen for a project are one of the few input variables that can be determined prior to the start of the group project. The authors summarize that the variable of "individual cognitive resources" is one that is little studied but likely has a significance in the distributed collaborative environment. A distributed collaborative environment can contain many complex interfaces and inputs, including collaboration tools, computers, email, conferencing (audio and video), etc... that may make it better suited for some engineers, but not for others. This issue also surfaced as important in this CDSE research and is further emphasized as a key area for future research in Chapter 7.[23]

The authors explain that the social attributes of collaborative group work are very important because many past researchers have hypothesized and demonstrated that the collective performance of a group is better than that of an individual alone. Therefore we need to understand how the social interactions and environment influences group work, to ensure those results are reproducible in a distributed setting. There are many relationships apparent in group work, including "patterns, power structure, division of labor, and interpersonal relationships." To better understand these relationships, Harvey and Koubek summarize two key areas of social factors: Group Cohesion and Group Communication. A description of both sub-topics for social attributes is summarized in the following bullets: [23]

- **Group Cohesion:** Cohesion is a confusing term, but the authors describe two different views: 1) "cohesion is a single construct," that is to say the team is drawn together by a single "socio-emotional" force; and 2) "cohesion is a multidimensional construct," that is many different aspect of the team, including interpersonal, emotional, and task cohesiveness hold the team together. Further research demonstrated the findings that group success required that the members of a group have the ability to solve the problem, each member must be able to defend the solution to the group, and the group must have agreement on the solution. Another study on cohesion found that performance of a group decreases as a result of interpersonal attraction and that pride in the group increased as a function of task commitment.[23]

- **Group Communication:** In a collaborative, distributed design environment, engineers and designers from many different disciplines must communicate. A common vocabulary and communication schema is vital to effective communication.[23]

Lastly, the authors researched the collaboration environmental attributes, including the design task, the group design process, and collaboration technology. The authors discuss the environmental attributes in three sub-topics: the Task, the Group

Design Process, and Collaboration Facilitation. A description of each sub-topic for environmental attributes is summarized in the following bullets: [23]

- **Task:** The authors summarize that tasks are essentially composed of 3 main components: the product (entity created), the steps that are required (to create the product), and information cues (the information that leads to decisions about the product). Several studies believe the task abstraction is the most important element of a task that influences collaboration. But abstracting the group task (to levels of "physical form, physical functions, generalized functions, abstract function, and functional purpose") is just a way to deal with task complexity, which is a large factor that affects collaboration and the distribution of work. To help lessen the complexity, tasks can be broken down into subtasks. Tasks have many dimensions, including their scope (sub-tasks, products, product characteristics, characteristic conflicts, and information), structurability (analyzability, alternatives, and coordination) and uncertainty (internal confidence, external constraints, and random events). In summary, the authors explain that the tasks' complexity and dimensions may determine the tools engineers and designers need when collaborating.[23]

- **Group Design Process:** The authors' discussion on group design focuses mainly on collaborative manufacturing; however the initial design processes appear very similar to SE activities, except the systems engineers do not themselves ultimately manufacture a product. A summary of research performed on collaborating software engineers revealed that during the design process there were verbal discussions on eleven different activities. A table summarizing those activities is reproduced in Table 2.5. Another study of video-based collaborative design interactions revealed the importance of hand gestures, drawings, and the process of creating a drawing in collaborative design.[23]

- **Collaboration Facilitation:** There are two main technological categories for the support of group design processes: Group Decision Support Systems (GDSS) and Group Communication Support Systems (GCSS). GCSS is used for information control, data representation, idea generation, compilation, etc. GDSS is used to support and structure the groups decision process. The authors summarize several studies, including one on the effectiveness of groupware (also called computer supported collaborative work). This study revealed that groupware "improved group task focus, increased time to completion, increased participation, and improved decision quality," as well as decreased group consensus and user satisfaction. There are obviously many shortfalls in the groupware arena, especially in the area of sketching and drawing, graphics, group decision making support, and computer databases. Recognizing these shortfalls, several researchers have developed new tools in attempt to address the holes, including *GroupSketch* (for gesturing with cursors and drawing functions), *GroupDraw* (for structured drawing), and SPIDER (to represent the different perspectives of decision makers to assist with group decision making). Also facilitating distributed collaboration is the emergence of videoconferencing. Videoconferencing allow users to both see and hear those they are working with; however there are several factors that may complicate video-conferencing, such as the task complexity, the number of groups and the group size, and the need for training for video equipment use. However, another controlled study found that the performance and product of collaborative design teams using high quality video was just as good as that of face-to-face teams. Note that in the same study, it was found that collaborative groups using audio only did not have as good results as those teams using video.[23]

Table 2.5: Summary of verbal discussion activities during collaborative design process reproduced from Harvey and Koubek, p385.[23]

Activity area	Activity	Definition
Design activities	Issue	major questions, problems, or aspects of the designed object that need to be addressed
	Alternative	solutions or proposals about the aspects of the designed object
	Criterion	reasons, arguments, or opinions that evaluate an alternative
	Clarification	questions and answers that someone either asked or seemed to misunderstand
Coordination activities	Project Management	statements concerned with the organization of the work, when to meet again, etc.
	Meeting Management	statements concerning orchestrating the meeting time's activity
	Summary	reviews of the state of the design or implementation to date, restating issues, alternatives, and criteria
	Clarification	questions and answers that someone either asked or seemed to misunderstand
	Goal	statement of the purpose of the group's meeting and some of the constraints to work under
	Walkthrough	a gathering of the design so far or the sequence of steps the user will engage in when using the design
Other activities	Digression	members joking, discussion of side topics, or interruptions
	Other	time not categorized by the other categories

In summary, the authors used these literature findings to develop the collaboration model described in Section 2.6.1.1. Obviously the literature studied and the proposed model does not encompass all aspects of distributed collaboration, but it does address several of the important aspects associated with human-side of group design.

2.6.4 Communication (Human-Tool Integration) in a CDSE Environment

Communicating through tools in a CDSE environment is difficult - it is hard to express happiness, anger, satisfaction, frustration, excitement, etc. For the most part, those emotions, critical to team work, decision making, and group consensus, are expressed via body language, tone of voice, and facial expressions. In a distributed collaboration environment where communications may occur using all kinds of tools - such as the telephone, email, chat, videoconferencing, documentation, etc. - a great deal of "personality" is lost.

Harris investigated the issues associated with collaborative SE and SE tool data transfer. Harris explains that SE data transfer is so important because SE is carried out by teams; therefore human communication of ideas, designs, models, rationales, etc. is vital. Of course, much of the communication between systems engineers takes place via tools. Harris introduces the idea of the "Network Agent," since SE data exchange typically takes place in a network-supported team (network connections are imperative to geographically distributed teams). The "Network Agent" (also refereed to as "agent") is therefore the combination of computer, software and operator. To communicate between agents, specific SE tools are typically used, such as system modeling tools, requirements management tools, functional modeling tools, and behavioral modeling tools. Harris explains, just as SE tools are specialized, so are the engineers manipulating them. Often, engineers require a great deal of training and experience to be able to successfully utilize tools to their full potential. It is important to note that without the experience, training, skills, and meaning imparted on the tool by the engineer using it, all that is left in the tool is useless data.[22], [21]

Typically, what is communicated between engineers via SE tools is what Harris calls the "knowledge base" - which is not only the computer-based data, but also the experience and "brain" of the engineer using it. Engineers may use models, diagrams, tables, databases, simulations, etc. to get across the meaning of requirements, for example. On the receiving end, another engineer may reply requesting clarification, or return another computer communication with an additional model or diagram, etc. To make these communication issues clearer, Harris uses the example of a prime contractor (A), developing a system architecture and high-level requirements that are then flowed down to a subsystem developed by a subcontractor (B). A sends B a system model and subsystem and interface requirements via computer tools, emails, phone, fax etc. B responds (via phone, fax, email, and computer) with clarifications about the requirements, and upon completion of system design and development, sends A a completed model and verification of subsystem compliance. Harris also points out that the communications between A and B are really negotiations about the system (for example, interfaces, design, and performance).[22], [21]

How engineers communicate using tools (and the models and data that reside within them) is critical to successful SE. Harris explains that "no two people share the same thought" and given that, if each person is fed the same information, "no two people will form the exact same meaning in their minds." Why? Harris describes that meaning to each engineer is influenced by their respective "training, experience, culture, and context." The engineers "mental model" shapes their understanding. Similarly, if different SE support tools are used by each engineer, the tool itself may add or change the meaning of the same data. When engineers' mental models or the capabilities of a tool environment do not overlap, then information is lost and data may be incomprehensible. The factors affecting shared meaning, Harris concludes, can be affected by two issues: "culture and training" and "validity of model transmission and re-construction." These issues in data exchange and communication must be studied and addressed to facilitate successful CDSE.[22], [21]

Some words of caution...Successful SE data exchange and computer-tools, such as models, diagrams, databases, simulations, etc. may improve the speed of development, engineering understanding, SE process and ultimately even the cost of SE projects. It is important to keep in mind, however, that models may be costly to construct. It is also important to note that with the addition of advanced tool-output (such as models), a lot more responsibility is placed on the engineer to correctly interpret the output (model, etc.). In juxtaposition, when SE outputs were traditionally represented only by text, no meaning or data was lost in the of text documents between engineers (because any word processor is as good as the next). This extra responsibility on the engineer is a motivation for the creation of standardized tools to facilitate CDSE.[22]

Harris concludes with several important findings regarding human-tool integration in SE: first, "There is a considerable amount of work being done in improving the computer-to-computer communication channel. There is much less being done in the human-to-human channel in the specialized domain of systems engineering;" and second, "In agent-to-agent communication, it is the people who supply the creativity, problem-solving, and decision-making role. This is far and away the most important role, but is the one least supported by current developments in network communication."[22]

2.7 CDSE Collaboration Tools and Information Technology (IT)

Hammond, Harvey and Koubek propose: "The success of such collaborative enterprises, to a great degree, lies in the satisfactory manipulation of information technologies to exchange, advance, and utilize information."[20] Almost all sections of this chapter and the chapters that follow are in some way related or reliant upon collaboration tools or information technology. Without either, CDSE can not exist. The need for SE tools for collaboration is evident. Several recent attempts have been made to develop new collaboration tools, expand upon existing tools, facilitate collaboration tool integration, and catalogue existing tool capabilities in order to enhance communication and productivity of collaborating engineers.

Note that collaboration tools are not only needed to work within and between CDSE companies, but also with CDSE customers, typically the United States Department of Defense and/or specific branches of the United States armed forces (Air Force, Navy, Army, Coast Guard, or Marines). In a 2004 Memorandum from a Chief Information

Officer of the United States Air Force, the Air Force expresses the importance and new requirement for a single product data management tool across the Air Force - UGS Corporation's Teamcenter. Teamcenter is the same collaboration tool used for product life-cycle management throughout much of the defense industry. This memorandum demonstrates the vitality and customer buy-in driving the need for information technology and collaboration tools.[14]

2.7.1 Existing Collaboration Tools

Becker, Ballentine, Lee and Townsley state that "The challenge facing many organizations today is how to fulfill the potential of teams *and* information technology." Motivated by global competition and enabled by advances in computers and telecommunication, the authors explain that many companies have developed some form of collaborative work systems to boost productivity. However, even after they have invested a great deal of resources, companies are not achieving the results they expected. Are they using the right information technology? To better understand what type of information technology is currently being used by collaborative distributed teams, a joint venture to gather current IT trends was carried out by the authors sponsored by the University of North Texas' Center for the Study of Work Teams and the Information Systems Research Center. To collect the company information, 35 surveys were collected from 30 companies. There were four main questions to the surveys: 1) What collaborative tools are being used?; 2) How frequently are these tools being used?; 3) How widely are these tools being used?; and 4) Which suite of collaborative tools are most widely used in industry?.[3]

To organize the collaboration tools available, the authors created a taxonomy which includes 18 telecommunication and computing technologies, summarized in Table 2.6.

Table 2.6: Taxonomy of collaborative tools by category, adapted from Becker, Ballentine, Lee and Townsley, p334.[3]

1. Audio Conferencing
2. Collaborative Presentation Software
3. Conference Rooms Videoconferencing
4. Desktop Videoconferencing
5. Discussion Databases
6. Document Management Software
7. Electronic Whiteboarding
8. E-mail/ Electronic Messaging
9. Group Authoring
10. GDSS (Group Decision Support Systems)
11. Group Scheduling and Calendaring
12. Knowledge Management Systems
13. One-way Bulletin Boards (BBS)
14. Personal Communication Tools (includes laptops, cell phones, pagers, etc.)
15. Project Management Software
16. Remote Dial-Up Access
17. Web Browsers
18. Work Flow Management Systems

The results for frequency of tool use is reproduced in Figure 2-12. Note the rankings for frequency: "4=Daily; 3=Weekly; 2=Monthly; and 1=Yearly." The authors report that on average, companies used email, personal communication tools, web browsers, and remote access dial-up on at least a weekly basis.[3]

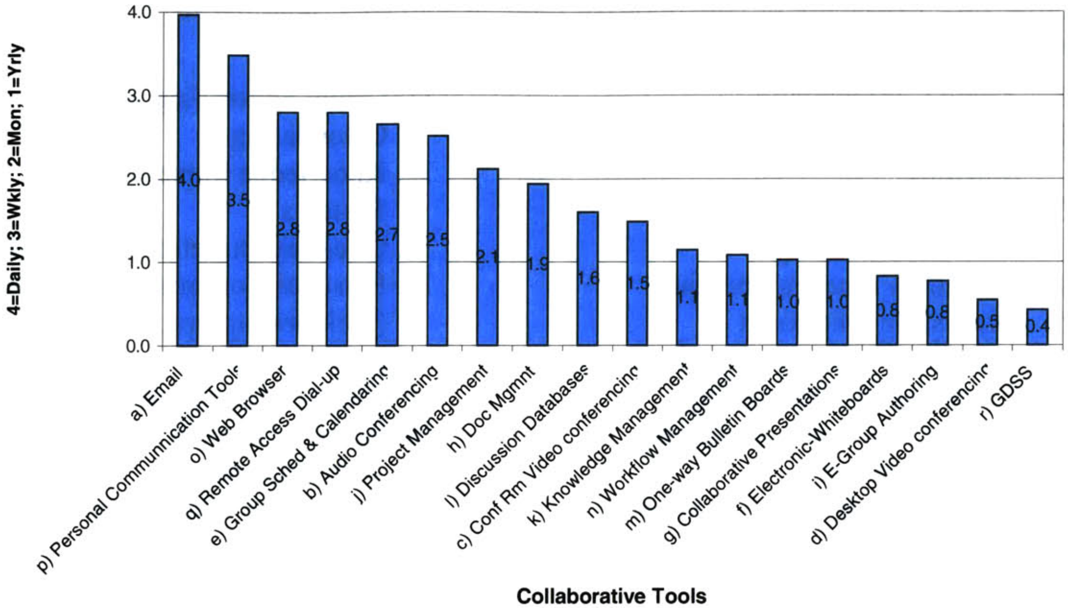


Figure 2-12: Summary of the frequency of use of collaborative tools by category, reproduced from Becker, Ballentine, Lee and Townsley, p335.[3]

There were five tools which were being used by at least 50% of employees industry wide: email, audio conferencing, web browsers, personal communication tools, and one-way bulletin boards (or intranet). The results for employee usage are summarized in Figure 2-13.[3]

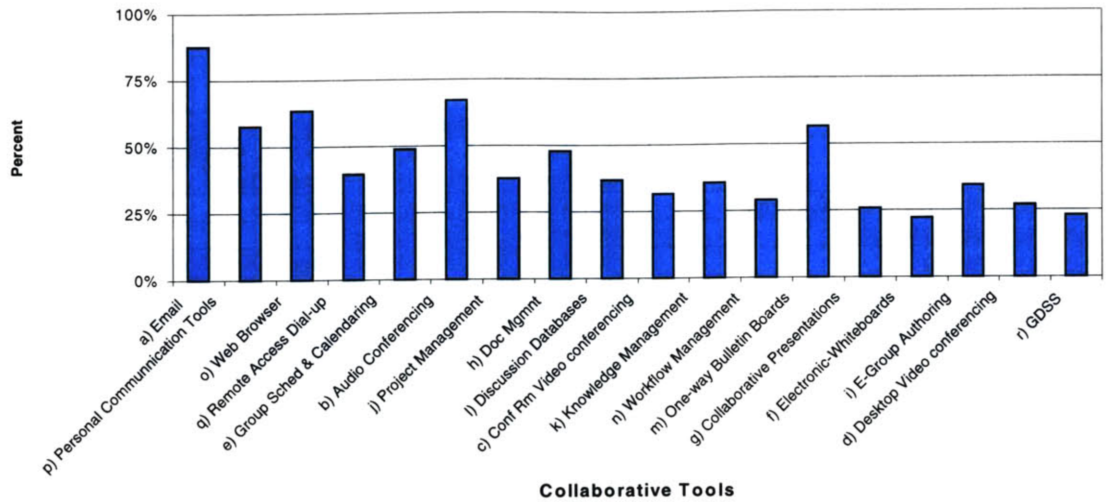


Figure 2-13: Summary of the usage of collaborative tools by category, reproduced from Becker, Ballentine, Lee and Townsley, p336.[3]

To understand pervasiveness of each technology, the authors multiplied the frequency of use for each technology/tool by the percentage of employees that used them. The most pervasive technologies include email, personal communication tools, web browsers, and audio conferencing. The least pervasive technologies were GDSS, desktop videoconferencing, group authoring, work flow management tools, and electronic whiteboards.[3]

Table 2.7 summarizes the tool suites most widely used in industry. Note the frequency of use of Lotus Notes. As the authors summarize, the results obtained in their research are not for a wide-scale industry survey, but they do identify several notable trends that may prove useful for future work on standardized tool suites.[3]

Table 2.7: Summary of tool suites used most often in industry reproduced from Becker, Ballentine, Lee and Townsley, p336.[3]

Category	Industry-wide Collaborative Tools
Email	Lotus Notes, MS Exchange, MS Outlook, Novell GroupWise, AOL
Audio Conferencing	Phone Service, Sprint, Lucent, Meeting Place
Conference Room Videoconferencing	PictureTel, Intel ProShare, Sprint, Eclipse
Desktop Videoconferencing	Intel ProShare, Sprint, NetMeeting
Group Calendaring	Lotus Notes Organizer, MS Exchange, MS Outlook, MS Schedule
Electronic Whiteboarding	SmartBoard, NetMeeting, ProShare
Collaborative Presentations	NetMeeting, PowerPoint, ProShare, Corel, Lotus Notes
Document Management	Lotus Notes, MS Work, Novell File Server, Vantive
Electronic Group Authoring	NetMeeting, PC Docs, MS Word, Lotus Notes, MS Exchange, Corel Suite, Internet Explorer 4.0
Project Management	MS Project, Lotus Notes, Primaveria
Knowledge Management	Lotus Notes, Intranet
Discussion Databases	Lotus Notes, Netscape
One-Way Bulletin Boards	Lotus Notes, ccMail, Inter/Intranet
Workflow Management	Lotus Notes, Workflow, InForms, In-House Applications
Web Browsers	Netscape, Internet Explorer
Personal Communication	Compaq, Dell, and IBM Laptops
Remote Dial-up Access	PC Anywhere, Sprint, MS Dial-Up, RAS
Group Decision Support Systems (GDSS)	Cognos PowerPlay, BPCS, Facilitation

The International Council on Systems Engineering (INCOSE) sponsors an accurate, up-to-date database of a wide variety of tools for SE. The overall database is organized into four separate databases and is available to the public. Each database is formatted as a matrix - for each category of database, several tool features are listed. Each tool in a category has a "full" "part" or "none" attribute indicating whether or not the tool supports a specific feature. The four separate databases are:[39]

1. Requirements Management Tools Survey
2. Systems Architecture Tools Survey
3. Measurement Tools Survey
4. General Tools Database

These databases offer a great deal of information about many tools (over 1,250 Commercial Off-the-Shelf (COTS) and Government Off-the-Shelf tools) and could provide a very useful reference to SE management and program leadership in determining which tool(s) is best for their team. Also, as of the beginning of 2005, tool vendors can update existing tool features and add new tools to the database directly.[39]

2.7.2 New and Proposed Collaboration Tools

As distributed collaboration becomes more widespread, more and more tools are being proposed and developed to deal with specific applications. The following paragraphs summarize some new, proposed, and attempted collaboration tools that may prove useful for CDSE.

Under the European Distributed Software Engineering project, Espinosa and Drira have developed a model and implemented a software package to support cooperation activities for design review meetings for geographically separated teams. To facilitate development of their tool, Espinosa and Drira specified participant roles (such as chairman, secretary, etc.), eight collaboration rules (such as what participants can and cannot do and in which order they must do them) and relationships between participants, including their dependencies (inhibit, enable, precedence). A session management service was developed to manage the dependencies between participants, and includes an instant messaging tool to facilitate coordination between sites. When using the tool, each participant and site is assigned a role, and the actions they can complete during a session depends on their role and location (i.e. Only the session "chairman" can open or close a session.) Once the model was developed, Espinosa and Drira applied the model to a Preliminary Design Review (PDR) activity, which was composed of many sessions. At the time the article was written, the tool still needed further updates to be useful from multiple distributed cites.[16]

Expanding on their previous work, Espinosa and Drira along with Villemur, developed a role-based cooperation scheduling system in an effort to support synchronous group sessions (where participants are acting simultaneously from distributed points of access on shared materials). The tool distinguished participants based on their skill set, role, and by topic and allowed for both an open group access (anyone can be in session at any time) and an invitation-based group access. The tool was tested and validated for a PDR scenario with participants acting distributively from Turin, Munich, and Paris. To facilitate the distributed PDR, the tool allows for the creation of a review "groups" from the input of participants, automatically sends invitations to selected participants, and is used to catalogue and organize all of the comments and concerns input by the reviewers.[17]

Zittel has reviewed the past and current collaboration tool endeavors to establish another "new tool", or suite of tools, for "Simulation Based (Systems) Engineering (SBA - Simulation Based Acquisition)," also called Collaborative Based Acquisition (CBA)." Figure 2-14 from Zittel summarizes the activities and relationships between various SBA actors and components. The basic idea, already in use by some major defense contractors, is that all of the simulations for a system (including hardware, software, requirements, performance, cost, and management) are integrated together to reduce overall program cost and schedule and increase fidelity and performance of the final product. These types of tools are already in use in both of the companies examined in this thesis. As Zittel further explains, many commercial enterprises have already begun using such tools and have documented "improved performance of a shortened development schedule, reduced cost and improved system quality." Zittel explains that in 1999 the United States Defense Advanced Research Projects Agency (DARPA) was testing a newer SBA systems that incorporated additional features. The system under-test combined 3-D graphical design with the mathematical and physical equations governing the system, as well as the associated costs, materials, manufacturing and logistical implementation of the system - allowing for improved calculations of life-cycle cost and accurate re-calculations of all of the above when the design changes.[43]

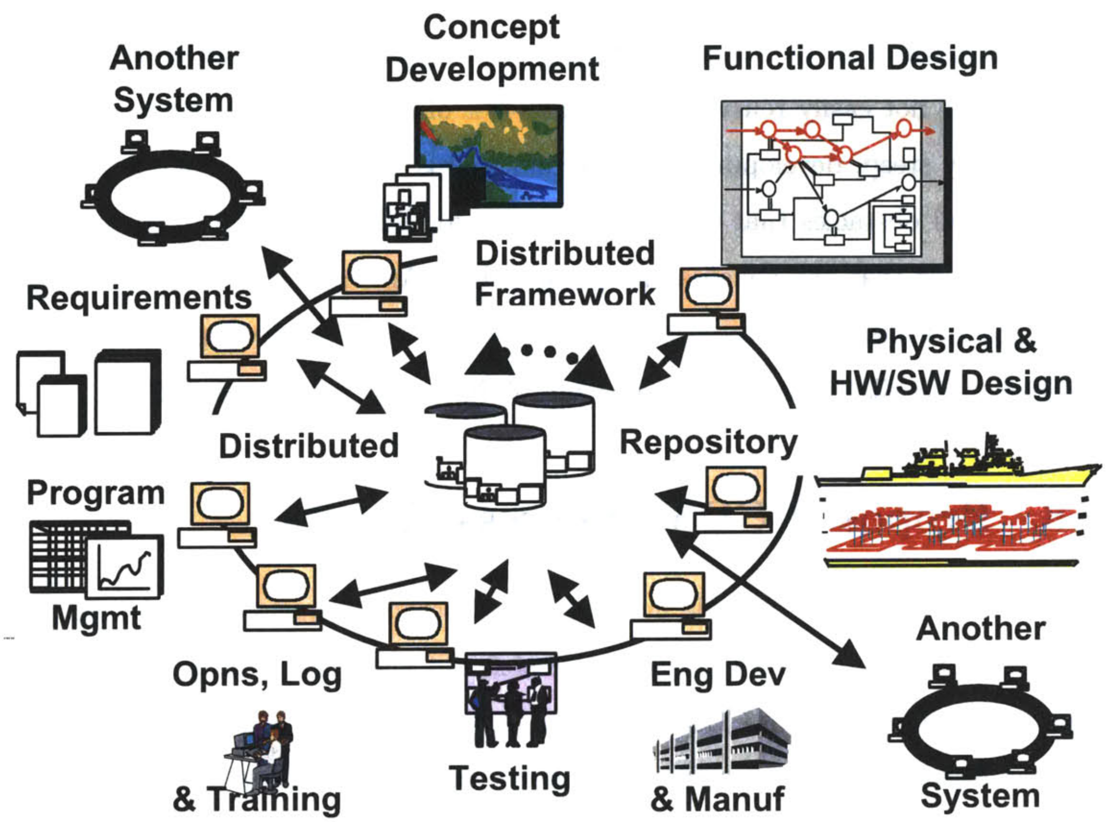


Figure 2-14: Summary of Simulation Based Acquisition activities, actors and their relationships reproduced from Zittel.[43]

Tools such as SBA or CBA allow for high-fidelity requirements analysis and design, which are the early-life-cycle critical functions (typically performed by systems engineers) that all development and costing in a program is based on. Since the SBA and CBA tools integrate all aspects of a system, each CDSE site can enter there aspect of the system simulation and run the simulations via real-time networking. However, there are several technical, cultural and managerial challenges that remain, including acceptance of these new tools. Clearly, especially if development of these simulations is distributed, the interfaces and overlaps between subsystem simulations are key and difficult to coordinate. Another issue is proprietary data sharing between companies. If all aspects of the subsystem design are disclosed in the simulations (for fidelity), and any company or CDSE site can view and run the simulation, proprietary information may be compromised. One specific challenge for the field of SE, is that these tools and simulations are often very tailored to a specific product or subsystem, and are therefore not easily reusable. Since the cost of development of these systems is already very steep, development of a second tool is often just as much as the first. These are all obstacles that must be overcome for SBA or CBA to take hold, but Zittel explains that the benefits are innumerable. Among them, improved designs, easier product integration, money savings accelerated product development processes, and better establishment and management of SE life cycle functions.[43]

2.8 Proposed CDSE Success Factors

CDSE "success factors" include not only what methods, processes, tools, etc. that have been proven to make a project successful, but also those lessons learned about what not to do to make a CDSE project a failure. Throughout this chapter several "lessons learned" and "suggestions for improvements" have been stated for CDSE-related research and factor. To summarize, these success factors include, but are not limited to:

- **Establish Trust:** Trust enables open communications between team members, and inspires confidence in the final product and cooperation between teams.[28]
- **Invest in up-front Planning Activities:** Spending more time on the front-end activities and gaining team consensus shortens the implementation cycle (and especially avoids the pitfalls they may occur if issues of team mistrust, conflict, and mistakes surface during implementation).[28]
- **Perform Visual Management of the Development Process:** A TPDS success factor described by Cleveland, visual management of the development process, may be useful in establishing a sense of team, as well as keeping the team immediately up-to-date on important programmatic and product related issues. This visual management may be possible by using the collaboration tools or environments and/or team rooms displays. Imagine signing on to a

collaborative environment, and upon logging in, immediately being informed of a subsystems current testing or development status (perhaps in red, yellow, green). Or similarly, entering a CDSE team room to find the color-coded schedule progress of each team on an LCD display. These visual cues provide immediate feedback without having to scour schedules, requirements, or test data and would be relatively simple to implement.[8]

- **Define Decision Making Responsibilities:** Hammer and Stanton described Duke Power's triumph when using a "decision rights matrix" to make collaborative management decisions. As described in Section 2.6.2.2, this matrix outlined the roles each of the managers would play in each of the major decisions. Included in the matrix was not only which manager would make the actual decision, but also which managers needed to be consulted beforehand, and which should be informed after decisions were made. Although this matrix was developed for a completely different purpose, it would likely prove very useful in the currently confusing decision making process (if one even exists) in current CDSE enterprises.[19]

- **Establish Clear Methods, Teams and Practices for KM:** Similar to the Toyota supplier relations network discussed in Section 2.6.2.1. These relationships improve not only explicit KM management, but also tacit KM.

- **Improve Collaborative Distributed Team Functions:** Avoid the "five dysfunctions of teams" and follow the advice and lessons learned by Vaughn and Fleming, summarized in the "How do we deal with it" column in Table 2.4 in Section 2.6.3.

- **Provide CDSE Training:** Training can make a huge impact. Take the example of the GE 6sigma black belts - GE recognized the need for virtual teaming as a future key mission critical need, and has trained all of their "Black Belts" since 1998 in virtual teaming. The same type of training can be used throughout SE organizations for how to work in CDSE environment.[28]

- **Maintain Standard, Defined Terms:** Harvey and Koubek found that a common language and vocabulary were necessary for collaboration success.

- **Have a Clear Mission and Goals:** Team leadership needs to provide the distributed teams with a common vision and purpose as well as goals. Doing so aligns the organization and creates a sense of purpose, team, and identity.[28]

- **Implement Lessons Learned from Distributed SWE:** The participant in Komi-Sirvio and Tihinen's study of distributed SWE made recommendations for improving their work environment and the issues they encountered. Refer to Table 2.3, "Problem Solutions" and "Additional Conclusions" column, in Section 2.6.1.2.

2.9 Possible CDSE Barriers

Unfortunately, success in CDSE can not be achieved without first overcoming several possible barriers. Interestingly, research has shown that distribution itself contributes to the issues encountered by SE teams. Distributed, collaborative projects inherit the same issues and problems that co-located, single-site SE efforts struggle with (budget, schedule, quality, risk etc.)- only the issues are complicated further by distribution and the addition of other players. Komi-Sirvio and Tihinen reference a recent study which concluded that the physical distance between development sites alone is likely to create additional delays in work. Other studies of collaboration referenced by Komi-Sirvio and Tihinen report that distributed collaboration also results in additional issues of task collaboration, project management, and communication.[26]

In addition to distribution and distance alone, the following lists summarizes several additional potential issues and barriers that have surfaced in the CDSE-related literature:

- **Face-to-Face Communication Preferred:** Most people still prefer talking to some one face-to-face, or at least over the phone.[28]

- **Difficult to Build Trust:** Lack of trust between teams, especially those from other companies with possible ulterior motives. Trust is more difficult to build in a distributed collaborative environment. [28]

- **Issues Encountered from Distributed SWE:** The participants in Komi-Sirvio and Tihinen's study of distributed SWE listed several problems they encountered that are just as likely to occur in CDSE. See the "Problems Encountered" column in Table 2.3 in Section 2.6.1.2.

- **Too many Perspectives:** Although toted as a CDSE benefit, research has demonstrated that the diversity of knowledge held by collaborating systems engineers (or any collaborating teams with diverse experiential and intellectual backgrounds) can also be a barrier to successful knowledge sharing. It is difficult to share and understand knowledge when engineers do not share the same social, occupational or cultural background. This is because different experiential and intellectual backgrounds can lead to different "perspectives, priorities, typical approaches to problem solving, and even terminology." These differences can often be overcome when collaborators work together frequently in highly interactive settings. However, in distributed collaboration, engineers are limited in their face-to-face contacts and the collaboration settings are not highly interactive or very frequent. [37]

- **Five Dysfunctions of Teams:** The issues encountered by Vaughn and Fleming in their distributed collaborative aerospace and defense project are also very likely to occur in CDSE. Refer to Table 2.4 in Section 2.6.3.

- **Collaboration Environment is Complex:** "The complexity of the distributed collaboration environment leads one to conclude it too may present a cognitive problem for individuals."[23]

- **Time Zones/Misaligned Schedules:** "Real-time collaboration is often an issue, especially when teams are distributed across different time zones. It is difficult enough when different organizations and locations have different daily schedules, but those differences are severely complicated when time differences are introduced."[28]

Chapter 3

Research Methods

To date, there is a minimal amount of previous literature and research available directly on the topic of CDSE - most research has focused on a specific element of CDSE, as summarized in Chapter 2. This research, therefore, is an exploration of the current CDSE practices in industry, encompassing many social and technical factors. This chapter explains the research methods used to collect and analyze field data on current CDSE practices in the United States aerospace and defense industry.

3.1 Research Methods Overview

Key CDSE characteristics were examined by completing two case studies in the aerospace and defense industry. The two case studies were carried out independently at two different United States aerospace and defense companies. Data about the companies and their CDSE practices was collected by performing semi-structured interviews with systems engineers, system engineering management, and SE support staff, collectively referred to as "SE personnel." In total, 21 interviews with useful data were completed, each ranging from one to three hours in length. Where possible, supporting project documentation was collected to substantiate and bolster the information collected via interviews. Interview methods and questions were sanctioned by the MIT Committee on the Use of Humans as Experimental Subjects (COUHES). Please note, to protect company and individual anonymity, all data collected from the organizations and individuals is "coded" (Company "A", Engineer "A" etc.).

Although exploratory research typically has the goal to develop grounded theory[38], this exploratory research serves to document the current state of CDSE practices in the aerospace and defense industry and indicate key areas where future work is required. By examining the current state of United States aerospace and defense company CDSE practices, the lessons that have been learned, collaboration success factors and methods developed, and indicators of possible best practices can be analyzed and recorded.

The following sections describe in more detail how the two case studies were selected, the case study level of analysis, the research sample, interview design, and methods of data collection and analysis.

3.2 Case Study Selection

An online search of several major United States aerospace and defense programs was performed to determine those companies who were likely performing CDSE. Participation in this CDSE study was then solicited by email from several of the major United States aerospace and defense enterprises by contacting the SE department leads from those companies. However, scheduling, programmatic issues, and company location made participation for many companies difficult. The collection of data was carried out over the summer (May-September) of 2006; which is typically a popular vacation time and program milestone delivery time (with the closing of a fiscal year in June). Additionally, my affiliation with one major United States aerospace and defense company made it difficult to gain entrance into competing organizations (for reasons of access to proprietary data and conflicts of interest). Therefore the two case studies were selected based on the companies': current execution of a CDSE project, willingness to participate, company schedule alignment with my research agenda, and the companies' proximity to Boston.

3.3 Research Level of Analysis

This research examines the CDSE efforts and methods at the program level of two different CDSE organizations. Although the CDSE efforts take place over several geographic locations and companies, all interviewees for each case study work for the participating company (A or B) within the same facility (Although many interviewed personnel travel between collaborating locations.). Table 3.1 summarizes the company and program information for Company A (or Program A) and Company B (or Program B). Note that the information has been generalized to protect the anonymity of the participating companies.

Company Information	Company A	Company B
Overall Company/Business Unit Size	13,000+ people	1000+
Number of Company/Business Unit Locations	18	5
Program Information	**Program A**	**Program B**
Overall Program Size within Company	1800+ people	350+ people
Projected Program Lifecycle Time	12 years	10 years
Program Customer	US Government	US Government
Project Type	Defense	Defense
Company Role in Program	Prime Mission Systems Integrator	Prime Contractor and System Integrator
SE Stage of Development	Detailed Design and Integration	Detailed Design
Number of Major Collaborating Companies (Including Company)	5	6
Approximate # of Collaboration Sites	Multiple Companies in over 45 states	Multiple Companies in over 5 US states and the UK
Collaboration Locations for CDSE Efforts	Massachusetts, California, New Jersey, Washington D.C., New Hampshire, Florida, Indiana, Maryland, Virginia, Colorado, Mississippi, New York, Minnesota	Massachusetts, California, Florida, Indiana, Minnesota, UK
Time Zones Involved for CDSE Efforts	4: EST, CST, MST, PST	4: EST, CST, PST, WET

Table 3.1: Summary of case study company and program information.

The focus of this research is the two companies' current CDSE practices and the lessons they have learned; this research did not examine how the collaborations were introduced, structured, or the contract bidding process.

The research conducted at Company A focused mainly on the Systems Requirements Cross Product Team (CPT), although interviews were conducted with personnel in the overarching SE organization and with systems engineers at the subsystem level of engineering to get a better breadth and depth of the SE organization and practices. Whereas the research conducted at Company B predominantly examined a wider breadth of CDSE practices. Interviews were completed with overarching SE-related program leadership and several task leaders and support staff from several Program B SE-related organizations (systems, subsystem, integration).

3.4 Research Sample

To get a better understanding of all of the aspects of each companies' CDSE practices, interviews were requested with each programs' SE management (including program managers), systems engineers, and system engineering support staff (information technology, configuration management, tool support, etc.). Initial interview appointments were facilitated by a SE contact at each company. At each interview, additional contact information for potential interview candidates was requested. In total 21 interviews were completed, including nine individual interviews at Company A, six individual interviews at Company B, and a group leadership interview of six program managers and SE technical directors from Company B.

Table 3.2 summarizes the engineering/leadersip breakdown of the research sample. Note that SE support staff were not included in this tally.

Table 3.2: Summary of research sample engineering/leadership breakdown by company.

Category	Company A	Company B	% of Total Sample
Practicing Systems Engineers	6	4	55.6
Management (Program Managers or SE-related Technical Directors)	2	6	44.4

In Table 3.3, the category of "Practicing Systems Engineers" from Table 3.2, is further broken down by job title (Team Lead, Task Lead). Additionally, the SE support staff are included in the tally for each company. It is important to note that due to the wide range of skills involved in systems development, not all engineers on the programs' SE organization charts belong to SE organizations. In this sample, several engineers supporting or leading CDSE teams were from mechanical engineering, electrical engineering, and software engineering disciplines. Last, note that although support staff interviewees may also be considered task leads or SE management, they were considered as SE support staff in this tally.

Table 3.3: Summary of total research sample breakdown by company.

Category	Company A	Company B	% of Total Sample
Integrated Product Team/ Cross Product Team Leads	4	0	19.0
Management (Program Managers or SE-related Technical Directors)	2	6	38.1
Task Leads	2	4	28.6
SE Support Staff	1	2	14.3

Obviously missing from the summary in Table 3.3 is the category "Systems Engineers" - the research sample was made up predominantly of experienced systems engineers with responsibility for several other product development engineers. Figure 3-1 summarizes a breakdown of the years of SE experience for all practicing systems engineers. (Note, that the experiential background of Program Managers and engineers from other engineering disciplines are not included in this figure. Additionally, note that data was not available on the specific years of SE experience from the six participants in the Company B group leadership interview.)

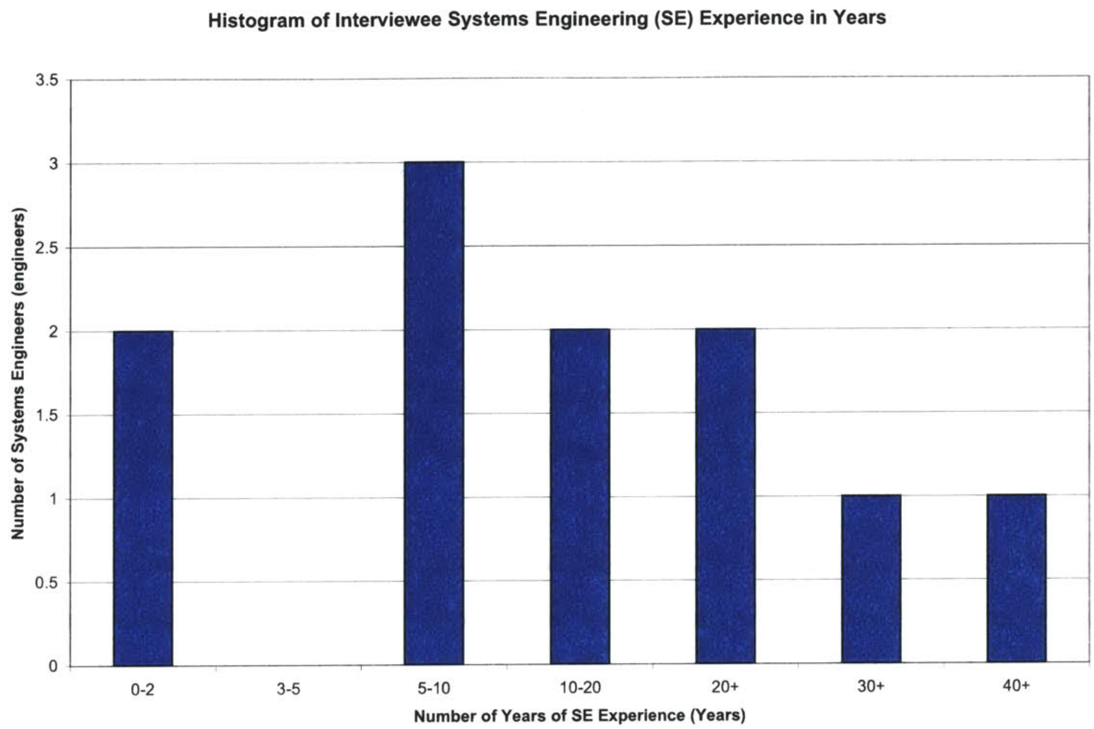


Figure 3-1: Summary of the number of years of SE experience for interviewee research sample.

Excluding the six participants in the Company B group leadership interview because data was unavailable, none of the engineers in this research had formal educational training in SE. The systems engineers, management, and SE support staff in this sample have a wide range of formal educational training, including: electrical engineering, aerospace engineering, physics, computer science, mathematics, dynamics, mechanical engineering, software development, ocean engineering, and biomedical engineering.

3.5 Interview Design

Initial questions for the CDSE interviews were identified based on the key areas discussed in the CDSE literature review, summarized in Chapter 2. After development of the initial questions, pilot interviews were conducted with five SE experts in the field (20+ years of SE experience). During these pilot interviews, questions were refined, added, deleted or moved based on the experts' personal experiences and feedback. Also during the pilot interviews, the need for an SE leadership-specific interview template was identified. An SE/Program leadership template was therefore developed to include many of the same questions as the standard SE interview questions, in addition to questions seeking information on basic company information, program information, and programmatic data (cost, schedule, metrics, performance, etc.).

Interviews were designed to last approximately 45 minutes to one hour and required no previous preparation on the part of the interviewee. The interview format was semi-structured, and the questions were written to be open-ended, allowing the interviewee to embellish with examples and personal experiences where possible. Appendix A contains the standard SE interview template. There are eight sections in the standard interview template, including:

1. Basic Information about Interviewee
2. Current Collaboration Situation
3. Collaboration Tools and Information Technology
4. Knowledge and Data Management
5. Technical Product Issues
6. Social and Cultural Effects
7. CDSE Benefits
8. CDSE Motivation, Success, and Future Work

Appendix B includes the SE Leadership interview template. The interview is very similar to the standard interview, but includes two additional sections:

1. Basic Information about Interviewee
2. **Program-Specific Information**
3. Current Collaboration Situation
4. **Programmatic Issues**
5. Collaboration Tools and Information Technology
6. Knowledge and Data Management
7. Technical Product Issues
8. Social and Cultural Effects
9. CDSE Benefits
10. CDSE Motivation, Success, and Future Work

Since the speciality and the experience of the support staff was not known in advance, no specific template was developed for these interviews. These interviews were much less structured, but consisted of a standard set of initial questions, including the interviewees' basic information and current job function, job lessons learned, issues encountered, and a request for recommendations on program improvements and future work.

Due to the schedule constraints of the program and SE leadership team at Company B, a tailored interview template was created to allow six program managers and SE technical directors (TD) of Program B to be interviewed in one hour. These questions were developed as a subset of the SE Leadership interview template and can be found in Appendix C. These questions were mostly used to set the "jumping off" point for discussions on each of the major CDSE factors discussed in the structured SE Leadership interview template.

3.6 Data Collection

Each interview, with the exception of the group leadership interview at Company B, was conducted in the interviewee's office on-site at the interviewee's company. (The group interview was conducted in a conference room at Company B.) All interviews were pre-scheduled and took place at their pre-determined time. Due to the defense nature of the programs and companies involved in this research, laptops and audio or video recording was not permitted during the interviews. Therefore, all interviewee responses were captured via hand-written notes.

An interview protocol was developed to ensure consistency across all of the interviews and the data collection. In all interviews, questions were read as stated in the interview templates summarized in Appendix A, Appendix B or Appendix C. Further information was then provided if the interviewee was confused as to the meaning of the question. The interview protocol (a checklist) that was used for all of the interviews is included in Appendix D.

A standard interview proceeded as follows:

1. Short introduction and exchange of business cards.
2. Provided interviewee with short summary of CDSE research. See short CDSE research summary in Appendix E.
3. Obtained informed consent from interviewee, as required by COUHES. See Appendix F for a copy of the consent form.
4. Explained interview process and post process to interviewee. (See the following paragraph).
5. Executed Interview.

The interview process consisted of asking the interviewee questions and taking hand-written notes of their responses. Unless the interviewee's answer was a standard "yes" or "no," I would often rephrase the interviewee's response back to the interviewee to ensure that the interviewee's response was accurately understood and recorded. Due to the interviewees' time constraints, in some instances, the complete list of questions was not covered if interviews lasted more than the scheduled hour and the interviewee had a meeting immediately following the interview. Following the interview, the hand-written notes were transcribed to a typed transcript (4-10 pages in length) that followed the flow of the interview and was organized by the interview topic headings. The typed transcript was then sent via email to the interviewee, where approval of the transcription was requested to ensure all responses were accurately represented. All interview transcriptions were approved after modifications, deletions, and comments from the interviewees.

Data collection was performed slightly differently for the group interview. To assist with the multiple responses and discussions by the six program and SE managers, a colleague assisted in the note-taking process. However, the interview process (although different questions) and the transcription process were identical to the individual interviews.

Where possible, information obtained from interviewees was bolstered by additional program documentation, observing engineers in the field, and searching the publicly available documentation. This additional information is not directly cited, to protect the identities of the participating organization and engineers.

Upon completion of the data collection process, there are sixteen typed, interviewee-approved transcripts - one for each of the fifteen individual interviews and one for the group interview. The transcripts are coded to protect the identity of each company and the participating interviewees.

3.7 Data Analysis

Data analysis was performed the same for each of the two case studies and was facilitated by the organization of the interview questions and transcripts. The coded transcripts were organized according to the interview question heading topics, similar to those explained in Section 3.5. The interview heading topics are:

1. Collaboration Situation and Management
2. Collaboration Tool Use
3. Knowledge, Data and Decision Management
4. SE Processes and Practices
5. CDSE Social and Cultural Environment
6. CDSE Benefits and Motivation

Case study data presented in Chapter 4 and Chapter 5 were initially analyzed and summarized separately. The data analysis process was as follows:

1. For each case study, the coded data from each interviewee was grouped together according to interview question topic.
2. Once all data was organized into interview heading topics, the transcripts were manually coded by interviewee experience, specifically: description, issue or barrier, recommendation, lesson learned, success factor, or irrelevant.
3. Once all data was organized by topic and experience, the data was further coded and grouped into common subtopics, for example "trust," "email" or "tool training." There was not a defined list of subtopics, as they varied from topic to topic and were sometimes unique to the case study.

Figure 3-2 graphically depicts the case study data analysis process used, with a specific example as it applies to Case Study B, interview topic heading "Collaboration Tool Use," interviewee experience "Issue or Barrier," and several associated subtopics.

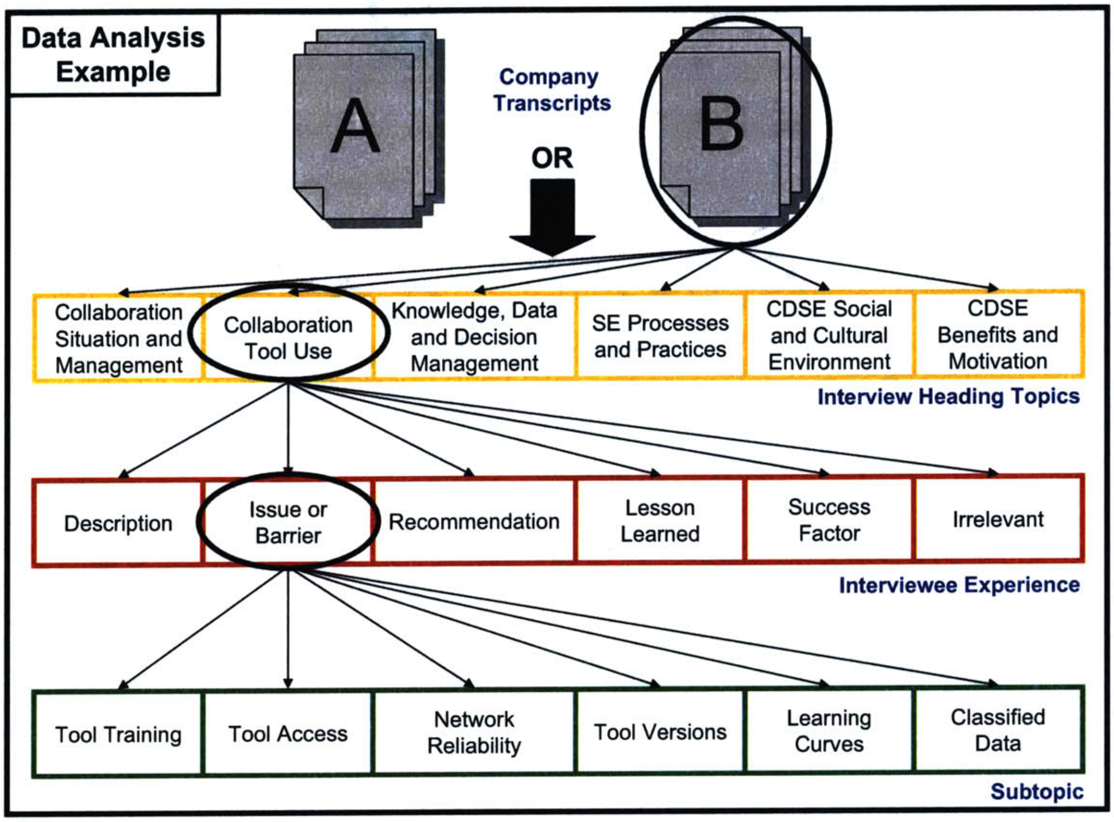


Figure 3-2: Case study data analysis process with collaboration tool use issue example.

As shown in Chapter 4 and Chapter 5, data is presented in the format it has been analyzed. All interviewee responses, unless irrelevant to the topics being examined, have been included (Including conflicting interviewee perspectives on subtopics; for example, CDSE effects on product integrity.).

The data from each case study was also compared and consolidated, as summarized in Chapter 6. For the consolidated data analysis, all of the data from each case study was compared at the "experience" level of analysis for each interview heading topic. For example, all of the uncovered issues were compared for each interview heading topic. Only "key" issues, success factors, benefits, and motivations were included in the consolidated analysis, where "key" refers to those issues, success factors, benefits, and motivations that are applicable to other programs. In some instances, the issues, success factors, benefits, and motivations discussed with interviewees were very specific to the program and/or company they worked for. Consolidated data was annotated with "*'s", indicating whether the specific finding originated in Company A, Company B, or both A and B. Table 3.4 depicts the format the consolidated case study data analysis is presented in.

Table 3.4: Example consolidated case study data analysis format.

	CDSE Issues and Potential Barriers	A	B	A and B
1	Key Issue 1.			X
2	Key Issue 2.	X		
3	Key Issue 3.	X		
4	Key Issue 4.		X	

Due to the several limitations on the research findings (described in Section 3.8), the consolidated data is meant to be used as "soft indicators" of potential CDSE issues, success factors, benefits, and motivations and to identify areas of future research.

3.8 Limitations of Findings

As this research is one of the first of its kind in the area of CDSE, it represents a narrow focus of the current CDSE practices in industry. This fact, coupled with the defense nature of these programs, imposes several limitations on the findings of this CDSE research, including:

1. **Small Sample Size:** Due to time constraints and the accessibility and availability of personnel, data was collected from a relatively small sample size of systems engineers. Therefore, although the data collected is indicative of trends in CDSE, the findings from this research may have limited applicability to other programs or organizations if not further investigated with a larger sample size.

2. **No Proprietary Data:** Access to and use of company or program proprietary data was prohibited. Therefore, discussions of the technical product, product

architectures, and certain analysis and design methods are missing from this research.

3. **No Classified Data:** For obvious reasons, access to and use of classified data was prohibited. Similar to the issues with proprietary data, interesting aspects of the technical products and architectures cannot be included in this research.

4. **Data Limited to Single-company:** Although many companies participate in each programs' CDSE efforts, access to interviewees only at the participating local organization was possible. This is a key area for future research (See Chapter 7) and the findings may obviously be biased toward the inputs from the contributing companies.

5. **Interviewed only SE Managers and Leads:** Almost all of the interviewees were Program or SE leadership, or SE middle management (integrated product team or cross product team leads or task managers). There may be different perspectives or issues encountered by less experienced or less responsible engineers that are not represented in this research. This is also a key area for future research (See Chapter 7).

6. **Examined Limited Breadth, Depth, and Lifecycle Phase of SE Activities:** Research was conducted with limited SE activity teams in each organization, as described in Section 3.3 and Section 3.4. For Company A, a more in-depth study was done with a single CPT (depth but not breadth); whereas in Company B, interviews were conducted with personnel from many subteams (breadth but not depth). Similarly, each case study was conducted during a subset of the SE development phase. Programs A and B are at different program maturity levels and phases of development. Therefore it is difficult to make direct comparisons.

Chapter 4

Case Study 1: Company A

This chapter summarizes the raw data collected from Company A. Included in this chapter is a company and program overview; a description of the current SE collaboration situation; a summary of the tools currently in use by Company A; an overview of Company A's knowledge, data, and decision making practices; a summary of the current collaborative SE practices and processes; the social and cultural experiences of interviewees; and a summary of the CDSE benefits and motivations as related by the Company A interviewees.

Please note the following concerning the material presented in this chapter:

- The personnel interviewed for this case study consists of: systems engineers, or engineers performing system-engineering related activities (they may belong to organizations other than SE); SE support staff, such as tool administrators or process experts; and SE or program management. For lack of a better term, the interviewees are collectively referred to in this report as, "SE personnel".
- As described in Section 2.1, the term "SE" in this research encompasses many of the overall product development related activities, including requirements development, system conceptualization and definition, system integration and qualification, and system simulation. Although many people or organizations have different definitions for SE, the term "SE team" is used to refer to all

personnel involved with the aforementioned system activities.

- Many of the SE factors discussed with interviewees are very closely related. Although all efforts have been made to separate interviewee comments into the topic areas they are most closely related to, there is some repetition between sections. (Some interviewee inputs are relative to many of the factors being discussed.)

- The identified issues, success factors, recommendations, benefits, and motivations summarized in this section are based on the data provided by a relatively small sample of Program A SE personnel, and are not meant to be declarative of program-wide applications (i.e. issues, lessons learned, motivations, etc.). However, the information discussed herein was provided by interviewees working in Company A, and on Program A, and is therefore an indication of Company A and Program A applications (i.e. issues, lessons learned, motivations, etc.).

4.1 Company and Program Overview

Company A is divided into several business segments, with office locations throughout the United States and the world. This case study was sponsored by Company A's "defense" business segment. The defense business segment employs over 13,000 personnel and has 18 office facilities scattered across the globe. Although there is a great deal of information publicly available about Company A and Program A, these details have been omitted to protect the company's identity and any program-related technical data.

Company A is currently acting as the prime mission systems integrator for Program A (the United States government is the design agent). Program A involves the development (from product conceptualization to delivery of the completed system to the customer) of an entirely new, cutting-edge technology system for the United States government. In total, Program A employs over 1800 personnel from Company A.

The development of Product A has been organized into several phases with a total projected product development lifecycle time of approximately twelve years. For this phase of Product A's development, there are five major United States aerospace and defense contractors collaborating across the United States (including Company A). To complete the overall Product A development, over 80 companies, including contractors, subcontractors, and suppliers, are collaborating across 45 states. For the CDSE efforts discussed in this research, collaboration takes place between companies located in Massachusetts, California, New Jersey, Washington D.C., New Hampshire, Florida, Indiana, Maryland, Virginia, Colorado, Mississippi, New York, and Minnesota. Currently, Program A is in the detailed design and integration phase of development, although various elements in the system are in slightly earlier development stages.

The Program A organization and structure (including tools and processes) has varied slightly between development phases, as contractors have changed roles and the program has in general evolved and matured. Figure 4-1 summarizes the top level general organization of Program A in the current development phase, adapted from Program A internal data.

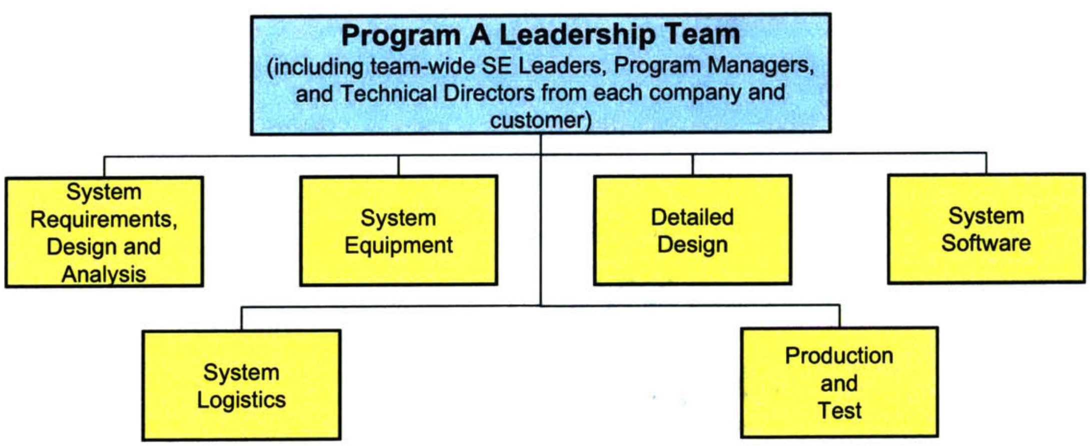


Figure 4-1: Summary of general Program A top-level organization.

4.1.1 Scope of CDSE Research within Program A

This research focuses mainly on the Systems Requirements Cross Product Team (CPT), although interviews were conducted with personnel in the overarching SE organization and with engineers at the subsystem level of SE to get a better breadth and depth of the SE organization and practices. The Systems Requirements CPT falls under the Systems Requirements, Design and Analysis (SRDA) team organization (See Figure 4-1). The SRDA team is responsible for five of the major Program A SE activities, including the establishment and management of system requirements, overall system design and analysis, system modeling and simulation, and human-system integration. The SRDA team also supports all other major SE and product development activities (supporting a total of 7 additional SE activities). The System Requirements CPT, as its name infers, is primarily responsible for the establish and management of the overarching system requirements. At this current phase of development, the System Requirements CPT is performing requirements maintenance.

The Systems Requirements CPT is broken into eight major subteams (specific to the product architecture and capabilities), each with a requirements area lead. Tool, quality, cost, process, metrics, and performance excellence support personnel also comprise the major organization of the System Requirements CPT. As described in Section 3.4, the Program A SE personnel sample interviewed for this research is predominantly made up of experienced system engineers acting as team or task leads. Several interviewees were retired United States military officers and all were educated in engineering or science.

4.1.2 Program A Status

The SE leadership believes Program A and the SE efforts thus far on the program have been very successful. The Program A team has successfully completed and tested almost a dozen engineering development prototypes to demonstrate the feasibility and performance of the new cutting-edge technology. Additionally, the program has successfully completed many key milestones with accolades from the customer. Compared to other SE programs one SE leader has worked on, he believes this program has been the most successful. Not only is it well-managed, but he believes a lot of "thinking ahead" work was done early on to make the program run smoothly. This early work has also helped to contain defects "in phase," saving lots of time and money.

The SE leadership believes a large part of the Program A success is due to close monitoring of key metrics. At team meetings and performance reviews, SE team members are informed of the metrics used to evaluate the team performance by the customer, so metric alignment and traceability is clear. Metrics are collected and portrayed in such a way as to clearly indicate the relative performance of each of the teams and requirements documents in development. The primary metric for success on the program are the reviews themselves, both internal peer reviews and formal customer reviews. Additionally, progress is measured using technical metrics such as activity and sequence diagrams; design analysis; and cost and schedule metrics such as the Earned Value Management System (EVMS). Because the metrics are widely available and distributed (to the contractor and customer community), a sense of peer pressure is felt by those under-performing. The System Requirements CPT even has a metric review committee, which meets monthly to record and monitor the CPT's progress.

The Program A SE team is currently completing SE tasks nearly on schedule and close to budget. Note, that although there are no specific metrics used to directly measure the cost or schedule impact of collaborating, it is clear that the System Requirements CPT has been impacted by the large collaboration effort. Despite its great success, the SE efforts from the previous development phase were slightly overbudget, mostly due to lack of prior experience managing a program of this magnitude and scope. Using SE processes different from those traditionally used by Company A to accommodate collaboration and program size caused additional cost and schedule slips (there is a learning curve effect). Additional cost and schedule slips have also been incurred as the result of requirements and design changes requested by the customer.

Program A SE Leadership also believes the great success of the program can be attributed to the specific resources that were allocated for collaboration from Program A inception - including both time and money. Several additional resources are allocated to collaboration - including a customer collaboration center near customer headquarters and funding for support of the development, implementation, and use of development and collaboration tools.

It is important to note that Company A process improvement initiatives have been used extensively over the Program A history to cut costs, meet deadlines, and improve areas where recommendations have been suggested. All personnel staffed on Program A must complete process improvement training within 90 days of joining the team.

4.2 SE Collaboration Situation and Management

The CDSE collaboration situation is different for many collaborating engineers, depending on their position (leader, task leader, line engineer, etc.), the geographic distribution of their team, the team make-up, and the SE task or activity they are involved in. This section summarizes some of the collaboration situations, management techniques, and SE environmental differences experienced by the SE personnel interviewed from Company A working on Program A.

4.2.1 General SE Collaboration Situation

The SE efforts for Program A are managed and coordinated by the Systems Engineering Management Plan (SEMP). The SEMP outlines the tools and processes that all of the Program A team members must follow (at all companies and locations). Although the SEMP documentation is widely available, it has been in general difficult for the SE leadership to get all team members "on-board" with the processes and procedures outlined in the SEMP. However, enforcement of the SEMP has not been an issue - its following is outlined in all of the contracts and Statements of Work (SOW) between Company A and the other contractors and subcontractors.

Program A system design, analysis, and development is performed by a combination of engineers whom are co-located and working distributively using collaboration tools to facilitate their communication. All interviewed personnel worked with engineers whom are located off-site and from other companies. One engineer interviewed worked *only* with off-site engineers. Face-to-face meetings due occur periodically as needed and normally result from periodic reviews (in-phase reviews, peer reviews, milestone reviews, etc.). A key tactic used by the System Requirements CPT was the rotation of these reviews between companies, to allow each company to take an active role, and to also share the cost and schedule burden of traveling. For the most part, due to time and cost constraints, company or program sponsored social face-to-face meetings are not scheduled between collaborating companies or locations. However, occasional "fun" meetings between engineers at the same location do occur to foster team-building.

A recent addition to the Product A development efforts is a new customer collaboration center located near the customers' headquarters. In addition to the collaboration taking place between Program A SE sites across the United States, approximately 60-70 engineers and Program A leaders from all companies convene at this new collaboration facility to address design issues and work intimately with the customer at the customer's convenience. This new center was designed to increase customer satisfaction and overall participation in Program A.

On a daily basis, all interviewed engineers report having to work with off-site engineers either via email, telephone, teleconference, or another collaboration tool. One engineer estimated that he communicates or collaborates with engineers at other development sites approximately 10-20 times per day, mostly via email. Almost all interviewed engineers and leaders also reported spending at least 50% or more of their time performing collaboration-type activities, such as talking on the phone, transferring data between classified and unclassified systems and/or collaboration tools, responding to emails, tracking down engineers, typing meeting minutes, etc. (This phenomena may be attributed to the fact that most interviewees perform some sort of management function.) Also, additional time must be allotted prior to meetings to prepare the tools needed for meetings (Like reserving and arriving at a conference room hour early, for example.). It is important to note however, that the additional preparation time before meetings allows the meetings to begin on-time and maximizes the effectiveness of the limited amount of time key stakeholders can spend together.

A great deal of the SE data and activities for Program A involves classified data communications, which additionally complicates the communication practices of the collaborating personnel, as many of their communications must take place via secure means.

As the Program A SE team is spread across all time zones, time differences are yet another factor that complicates SE communications. Although all systems engineers and SE leaders claimed that their work was not significantly impacted by time differences, they did report that meetings and teleconferences did have to be scheduled to accommodate the different arrival, lunch, and departure hours of the different time zones. However, one of the tool support engineers mentioned that time differences were actually a blessing for her team. With the time differences in their favor, the tool support team could address issues in the eastern time zone first, and then before the engineers in the pacific time zones even arrive at work, begin addressing their issues.

4.2.2 System Requirements CPT SE Collaboration Situation

The Program A system requirements development efforts are coordinated and governed by the Requirements Development Plan (RDP), which augments the information in the SEMP as it applies to the System Requirements CPT. The RDP outlines the requirements development processes to be used by all system requirements developers from all companies and locations. The RDP, like the SEMP, is a formal, configuration managed document, and its following is mandated contractually by all teams. The RDP describes the process, hierarchy, development tools, configuration management protocol, requirements review process and the required characteristics and attributes for requirements development. The RDP is supported by specific work instructions and program directives which further describe the details of key SE processes and practices.

The System Requirements CPT has developed and now owns responsibility for over 13,000 system level requirements. As the System Requirements CPT is primarily performing requirements maintenance in the current development phase, (i.e., engineers are making corrections, modifications, insertions, clarifications, or deletions to the requirements based on customer needs, program constraints, and system modeling, testing, and integration results), the requirements configuration control processes and directives are very important to the work of the Systems Requirements CPT.

Each of the eight subteam requirements area leads hosts a requirements change review board (RCRB) meeting at least once a week to discuss recorded issues, new issues, and changes proposed for the requirements. Change requests or issues concerning requirements are managed via a web-based trouble reporting tool (WTRT). One engineer explained that discussions (via email, telephone, or teleconference) about the requirements occur all week leading up to the RCRB meeting. During the RCRB meetings, the key stakeholders come together to agree to the requirements updates. An example issue is the agreement about the wording of requirements - the requirements developers may want the requirements worded one way and the subcontractor another. For some subteams, the meetings are coordinated by a "Configuration Manager" (a member of the Program A configuration management team), who sets up the "virtual meeting" by arranging a teleconference and the necessary tools (typically the WTRT system and requirements management tools) to conduct the meeting. A teleconference bridge and a collaboration tool that enables the sharing of computer desk-tops (so that all meeting participants at every location are able to see the same presentations, data, or tools) is used to coordinate the virtual meetings.

The System Requirements CPT is also responsible for the development of the system interfaces, which is a difficult task when engineers and developers are themselves distributed geographically. The interfaces in the Program A system are grouped into electrical, mechanical, or logical interfaces. To help sort out the development and design of the system interfaces, a system interfaces working group has been formed and is made up of representatives from all of the major system components. The interfaces working group meets weekly for approximately one hour to address the interface issues as a group. One engineer explained that part of the difficulty in developing and maintaining the system interfaces requirements is the fact that for some requirements changes, multiple parties (at least six, representing the major system components as well as the customer representatives) must agree to the interface requirements changes.

To facilitate smooth meetings, some System Requirements CPT members perform additional work before and after the RCRB and working group meetings. For example, one engineer explained how he organizes these meetings by sending an email agenda the day before the meeting, which lists the issues to be discussed, the people who are required to attend, and the document/issue reference locations in an online database where the relevant documents can be found. To maximize the usefulness of everyone's time during meetings, the engineer permits those that can resolve their issue or action item off-line to send him an email with the resolution, and they need not attend the meeting. Preparation before and after meetings is not trivial: one engineer estimated that he spends up to an hour preparing the weekly status/agenda email before the meeting, and after the meetings, he spends an additional hour summarizing the meeting minutes and posting them to an online database.

4.2.3 SE Collaboration Management

Successful coordination and management of the CDSE work has developed over several years and after encountering many issues. The following is a list of some of the practices that have been adapted over time to manage the SE collaboration:

- **Hold Weekly Team Meetings to Disseminate Information:** Team meetings are held weekly for all of the SE team leaders, at which time key system-wide issues, processes, and decisions are discussed. At these meetings, additional Program A information and decisions are described (Typically, discussions center on information passed down from Program A program management and technical directors). It is the responsibility of the SE team leaders to disseminate the information shared at these meetings with their teams. At the SE weekly team meetings, 100% attendance by all SE team leads is required (if a lead cannot attend, he/she must send a representative).

- **Up-Front Work Necessary:** Successful CDSE needs a good amount of up-front planning, including defining the management role, understanding the contract (scope), determining the resources needed for collaboration, setting out the key milestones, determining the metrics, and defining the responsible parties.

- **Identify Key Subject Matter Experts for System Coordination:** Key subject matter experts (SME) are identified as the contact people for specific requirements books, system components, or system performance capabilities. These SME's are there to weigh-in on decisions, provide additional support and information to the development teams, and also to ensure consistency across the system with the customer's needs.

- **Establish and Mandate Cross-Company Formal Processes:** Cross-company formal processes are established and mandated (such as the SEMP, RDP and RCRB). In general, these processes were originally adapted from the Company A internal product development process, and were bolstered by the shared experiences and inputs of all collaborating companies.

- **Share "In-house" Developed Tools:** "In-house" developed Company A tools (such as WTRT) are shared with other companies and sites and mandated for use. Using the "in-house" tools resulted in several benefits: the tools are tailorable; Company A already has experience with them; and there is an already established tool support system to ensure their smooth operation.

- **Provide Common Collaboration Tools:** Docushares and other online ("firewall protected") collaboration tools are used to share information and maintain configuration control. These tools are available and used at all sites and provide both classified and unclassified data storage and transfer between sites.

- **Include Customer as Integral Team Member:** The customer and customer technical representatives are included as an integral part of the SE teams to maintain customer-driven focus and to mediate cross-company issues as they arise.

- **Standardize Templates and Development Plans:** Formal development plans and standardized document templates, such as the RDP and the requirements development work instructions, are written and agreed upon by all key stakeholders as the "formalized process."

- **Have Defined Metrics to Measure Program Progress:** Program progress is measured and tracked through the use of formal processes, such as Earned Value Management System (EVMS) and an Integrated Master Schedule (IMS) (with a clearly defined critical path), and carefully selected development metrics (open action items, # requirements, etc.).

- **Establish Processes to Update/Change Requirements and Artifacts:** SE leadership additionally hosts weekly system configuration control board (SCCB) meetings, where key stakeholders come together to approve and discuss changes to system level specifications, architecture, SE plans and SE work instructions.

- **Publicize SE Processes Across Sites:** To assist with dissemination of SE information, changes, and SE processes, Program A SE leadership invented the concept of "process days," where leadership tours different development sites to give presentations about SE processes and process changes.

4.2.4 SE Collaboration Differences compared to Traditional SE Environment

The SE personnel interviewed describe several key differences between working in a collaborative, distributed SE environment and working in a traditional SE environment. The following list summarizes some of the differences, as described by the engineers and leaders:

- The SE processes are actually followed to a much greater extent due to their better definition, the contractual obligation to follow them, and the wide-spread dissemination of the process materials in the CDSE environment.
- The customer is involved to a much greater extent to help provide focus and to mediate issues between companies and development sites.
- New and different processes are structured to accommodate the differences between company-internal processes.
- Untraditional channels and methods are used to enforce all stakeholders to use the agreed upon processes. For example, the management from each company agreed to the common requirements process, however an engineer from a different company working the program may refuse to follow a specified process because it is contrary to the way their company normally does business. To remedy this situation, the upper management from one company would need to interface with the upper management of the dissenting company in order to enforce the agreed upon process.

- Raw data metrics are used to show the customer the relative performance of each company. Rather than polishing or finessing the metrics, the SE metrics experts use the raw data to honestly portray to the customer the status of each key criterion, and thus the status/progress of each company's contributions. No single team has the capability to "hide" problem areas or finesse the data in such a way as to bolster performance.

- New and different SE "middle leadership" positions are created to oversee and coordinate SE development efforts.

- Program A has introduced distributed collaboration tools to better facilitate development efforts, even better than those performed in a traditional environment.

- Engineers rely on the telephone and email to a much greater extent to get their work done.

- The traditional development tools used have been adapted to facilitate collaboration and to bolster finished-product consistency across all of the distributed sites. For example, the DOORS interface requirements modules are set up to automatically generate the requirements that go into the requirements documents for each "end" of an interface.

4.3 SE Collaboration Tool Use

The discussions with SE personnel on collaboration tools was centered on their use in the daily practice of SE, potential issues and barriers personnel have encountered, and success factors that have been developed to improve collaboration tool use. The paragraphs in this section summarize the input from interviewees on the aforementioned topics.

4.3.1 Collaboration Tool Use Overview

The overarching processes agreed to by all collaborators, such as the SEMP and subservient plans and work instructions, call out the tools and processes to be used for development. The SEMP lists over 33 tools to facilitate SE activities, such as tools for requirements management, cost estimation, configuration control, and issue reporting. However, the SE personnel interviewed consistently referred to the following list of tools as those they used on a daily basis, many of which are obvious and not included in the SEMP tool list:

- **DOORS (with DXL add-in):** DOORS is a Commercial Off-the-Shelf (COTS) software tool distributed by Telelogic aimed specifically at requirements management. The System Requirements CPT used DOORS extensively to develop and manage over 13,000 Program A system level requirements. Some engineers say they "hate DOORS but love it," since it has a very non-intuitive set-up, but has wonderful features like change tracking, history recording, access history logs, and "rights" for editing privileges, all allowing for user accountability. There is an unclassified (where possible) and classified DOORS database on Program A. DXL is an extension language to DOORS and allows control of DOORS programming by executing scripts that are similar in nature to C-code. The Program A tool support team has created extensive DXL scripts to automate a great deal of DOORS upkeep, and also perform tasks that provide greater artifact consistency and stability. DOORS is discussed in more detail throughout this SE Collaboration Tool Use section.

- **Teamcenter Community and Enterprise:** Teamcenter has 2 versions, Teamcenter Enterprise, which is used predominantly for configuration management of documents and contractual-related materials; and Teamcenter Community, which is a docushare, complete with instant messaging, team calendars, notepads, message centers, and other collaboration tools. The Teamcenter Community tool is especially useful because it allows sharing of computer desktops and net-meeting host capabilities behind a secure firewall. This is very important because it allows sharing of program sensitive (contract sensitive) data, which is not allowed to be shared with other standard net-meeting tools, such as Sametime, over an open internet. There are both classified versions and unclassified versions of these tools available. UGS Corporation creates and distributes Teamcenter.

- **Rational Rose:** Rational Rose is an IBM product used for system modeling, idea generation, and has some applications for software management. One SE leader described Rational Rose as a "giant scratchpad" allowing engineers to develop, trace, and examine requirements. This tool is available for on the classified networks only.

- **Sametime:** Sametime is an IBM product used in conjunction with Lotus Notes (which is predominantly used as an email tool by Company A) and provides capabilities for online conferencing and instant messaging. Currently it is available only on the unclassified network.

- **Web-based Trouble Reporting Tool (WTRT):** WTRT is a "home-grown" tool developed and maintained by Company A used for tracking issues and managing the requirements configuration control process. It is used to document and trace all changes to the requirements. The WTRT tool is also distributed and linked to all of the distribution sites via Teamcenter Community. There are classified versions and unclassified versions of this tool available.

- **Microsoft Office Tool Suite:** The Microsoft Office tool suite includes Microsoft Word, PowerPoint, and Excel. These tools are used informally to support team communications and organization. For example Excel is used often to manage the team (scheduling, metric collection and graph generation, easy system analysis). Whereas PowerPoint is used often to support team communications (presentations, reviews, data sharing). Some engineers use Excel extensively to record weekly meeting minutes, WTRT status, and for action item tracking after meetings. There are classified versions and unclassified versions of these tools available.

- **Action Item Tracking System (AITS):** AITS is accessible by all Program A users in Company A and is used by the Program A tools support team to address issues and outages with the collaboration tools, including DOORS. This tool is available on the unclassified network only. This tool is not yet available at all sites.

- **Rational ClearCase:** ClearCase is an IBM tool used to provide software configuration management and version control. This tool is available on the classified network only.

- **Email:** Company A predominantly uses Lotus Notes for email and messaging, a software program developed by a subsidiary of IBM. Since many team members work from different locations and in different time zones, email is a major form of communication and knowledge dissemination. Unfortunately, other Program A collaborating companies, including the customer, use Microsoft Outlook. This unfortunate difference in tools makes it difficult to use common features, such as the calendar or meeting invitation capabilities. Lotus Notes is available only on the unclassified network. Email on the classified network is available, but has limited utility since it is typically only available on a particular small closed network.

- **Teleconferencing System:** Although easily overlooked, the teleconference system is a key enabler for virtual meetings and CDSE. The ability to have a dedicated open phone line where teams from across the United States can call in and "conference" at the same time is critical to many of the weekly meetings, design reviews, contractual discussions, and RCRB meetings. The teleconference system is currently only available and useful on the unclassified network. Classified teleconferencing is available, but the voice quality tends to be very poor. Classified teleconferencing is typically foregone in favor of an unclassified meeting with documents as backup (to refer to when classified numbers/data needs to be referred to).

Net-meeting is an additional tool available for use on the classified network, since the network itself is encrypted and firewalled. However, to use this capability, all the customers, contractors, and subcontractors must have classified network access, which is not the case.

It is important to point out that there are different types of "networks" in use on Program A. Wherever possible, work and communications take place on an unclassified network, via standard internet and company firewalled communication networks. However, most of the requirements development and all software development is stored and takes place on a secure, classified network (and in secure team rooms), with limited classified communication between sites made possible by encrypted classified networks.

It is also important to note that many of the collaboration tools used by the System Requirements CPT and by Program A in general were originally set-up and controlled by another contractor during an earlier phase of the Program A contract. Since there was already a precedent and established history (including tool training, tool support, and account creation processes) associated with these tools, use of these tools was continued by Company A when they took over as the system integrator in the current phase of Program A development.

All interviewed system engineers and SE leaders found the tools they use to do their job incredibly useful. To communicate with off-site and even co-located engineers, many engineers feel email is the preferred form of communication. There are several reasons stated for this preference:

- Sending email creates a record, whereas when compared with telephone conversations, there is a great deal more accountability.
- With email, you can have a discussion or send a conversation to an entire

distribution list (whereas, with the telephone, impromptu conversations can normally be only 1-on-1, or a teleconference must be scheduled with adequate notice).

- Email is not "real-time" - people can answer anytime (this is really both a pro and a con, since sometimes no response is given at all). Since time differences are a factor to consider, email allows communication to take place even though engineers may not presently be in the office or at their desks.

However, some engineers are concerned about email: it is possible to lose history with email (you cannot rely on it too heavily - it can be easily deleted); there is so much email created now it is sometimes difficult to keep up and sort it out; and email can inadvertently be sent to the wrong people or to an entire list, sometimes creating havoc or security catastrophes.

A great deal of thought has gone into the tools training program and formal training that has been offered and utilized by many Program A personnel. However, several interviewees did not receive formal training on the tools, despite the established training programs. Many SE personnel that did receive training did not find the training very useful. As one engineer explains, although he did received some training on the tools he uses, he did not find the training completely sufficient to accomplish the specific tasks he performs for his job. This is unfortunate because the engineer further explained the necessity of formal tool training: 1) From on-the-job training, you pick up other people's bad habits; 2) Without training, the tools are not properly used; and 3) Without training, you are not using the tools to their full potential, meaning you may be wasting time and resources. A good alternative to formal classroom training, however, is the use and availability of online user's guides and training manuals. Several interviewees expressed their contentment with the availability and ease-of-use of these available tool supporters.

DOORS is used extensively on this program, especially by the System Requirements CPT, and has been bolstered by the "DXL" extension described above. DOORS is very flexible, and to be manageable on such a large and distributed program, its flexibility has been limited by removing access to certain DOORS features and creating standardized templates for all of the requirements documents. As mentioned, DOORS is the main tool used for requirements management and development. The main "official" DOORS database is located on a classified network at a single identified location. (The tool version and specifications in this database are the Program A "master.") It is important to note that two requirements documents are formally controlled in an unclassified DOORS database at other locations (due to accessibility issues). These two databases are synchronized once per week with the "master" and users must understand that these requirements documents in the classified database lag the unclassified master by up to 7 days.

Several tools support DOORS use and DOORS use processes. WTRT and AITS are support tools that facilitate tool and requirement management: AITS is used to manage DOORS action items; and WTRT is used to record and implement changes to requirements documents. ClearCase is used to configuration control all the DOORS DXL extension scripts employed to collect metrics and additional data from DOORS. Metrics on the DOORS requirements and test artifacts are collected by automated scripts every night (this data includes such information as the number of TBD's, the requirements traceability, the number of shalls, the number of open action items, the number of proposed requirement changes, etc.) and is consolidated monthly for review by management.

4.3.2 Identified Potential Collaboration Tool Use Issues

Although the Program A SE personnel have done a lot of up-front work to make the collaboration and development tool environment very helpful to engineers, several potential and engineer-experienced tool issues were discussed during interviews, including:

- **Different Tool Versions in Use:** Although all efforts are made to ensure that all teams and locations are using the same tool versions (by way of contracts, process documents, and work instructions), not all sites use or have access to the the same tools or versions of tools.

- **Tool Account Set-Up Delays:** To obtain accounts to collaboration tools, especially those controlled by another company, can take almost a month, and requires that an account request form be filled out by then engineer and signed by one of the program managers.

- **Classified Data Security:** Not all development facilities have access to or are accustomed to using classified data systems, which leaves SE leadership concerned. Special accommodations have been made for these locations, including frequent unclassified importations of unclassified data from the classified system. The concern here is obvious, all parties want to ensure that no classified data is accidently imported onto the unclassified system. Although established security processes exist, classified material does occasionally contaminate the unclassified system, and the system must be taken off-line for several days and de-contaminated.

- **Tool Accessibility - A Network is Necessary:** In this CDSE environment, almost all tools are only useful if one can connect to a network or the internet. Compared to traditional environments where a specific engineer may have the work he/she uses locally on their computer; in a collaborative environment such as that for Program A, most of an engineer's work resides in a database somewhere and is accessible only via network connectivity. This has lead to several interesting new issues. For example, since some travel to contractors and partners is still necessary, one cannot do any work while traveling, since networks are not available. The same is true when trying to work at a partner's or subcontractor's location. Often, it is very difficult for traveling engineers to gain access to their "host's" network - there are issues such as firewalls, proprietary agreements, and lack of a high-speed internet connection available.

Again, these issues make getting work done while traveling very difficult to accomplish. The irony: in some cases, engineers are now more productive when working distributively, than when co-located.

- **Tool Accessibility - Classified Network Scarcity:** Collaboration between classified environments is very difficult - not all sites have access or equipment to support encryption, including the customer. (Necessitating the constant exchange of unclassified and classified data through different means: courier service, United States mail, etc.)

- **Tool Accessibility - Account Necessity:** Not everyone who needs access to the tools has access, predominantly because it is so difficult to acquire account access. When all engineers who need access do not have access, the engineers that do have access are additionally burdened by having to perform data transfer and collection to/from/on the tools to assist their teammates. This is unfortunate and a waste of an experienced engineers' time.

- **Email Incompatibility:** As mentioned, Company A uses Lotus Notes as the main email program for the company. One engineer explains why this is an issue: "the rest of the working world uses Microsoft Outlook," including the Program A customer. The issue with compatibility is that address books, calendars, contacts, etc. are differently created and stored between the two software packages. Therefore, something as simple as meeting requests (calendar entries) cannot be sent via the automatic calendar functions. Instead, standard email is sent and engineers must manually update their calendars. Obviously, this is not an issue of great importance, but it does take additional time to make the different tools work together.

- **Data Transfer between Tools:** As mentioned, because not all engineers and/or locations have access to all of the tools they need, a great deal of time is spent by more senior engineers (who have access to all tools) transferring data between tools, unclassified and classified networks, and even between company internal tools and program-wide tools.

- **Network and Tool Reliability:** Network hubs and databases for Program A are distributed across the country. Unpredictable events that disrupt the tool servers at one location can affect all of the locations where the tools are used (one example event was Hurricane Katrina). Often, because of tool server locations, there are also delays at different sites for getting tools online, linked in, or updated. Additionally, some tools themselves are not always stable, and can crash often - but because of the experience gained with these issues, the tools can often be restored to functioning rather quickly.

- **Tool Process Ambiguities:** A great deal of the use of collaboration and development tools is governed by tool processes (in the SEMP, RDP, and work instructions). However, some engineers feel that the tools are sometimes used incorrectly and that there are not always sufficient references for proper tool use (example: ambiguities with how to correctly fill in requirement Change Proposals in DOORS.)

In addition to the general tool issues described above, there were some specific issues relating to DOORS and the DOORS support environment:

- **Delays with DOORS Network Connectivity:** DOORS network connectivity took a very long time - it was very difficult to get all necessary users online.

- **Out of Date Specifications:** Until the customer and other contractors were finally online, a weekly CD archive was sent to each sites - the distributed teams were therefore always working with specs that were at least one week old.

- **WTRT and DOORS Synchronization Delay:** The DOORS Support tool WTRT is not entirely integrated with DOORS - the two systems are "synched-up" once every 24 hours.

- **DOORS Legacy Issues:** DOORS was originally a British software package, later bought by an American company. To this day, there are still various issues associated with printing (due to different paper sizes in the United States and UK),

- **Baselining in DOORS Erases Specification History:** Although the history tracking features of DOORS are very useful, the entire history gets erased when the document is baselined for each customer release. Therefore, no single DOORS document has the accumulated history from the "start" of the document. This is unfortunate when issues or questions arise about the requirements, since the history for why the requirement changed may have disappeared.

4.3.3 Identified Collaboration Tool Use Success Factors

Despite several experienced and potential tool-related issues, the Program A team has learned many lessons about the collaboration and development tools and have therefore identified several successful collaboration tool practices. The following bullets summarize some of these success factors and lessons learned, as described by the SE personnel interviewed.

- **Wide Variety of Tool Training:** Several different varieties of tool training are available by means of formal classroom training, online training, and online tool user guides for most of the tools used by Program A developers. To demonstrate management commitment to the importance of tool training, "tools training" is a metric that is tracked monthly for each team on Program A. Training is always available for team members - it is up to the team leads to align personnel with work scope, experience and training needs.

- **Tailored Team Training:** Not only is tool training available, but where necessary, the tool training has been tailored to only teach the system engineers what they need to know to complete their job functions, rather than overwhelm them by all of the features and capabilities of a tool. To overcome time and budget issues associated with training, "learn at lunch" training sessions are also held, where team members buy or bring lunch to a conference room and receive training while eating.

- **Tailored Tool Use:** To assist with consistent artifacts and to make tool use easier, specific SE development tools were tailored to include only the necessary features and attributes to get the job done. This tailoring made the tools much easier to use, and allowed for more efficient development and collaboration. (Note that the tailoring of tools for specific applications was unprecedented at Company A.)

- **Up-front Work and Planning:** SE leadership explained that a key success factor for the Program A tools is that a lot of up-front time and money was invested in tool use, development, and planning. Included in the up-front work was the creation of scripts to automate many of the key processes, including: collection of key metrics; implement document changes; and perform document and action item tracking.

- **Implement Lessons Learned:** The previous Program A design phase was a significant learning experience and SE leadership is in the process of executing the lessons learned from that phase. Several recommendations for improving collaboration tools were made and the leadership team is investigating how they can be implemented. This is a difficult task as at this late stage in Product A development, as the leadership must prove that changes are of a significant cost benefit to the future of the program.

- **Tool Support Staff Provided:** Full-time DOORS and tool support staff is provided to support the SE teams.

- **Accommodate Tool/Network/Data Transfer:** As mentioned in Section 4.3.2, transfer of data between unclassified and classified networks and tools is inevitable, due to the accessibility issues on Program A for some engineers and locations. To facilitate quicker data transfer, classified and unclassified work stations have been placed side-by-side in the classified work rooms.

- **Use Standard Tools wherever Possible:** With the learning curves associated with new tool use and the sometimes hesitancy experienced by older generation engineers, leadership recommends, "Use the traditional tools everyone knows how to use (like Excel, DOORS, WTRT) and adapt them for your team's specific needs and uses," rather than introduce new tools.

- **Make Simple Tool Facilitation Investments:** Since a great deal of virtual meetings involve teleconferences, investing in hands-free telephone headsets enables engineers to type while talking on the phone, makes them more comfortable, and prevents neck and shoulder injuries.

In addition to the more general tool use lessons learned and success factors discussed above, the tools support team has learned many valuable lessons and formulated several success factors over the past several years from their experience with DOORS and the DOORS support environment, including:

- **Know What You Want/Need from the Beginning:** To assist with the DOORS development process, Telelogic, the DOORS vendor, can provide a Product Architecture Workshop (PAW) to establish schema and usage of a new or existing DOORS database. Locally, this is an on-going effort where the DOORS team will meet with the key Program personnel and determine what should be in the documents (i.e. what features need to be used in the tool).

- **Limit Tool Flexibility:** Several small teams that use DOORS have little experience and "go wild" - allowing each user complete flexibility. Allowing this type of behavior on Program A, a large complex system with over 300 requirements documents would have resulted in un-standardized, difficult to navigate requirements. Program A took advantage of the flexibility of DOORS, but limited the user flexibility - it essentially balanced the needs of the many at the expense of the few.

- **Be Consistent with Tool Changes:** Not only can the information in the tools be changed - but the tools themselves can be altered or bolstered with additional scripts and coding. It is therefore necessary to have an established set of criteria to determine what changes are allowable so that when requests come in to change the tools, they are treated fairly. The DOORS support team has created a list of "20 Questions" to determine when additional features of DOORS can be added.

- **Management Backing is Essential to Widespread Successful DOORS Use:** The introduction of the DOORS database tool was supported by the leadership on the proposal team, and DOORS was adopted very early on, allowing a great deal of automation from the start of Program A.

- **Establish Tool Guidelines:** From the very beginning of the Program, guidelines for tool use were established for the team to get started right away (Even before the SE processes were in place.).

- **Develop Tool Processes and Guidelines in Formal Documentation:** Standardization is difficult - capture processes and guidelines in formal, configuration controlled documentation (process documents and work instructions) how the tool will be used as early as possible and enforce the processes.

- **Automate where Possible:** Scripts run overnight and all weekend to collect metrics, remove user locks, update the system, search for peculiarities, check flow-down and traceability in requirements, archive the databases, synchronize the systems and other housekeeping functions that are necessary to maintaining a "healthy" tool environment.

4.4 SE Knowledge, Data and Decision Management Practices

Upon completion of all interviews and data review at Company A, it was abundantly clear that SE and overall Program A knowledge and data management is in general facilitated by collaboration tools. This section discusses how tools are used to support knowledge and data practices, in addition to other data management techniques on

Program A. Also included in this section is a review of the issues and success factors associated with knowledge, data, and decision management.

4.4.1 Collaborative SE Knowledge and Data Management Overview

The SEMP defines a data management plan, calling out specifically the tool and level of management (informal, formal) for design artifacts, deliverables, SE processes, decisions, organizational information, and meeting records. All data that is formally managed is under the control of Program A configuration management processes. As expected, all interviewees report that knowledge and data management on Program A is facilitated and organized by the collaboration and development tools used, including:

- **Teamcenter Enterprise:** For document, artifact, and contract formal configuration control.
- **Teamcenter Community:** For data repository, informal configuration control of documentation.
- **DOORS:** For formal requirements management and requirements configuration control.
- **WTRT:** For formal action item management and decision making records.
- **Email:** For informal knowledge sharing and dissemination, as well as for transfer of program artifacts via attachments.
- **ClearCase:** For formal software and system model version control.
- **Engineering Model Tool (EMT):** Tool to manage technical product, including mission scenarios, system interfaces, and subsystem relationships.
- **Microsoft Office Suite:** For informal development of presentations, documents, white papers, action item spreadsheets, and meeting minutes.

- **Classified and Unclassified Networks and Shared Databases:** Used to pass and share information - most users have "User Folders" where the information they personally work on and share with their team members is located.

- **Telephone and Teleconferences:** Used to share and disseminate knowledge and also aid in decision making.

The tools described above provide knowledge, data and configuration management for: decisions, design artifacts, deliverables, meeting minutes, design rationales, requirements traceability, contractual documentation, procedural documents (such as the SEMP and RDP) and other important Program A organizational, procedural and technical information. In addition to the tools, knowledge, data and decisions are also managed via the formal deliverables, such as design reviews, specifications, peer reviews, and procedural documents.

Email is another tool that is used to capture knowledge and data. As discussed in Section 4.3.1, email is a preferred method of communication of many engineers and can be used informally to share information, knowledge, decisions, and facilitate widespread communication within and between all Program A teams. Other widely available tools such as Microsoft Excel are used at an informal level to record meetings and track progress. One engineer describes how he uses Excel to manage his team meetings: Excel spreadsheets are updated in real-time during meetings to track the open issues, the agreed to actions, the agreed to dates to complete the actions, and the history of the issue. These spreadsheets are updated every week, and as actions are closed they are removed from the list. The engineer also keeps an organized list of all of the past weekly meeting spreadsheets for documentation of the history of completed items.

Knowledge is in general disseminated via the tools and formal deliverables, as well as at team meetings, by sending team emails, and via general "word-of-mouth" discussions between engineers.

Most engineers reported that they did not have trouble finding the information they needed when they needed it, but that navigating through the databases and collaboration tools can sometimes be difficult. Because of the difficulty in navigating some of the databases, some engineers recommend that training may be useful in locating the information needed when it is needed. Many engineers did relate that it was difficult in general to find artifacts in Teamcenter Enterprise, since the fastest way to find something is to have a 12-digit document number, and that no "good" search tool exists to navigate the wealth of information in the database.

The tools themselves also provide knowledge and data management, as well as configuration control assistance. For example, DOORS and WTRT have well-defined user and administrator access, so data such as requirement or action item change history and account log-ons can be tracked and used for later accountability. WTRT is also configured to capture the decisions of key stakeholders when it comes to requirements or design changes, also allowing accountability and widespread dissemination of decisions. An additional feature of some of these tools is automated notification (via email) to key stakeholders when changes are made.

It is obvious that the tools and the processes are clearly used to manage the requirements knowledge and data. As issues arise and lessons are learned, the natural consequence of these issues is to update the tools. Therefore, the final product itself (the DOORS or WTRT databases) reflects the changes. For example, the DOORS attributes (the attributes that define each requirement) are standardized (by a work instruction), and as requests to update or change the attributes are made, the tool (DOORS) is changed or updated (so is the work instruction). Further, to manage the interface requirements in DOORS, one member of the System Requirements CPT and his team have added additional DOORS attributes that capture the necessary information for each interface and allow for consistency between interface documents. DOORS' scripts automatically create the correct links for requirements traceability and generate the DOORS interface requirement wording to be consistent on both "ends" of the interface.

4.4.2 Collaborative SE Decision Making Overview

One engineer explained: "Decision making is governed by the processes - on a program as large and complex as Program A, the Work Instructions, SEMP, and RDP are the gospel." Decisions are made in accordance with the processes and directions outlined in key program and SE documents. The SEMP explains that technical decision are to be made at the lowest level possible within a team. Issues that cut across multiple teams and/or that span multiple budget areas and/or issues that cut across multiple contractors or must be brought up to the overall program leadership.

Decisions about system requirements are made predominantly in the weekly RCRB meetings by following the RCRB and configuration management process. Requirements changes and the final decisions are officially documented in the WTRT tool. All decisions are pretty much made together as a team - with key stakeholders weighing in and final approval being given by the customer. There is no defined process for informal decision making; however one System Requirements CPT member explained how he keeps a record of the informal decisions made during weekly working group meetings in Excel spreadsheets and then later emails them to the team.

Decision making is an issue for the program (in fact, decision making was an issue called out in their recent CMMI - Capability Maturity Model Integration - qualification). Decision Manager Database is currently being used to record decisions - but widespread adoption and utilization is an issue. Unless everyone is using the tool, it is of limited effectiveness.

Many engineers related that their decision making is impacted by collaboration. Specifically, it is difficult for them to make decisions, since to make decisions stick they need to be formally documented in a deliverable or contract. There is currently a Company A process improvement project being performed examining the hierarchy of decision making on Program A. To compensate for distributed collaboration, some engineers even build decision making inefficiencies into their schedule, specifically the time it takes for multiple companies and the customer to review documents for formal reviews. Several engineers report that they use a bit more caution when making decisions, as Program A is large and so is the amount of data available - therefore engineers are thorough and careful before agreeing to decisions.

Most formal decisions regarding the program (specifically those made at the program level) are typically distributed via email or disseminated via word-of-mouth in weekly team meetings or at the engineer-to-engineer level.

4.4.3 Identified Potential Collaborative SE Knowledge, Data and Decision Management Issues

Although almost all SE personnel report that they believe the knowledge and data are well managed on Program A (especially given the program size), there are several issues that have arisen for knowledge, data and decision management, predominantly due to the size and distributed nature of Program A. These issues are:

- **Not Enough Time to Thoroughly Document:** Although knowledge, data, and decisions are fairly well documented, there is not really enough time to properly document all issues and decisions. A specific example: the teams tend to document what was done, like choosing Approach B, but often do not document why it wasn't done a different way, say Approach A. So later, when one wants to try approach A, there is no documentation explaining why Approach A shouldn't be chosen because it failed previously. Sometimes, one team member explains, the "why not's" are more important than the "why's."

- **Peer Reviews are Difficult to Organize:** Getting widespread peer reviews is an issue on Program A, as they need the customer technical team, which is often very slow to respond to requests to review material and attend reviews.

- **Classified Data Sharing:** The free flow of data, information, and knowledge between companies and locations is impeded by data classification. When it comes to classified data sharing, the issues are obvious - the necessary networks, classified collaboration tools, encryption devices, classified telephones, security personnel, etc. are necessary for classified information to flow freely between cleared personnel. Because not all of those tools and channels of communication are in place at all development sites, classified information can often be delayed (for example, since it must be sent by courier) or not distributed at all.

- **Company Proprietary Data Sharing:** The free flow of data, information, and especially knowledge, between companies and locations is also impeded by company proprietary data. Because competitors are working together collaboratively on SE and product development and integration, alleged company proprietary data must be shared in order to understand the system design, interfaces, and integration. Company proprietary data sharing agreements are in place, however there is often tensions and disagreements over what constitutes company proprietary information, who has access to view the proprietary information, and what level of information can and will be shared.

- **Email Can Disappear:** Unfortunately, email is deleted off the Company A internal server within 30 days, and unless you know how to set-up mail archiving (apparently, it is not a simple process), it is possible to lose your email. Therefore, if email is not backed up properly (archived), it is difficult to recall for further reference, and there is no history of decisions, data, or communications.

- **Non-Uniform Specification Development and Configuration Control Enforcement:** Toward the beginning of the program, Company A management permitted two specifications to be configuration controlled and developed entirely within a contractor's internal database. This DOORS process variation has created several issues for some members of the System Requirements CPT: 1)The contracting organization is not as cognizant of RCRB and WTRT processes; 2) The lack of process following creates additional work for the team members, who must follow-up with the contractor very often to ensure that their specification is updated as specified; and 3) The "master" DOORS database becomes dependent on the contractor database for these two elements - it only gets synchronized one time per week (creating synchronization issues and delays). One must always be careful to ensure they are working with the most current requirements documents.

- **Tools and Processes Needed for Meeting Minutes:** There are currently no easy-to-use, readily available tools to document meetings, meeting minutes, and actions for virtual meetings. Different teams collect and distribute meeting minutes in various formats, and often do not capture all of the necessary information or key actions. A tool and/or process to record meeting minutes could make team meetings run more smoothly and ensure all necessary information is captured and dispersed to the right team members.

- **Recording Meeting Attendance:** Part of the difficulty of assigning and following-up on actions, is not knowing who is attending each meeting. When taking and recording attendance at virtual meetings, with so many sites calling in and people able to arrive late or leave meetings early without being noticed, it is difficult to maintain an accurate list of attendees. Attendance is necessary for accountability. Many engineers have tried to take attendance at the start of meetings, however it wastes precious time to take a roll-call. As time is critical, people even intentionally show up late to miss the introductions. Some team leaders have requested people email them if they attended the meeting, however some still do not, and the team leader must waste time reading emails and compiling a list of attendees. Establishing or creating tools or formal processes to automatically record meeting attendance could be of great benefit to the Program A team. (Note that Teamcenter Community does keep a list of those signed on, and it even allows you to copy and paste the names of those logged-on, which is a step in the right direction.)

- **Collaboration Tool Navigation:** As mentioned, Teamcenter Enterprise is the tool used to formally archive and configuration control materials and data for Program A, but it is very difficult to navigate. If you don't know the unique 12-character code for a specific document, much time could be wasted searching the database. Also, Teamcenter has a very non-intuitive user interface, making it even more difficult to use. Therefore a lot of teams do not even use Teamcenter for version control, as many people do not know how to use it and are not normally trained on how to use it. This issue is complicated by the difficulty in obtaining account access.

- **Distributed Decision Making:** This is in general a Program A issue, predominantly due to the size of the program and the system complexity. To record decisions, it was mentioned that a tool called Decision Manager Database was adopted; however not everyone used it, and since it was not up to date, it became even less useful and less utilized. It therefore has limited effectiveness.

- **Need Better Database Search Tools:** In both Teamcenter Community and the shared classified network databases, any user with an account can post, move, or delete anything and the amount of information available increases daily. One engineer described that it is pretty much impossible to use the standard "search methods" to search the wealth of data available about Program A. A simple "Ctrl-F" can result in hours of search time, and may never find the information you are looking for, or worse yet, may result in using an obsolete version of a file! Note that one of the shared databases (at the time of the interview) contained over 2.54 terabytes of Program A data. In general, real search capabilities are non-existent in all of the tools and databases used on Program A.

- **Missing Decision and Design Rationale:** An interesting aspect of Program A was that a physical baseline was needed before the functional system had been well thought-out, due to politics and the need to demonstrate to the government the system feasibility. Because of the pressing need to complete the physical baseline, a great deal of concurrent physical and functional architecting was done, and unfortunately, not all of the design decisions and rationales for decisions were well documented. These issues are being confronted now...

- **Dissemination of Information:** There are many different Program A teams and locations, and rightly so, not all information is sent to all personnel. However, since it is so difficult to locate the information you need when you need it, disseminating information is a key issue in this type of environment. Unfortunately, due to the size of the program and the distributed nature of teams, it is difficult to spread news, updates, changes, and decisions to all personnel at all locations in a timely manner.

4.4.4 Identified Collaborative SE Knowledge, Data and Decision Management Success Factors

As mentioned early, Company A relies heavily on process improvement initiatives to develop ways to improve current issues. At first, the System Requirements CPT had a great deal of trouble managing data and making decisions, including locating the information they needed when they needed it and providing accountability. Information was discrete and located in dispersed databases. Relationships were not well-defined and all the key players were not able to weigh-in on decisions. Several of these issues were remedied when several key team members participated in a process improvement project that looked into ways to get all of the information to support the team organized. This specific process improvement project team is very similar in nature to the very successful "voluntary learning teams" used by Toyota. (Refer to Section 2.6.2.1.) Several very important and successful methods/processes were developed by the System Requirements CPT to manage knowledge/data and make decisions (Note that all of these factors are lessons learned.), including:

- Use Teamcenter Enterprise (specifically the WTRT tool) as a central location to store and share data.

- Record and share RCRB meeting minutes to document agreed upon changes.

- Keep track of and publish attendance, ensuring decision accountability and awareness for a decision is traceable.

- Allow each team member to be heard/weigh-in on decisions so their opinion is known and recorded - but only those with authority actually have bearing on decisions.

- Appoint SMEs and key personnel for each subsystem/area - hold those people accountable for sharing information with their teams and also for weighing in on decisions/changes that effect or impact their product.

- Meetings will only begin when ALL key stakeholders/SMEs are present and accounted for - creates peer pressure for not wasting your colleagues time.

- The customer must agree to changes as well.

- In order to use SME time efficiently, most of the work for review meetings must be done off-line (outside of the formal meeting time). Formal meetings are efficiently run, and there purely to formally record decisions, positions, and data.

- Create additional attributes in DOORS to provide additional data and history: such as traceability to requirements flow-down, changes, design rationale, and change rationale.

- Tailor the data and metrics collected by tools (such as DOORS and WTRT) to see the important trends, new requests, changes, and status.

- Use the accessibility feature of tools to limit those that can edit/update/make changes to requirements documents and to WTRT records. This feature also provides an accountability mechanism and tracks decisions.

- Use the processes to your advantage. For example, the process to update an issue in WTRT includes mechanisms for managing and documenting changes and decisions regarding requirements. There are specific "codes" in WTRT that document the stage of the process, the decisions and agreements made, and decision makers' perspectives.

- Create processes with input from all management, therefore creating management buy-in and accountability. For example, the RDP was created and agreed to by SE management and leadership - and therefore all team members are accountable to follow the process.

- Define the relationships between subteams and have an arbitrator/coordinator to work out the issues/overlap between each team and ensure that the right people are interfacing.

- Define a team glossary/dictionary to foster uniform interpretation and to minimize misunderstandings.

Additional success factors and lessons learned have been developed since the last knowledge, data, and decision management process improvement initiative, including:

- Keep track of and share useful document locations with teams, since the tools are not easily navigable. One engineer therefore includes a list of document location links in his weekly meeting minutes, for use by both himself and the team members he writes to.

- Update tools to correct known data management issues. For example, one drawback of the WTRT system was the ability to lose track of the WTRT reference documents. The most recent tool release of the WTRT system has allowed attachments to be "attached" which has improved accessibility to the needed references significantly. The only downside of the file attachment in WTRT is that anyone can currently delete the file.

- Enhance the development environment to support knowledge sharing and creation. To facilitate development at one of the major System Requirements CPT work sites, the classified computer laboratories have "individual" and "teamwork" workstation set-ups. When group work is needed, one side of the room is set-up with open tables where workstations are within close proximity to facilitate team-work, knowledge, and data sharing. The other half of the room is set-up with individual work stations sparsely located and separated by cubicles. This laboratory set-up is a great example of flexibility and variety so that different types (group and/or individual) of work can be completed simultaneously.

- Use processes to support decisions, and knowledge and data management. Data and documentation explosion is an issue on a program of this size and complexity. Therefore the requirements specification tree has been used as a decision making tool: all new requirements documents must first be on the "spec tree", which itself is a configuration controlled DOORS module. All requests by engineers to add a specification that is not on the defined "spec tree" are denied. Therefore management must agree that the additional specification is absolutely necessary by adding it to the tree before development begins. Requirements management, development, and upkeep is costly.

- Document the lessons learned about tools and processes for use on future programs. There is no need to keep "reinventing the wheel." After all of the hard work by the Program A tool support team to create scripts, automate maintenance and metric collection, and create tool processes and standards, Company A has recognized this success and has devoted dedicated resources to capturing all of the tools and the processes for use on other Company A programs.

At the time interviews were conducted at Program A, another process improvement initiative was underway at Company A to address many of the additional knowledge, data and decision management issues discussed in Section 4.4.3.

4.5 Collaborative SE Practices and Processes as they relate to Product Development

Although discovering how the SE practices and processes as they relate to the development of Product A is one of the key areas of interest of this research, many of the SE processes, methods, and technical artifacts are considered either Company A proprietary, are classified, or are export controlled and therefore their sharing is restricted. Consequently, this section discusses at a high-level the general SE practices and processes as they relate to Program A development.

4.5.1 Collaborative SE Practices and Processes Overview

The Program A SE practices and processes are documented in the SEMP, the RDP, and other Program A work instructions. Included in the SEMP are SE processes for system requirements analysis, functional system analysis, architecture and design analysis, cost analysis, system verification and validation, risk-handling, trade-off analysis, and many other detailed SE activities. A separate process also exists for the System Requirements CPT RCRB.

All development sites and companies are under contractual obligation to comply with the SEMP and the RDP. Because all SE processes are mandated or referenced by the SEMP, all locations use the same SE processes and procedures. Therefore there is a high-level of SE consistency between different sites and companies.

Program A is unique in its establishment of "mission system design," which is essentially a mission thread based object-oriented approach that outlines and coordinates all of the necessary hardware, software, firmware and user tasks that comprise end-to-end mission system requirements. This non-traditional process for development was born out of a one of the critical system requirements. Therefore, to prove to the customer that Product A would meet this critical requirement, every aspect of a mission had to be modeled in real-time. The mission scenarios being modeled cut across multiple systems of Product A, and thus multiple development sites and companies. Consequently, the mission system design has helped to create a better integrated product.

Another Program A innovation, created to assist with the mission system design approach, was the Engineering Model Tool (EMT), which integrates the system architectural layout and is used to decompose the mission threads into system interactions and activities. Last, the EMT facilitates requirements analysis and a modeling environment to help with the development, analysis and verification of system and software requirements.

The product A system architecture is largely influenced by the Program A team organization. The program is organized based on areas of expertise and contracts; thus so is the system. One SE leader hypothesizes that the architecture would likely be very different if the program was "in house" or of smaller magnitude - for example, more common services would be utilized (as opposed to the distributed services used currently on Program A).

The interviewees described some SE process and practice differences on Program A compared to some of the traditional SE programs they have worked on. One SE leader explained that the product and requirements have been modified to accommodate collaboration, specifically in regards to the interfaces. For example, interface requirements must be defined first to set the boundaries of the subsystems and define the necessary relationships between developers. Another example is the way in which the system models and designs are shared, built and incorporated had to be changed. Sharing, building and incorporating system models involves having to "check out" models and send them across different classification domains to other companies.

Additionally, differences between companies in requirements philosophies have made requirements development more difficult - specifically regarding "what is a good requirement." As one engineer points out, when companies have been making the same product for a while, they tend to start imposing designs in their specification, rather than defining performance requirements. (Example: Company Y always uses wire X, they then require use of wire X. In reality, any wire that allows n Amps and m Volts meets the performance requirements.) This philosophy, in some sense, compromises the integrity of the requirements, as a design should not be imposed by the requirements allocators. To help overcome this problem, "Good Requirements" guidelines were developed to help teams write testable, achievable, performance requirements.

4.5.2 Collaborative SE Final Product Integrity

All interviewees were asked how they believed the final product integrity would be impacted by CDSE, and the answers the SE leaders and engineers provided varied greatly.

After discussing many of the issues involved with CDSE, I was pleasantly surprised to hear that several engineers believe the final product integrity has been positively affected by the CDSE efforts. Engineers and leaders related that the positive benefit is largely due to the fact that Program A has taken advantage of the national industry, and together with their expertise they have created a final product that would have been otherwise unrealizable by a single company alone.

A few engineers believe that the final product integrity is unaffected by CDSE, specifically due to the strict development processes in place.

Unfortunately, other engineers feel the integrity of the final product has been negatively affected by CDSE. One reason for this was the belief that due to distribution and proprietary data sharing issues, the systems engineers cannot see what everyone else is doing/designing/building, and therefore system integration may be more difficult and result in a less integrated final product. Another engineer felt that because different companies likely have different priorities, they may inadvertently be doing what is best for the future of their respective company, but not best for the final product.

4.5.3 Identified Collaborative SE Practices and Processes Lessons Learned and Success Factors

As this is a little-explored area of research, the lessons learned and success factors relative to CDSE processes and practices are very important. The following list summarizes the key lessons learned and success factors related from the Program A interviewees:

- **Coordinate Development Efforts with formal SE Processes:** Well-defined, formal, configuration controlled SE and development processes are in place to coordinate product development and facilitate development, product, and artifact consistency.

- **Perform Process Improvement Programs** Process improvements are undergone and utilized very often and are a key aspect on Program A. Process improvement programs have been utilized extensively to save money on design and manufacturing, solve problems, and reduce efforts (such as program staffing needs). The RCRB process, stakeholder identification, design and build process, tool tailoring activities, and others endeavors are all the result of process improvement initiatives programs. The team has a dedicated process improvement "Expert" to help direct programs and ensure their successful completion.

- **Define a process to raise SE issues for processes, designs, changes, etc.** One lesson learned by Program A was the need for a standard process to raise issues concerning SE processes, design flaws, and design changes.

- **Define a process to decide how system changes get flowed down to subsystems or contractors.** Another lesson learned by Program A was the need for a standard process to determine how system changes propagate through the subsystems and contractors. Especially when product development is done distributedly across multiple companies via formal contracts - changes are difficult to manage. Therefore, in advance of an avalanche of system changes, a process or clause should be put in place to determine how subsystems and contractors will be affected by system-level changes.

- **Define a process to request assistance, from either the customer, contractors, and even within individual team structures.** One aspect of collaborative, distributed teamwork that is often overlooked is how to ask questions and receive assistance from your teammates, especially when work involves formal contracts and competing teams. Thus, Program A also learned that with a distributed team structure, a process for receiving assistance at all levels of teamwork (intra-team, inter-team, contractor, customer, etc.) is necessary.

- **Update the Tools to Support Processes and System Architecture:** For example, the Program A system is so large, and there are so many interfaces, that to coordinate the interfaces, each interface in the system was given a unique identifier and description in separate DOORS modules. To ensure that each interface was being properly and adequately required in the system, additional DOORS scripts were written to ensure that each defined interface has 2 allocated requirements (1 for each "end" of the interface). These modules and linkages are consistent throughout the entire system and can be used to resolve discrepancies and ensure completeness. Additional DOORS scripts also monitor the linkages between higher level and lower level requirements (to check for requirements traceability).

- **Use Integrated Modeling and Design Approaches:** The mission thread design and EMT approach coordinates all aspects of a mission to help produce a better integrated product, verify system requirements, and model all aspects of critical system functioning.

4.6 Social and Cultural Collaboration Experience

A variety of different social and cultural topics were discussed with the Program A interviewees. The following paragraphs summarize the findings from the topics discussed, as well as some of the issues and success factors that were identified.

4.6.1 Social and Cultural Collaboration Experience Overview

The following paragraphs summarize the responses on some of the key social and cultural topics discussed:

- **Job Satisfaction:** All systems engineers, SE leaders, and support staff interviewed expressed that they really enjoy working in the Program A CDSE environment, despite the additional challenges introduced by distribution and collaboration. SE leadership did express their awareness that there are some systems engineers who are dissatisfied working in such a complex, fast-paced environment.

- **Team Distribution:** The number of immediate team members co-located with the interviewee ranged from being the only team member on-site, to most of an immediate team being co-located. Due to the nature of the requirements development work and the fact that all systems must interface, almost all development requires that team members interface with engineers located off-site and/or within a different company.

- **Face-to-Face Meetings:** At peer reviews, formal design reviews, and sometimes for general meetings, many of the engineers and leaders have met face-to-face with the engineers they work with distributively . One engineer felt strongly that meeting people face-to-face allows for the establishment of future contacts. It is difficult to develop a lasting relationship with someone you only speak with over the phone or email - the face-to-face contact allows for recognition and a greater comfort level when you need to contact the same people again. Some engineers just feel more comfortable when working face-to-face with other engineers.

- **SE Team Leadership:** SE leaders expressed that the priorities for leading in this type of environment are slightly different than in a traditional environment.

For example, metrics, plans, and accumulated data must be elevated to upper management to keep the disagreements between companies in check. Management of those engineers that are not co-located is difficult, and leadership relies more heavily on the metrics collected to tell if there are "problems." If a team is not performing, or their metrics change drastically, this occurrence is a prompt for leadership to investigate the reasons behind the performance change.

- **Language and Cultural Experiences:** Although many engineers have encountered some language (interpretation) and cultural issues, some engineers really enjoy the exposure to different languages (i.e. United States regional accents) and cultures. One engineer believes that it is very eye opening experience for those who may not have left the Northeast to experience other cultures remotely via email/phone.

- **Trust:** Despite the fiercely competitive environment when working with your competitors, the overall attitude on the program is "one team." Most engineers expressed that they have no issues trusting their counterparts or team members at other companies and locations. However, trust takes time to build and can be easily shattered. One engineer summarized his trust philosophy as: "assume the best until that trust is violated."

  In an environment where competitors are collaborators, it is very difficult to begin with established trust. One obstacle overcome by the tool support team was the collection of collection of metrics and the information being relayed to the customer. Every team wants to have their own set of data to check and double check that the metrics are correct. In such a large, dynamic environment like that of Program A, it is very unlikely that the metrics collected one moment are exactly the same as at any other. Obviously, the metrics must be credible, or else every team would be fighting to finesse the data in their favor. Trust of the Company A metrics collection process was a key issue. Company A had to convince the other contractors that only one set of metric data should be collected to maintain credibility, and also that their metrics collection was correct and error-free. In order to convince other contractors, Company A held a full review of the scripts used to collect the metric data. These scripts are now configuration controlled and independent verifiers from other companies must agree to all changes.

- **Collaboration Environment Training:** There is some collaboration training offered to team members on working in a collaborative environment, including all-hands meetings, process days, and skip-level lunches. However, most teams are not specifically trained on working in a distributed/collaborative environment. One leader recalls that at the Program A start-up several years ago, the leadership teams did actually attend team-building training sessions.

- **Line Management and Career Development:** All respondents replied that line management and career development has not been a concern or an issue.

- **Collaboration Tool Use Comfort:** Most engineers stated they were comfortable using the SE collaboration and development tools. SE leadership explained that they tried to make the teams comfortable by: limiting the number of tools they must use, tailoring training courses to lessen the tool learning curve, and having tool specialists on the team to assist team members. However, several engineers were not as comfortable as they would like to be with the tools and believe that additional training may help if they only had the time to attend it.

4.6.2 Potential Social and Cultural Collaboration Experience Issues

Despite the fact that all SE personnel interviewed are satisfied working in the current distributed, collaborative work environment, there are some cultural and social issues that they have experienced. The following bullets summarize many of the issues discussed:

- **Company/Industry Tensions:** Some tensions exist between competing companies in the same industrial speciality, as well as tensions between different engineering and manufacturing specialities, where things are sometimes "just done differently" and "it is hard to see eye-to-eye."

- **Language and Interpretation Issues:** Another issue that surfaced from the differences between company and industry norms has been language or interpretation misunderstandings. In one company "ABC" means this, and in another company or industry, it has an entirely different meaning. One engineer relates a huge issue that took months to surface and correct, when a misunderstanding ensued over a requirement interpretation. That specification was written by Company A, who often worked with performance characteristic "XYZ". They required an "XYZ" threshold in the specification for a subsystem to be developed by another company (Company Z). Company Z had a different internal definition of "XYZ" and the misunderstanding was not discovered until after the Critical Design Review.

- **Company Philosophy Differences:** There are also notable differences in the way other collaborators, even the customer, does business, sometimes making it very difficult for the engineers and support staff to sometimes do their job.

- **Cultural Differences:** Other issues include pure cultural differences between different geographical regions (north/south). Some engineers mention that there is a "Yankee" mentality in New England and a"Good Ol' Boy" mentality in some Southern development locations. Although this issue is not really being addressed openly by the teams, based on the experiences of the engineers, they believe it is clear that life is just viewed differently by these two geographic regions.

- **Proprietary Data Sharing and Trust:** Trust was discussed as an issue in the context of proprietary data sharing. Sometimes, since information cannot be shared between all team members due to proprietary data sharing agreements between companies, it is difficult to trust the final results produced by a company if the means by which those results were reached are proprietary, and cannot be shared.

- **Environmental and Interaction Differences:** There are also "some things you just can't do" in a CDSE, or even just a pure distributed environment, in general. These "things" include: walking down the hall to ask someone a question, using "internal" company address books to locate co-workers, scheduling an impromptu meeting, etc. Instead, meetings must be scheduled way in advance, a lot of discussions must take place only over the phone, and meetings are very structured, formal and must include an agenda. In essence, it is more difficult to form and build relationships. One engineer recalls the "virtual" good-bye of a distributed team member. He relates that it is sometimes strange to say goodbye to a person whom you've gotten to know well, but never "met." As this engineer explained: "What do you do? You can't shake this person's hand or attend their farewell party."

- **Relationship Building:** Relationship building itself has been brought up as being more difficult in this CDSE environment, which is a big issue since relationship building is necessary to establish trust in your co-workers. Some engineers relate that it still feels better and more practical to work in a traditional face-to-face environment. It is difficult in a CDSE environment to hear and interpret (accents, quality of voice connections, etc.) conversations and also to visualize discussions and reactions on a telephone/teleconference.

- **More Formal Social Environment:** Many engineers additionally feel that the social environment is a bit more formal than the environment in a more traditional, co-located SE program. Most meetings have strict agendas and timelines and what used to be a relaxed atmosphere where co-workers would get coffee and discuss the previous evenings sports games before meetings, has developed into a much more strenuous work environment.

- **Social Isolation:** Sometimes engineers are isolated, being the only immediate team member working at a development location. These engineers tend to reply more heavily on co-workers from previous programs or those that are seated near them to interact with socially (have coffee, breakfast, lunch, etc.) . Obviously, social interaction at work is a necessary aspect for happy employees. One employee really misses her previous more traditional work environment, where she used to enjoy getting breakfast, lunch, dinner or drinks after work with fellow teammates.

4.6.3 Social and Cultural Collaboration Experience Success Factors

Although the social and cultural environment is not sometimes considered as important as the many other issues on a large program, the Company A and Program A leadership teams have developed some successful practices to improve social and cultural-related interactions on the Program A team, including:

- **Create Team Glossary and Acronym Dictionary:** As mentioned, there is a wealth of different cultures, companies, educational backgrounds, and industries that comprise Program A development, and interpretation issues abound. To help get all team members on the same page, a team glossary and acronym index was developed to minimize misunderstandings. The glossary and acronym dictionary is available on the program web-site and other collaboration tools.

- **Create and Enforce "Team Rules":** From the beginning of the program, Program A leadership created the "Team Rules" - which are the guidelines all team members must abide by concerning the treatment of all companies and teammates on Program A. Copies of the team rules have been given out to many engineers and team members in a "business-card-like" form, and one engineer even carries a copy of these rules with him at all times. He can recall meetings where the "Rules" have been recited when tempers have risen.

- **Provide Team Paraphernalia:** To foster team camaraderie and identity, Program A leadership has also sponsored the buying and distribution of Program

A logo hats, mugs, T-shirts, posters, pens, pins, mouse pads, etc. some bearing all company logos.

4.7 CDSE Benefits and Motivation

All interviewees were asked what they believed were the benefits of and the motivations behind the CDSE efforts. The following bullets summarize their responses to what they believe are the CDSE benefits:

- Less Travel.
- Sharing of the defense industry's national expertise.
- Get to work with customer as a partner.
- Exposure to a broad range of information and personnel. The increased exposure leads to a great deal of interactions (different processes, ideas, practices, cultures, etc.), which overall leads to improvements in SE.
- Get to experience diversity in many things: companies, cultures, people, ideas, etc.
- Availability to the breadth of ideas from different companies, industries and regions - the program can't help but reap the benefits.
- Different industrial and experiential backgrounds allows the program to take advantage of the national industry.
- Technology enables engineers to not have to travel, allowing the engineers to save time and remain with their families. It also saves the company and the program time and money.
- By having the collaboration tools in place, impromptu or emergency meetings can be called on short notice; whereas in a traditional environment, an entire day of travel may have been needed to have a face-to-face meeting with the customer, etc.

- Ability to have a challenging and rewarding job position, which might not be available if work was localized.

- Additional information you can gain from the engineers at other locations and/or with different disciplinary backgrounds. Having everyone come together for the weekly meetings opens your circle of contacts and you may ask a question and be very surprised by the additional information you can find out. For example, one engineer was asked to investigate an issue, he mentioned the issue at his working group meeting, and was pleasantly surprised to find it was already being worked on at another location. If would have likely taken him days to track down the right people to find that out.

- This environment forces us to enforce the processes, standards, and documents.

- There is a greater level of predictability (in people and products), since the processes are well-followed.

- We are utilizing the "hot beds of expertise" - in essence we are exploiting the expertise that is resident throughout the United States. Co-location would not be able to this!

Many of the motivations for the choice of CDSE environment can be summarized by the benefits described above. However, some additional responses from interviewees include:

- The motivation is very political and is largely based on the customer's desires.

- Customer satisfaction: with collaboration, a better system at the best value can be created.

- By increasing capabilities and experience in this type of defense system, Company A hopes to achieve future business with this customer.

- CDSE is a smarter way to do business, allowing programs to be bigger, better and easier to complete. This method is less expensive and the technology that results from it is a huge benefit. Since more work can be done, these types of programs are likely more profitable as well.

- The job gets done to the maximum product capability, since the product development utilizes the best industry resources available.

Chapter 5

Case Study 2: Company B

This chapter summarizes the raw data collected from Company B. The chapter includes a company and program overview; a description of the current SE collaboration situation; a summary of the tools currently in use by Company B; an overview of Company B's knowledge, data, and decision making practices; a summary of the current collaborative SE practices and processes; the social and cultural experiences of interviewees; and a summary of the CDSE benefits and motivations as related by the Company B interviewees.

Please note the following concerning the material presented in this chapter:

- The personnel interviewed for this case study consists of: systems engineers, or engineers performing system-engineering related activities (they may belong to organizations other than SE); SE support staff, such as tool administrators or process experts; and SE or program management. For lack of a better term, the interviewees are collectively referred to in this report as, "SE personnel".

- As described in Section 2.1, the term "SE" in this research encompasses many of the overall product development related activities, including requirements development, system conceptualization and definition, system integration and qualification, and system simulation. Although many people or organizations have different definitions for SE, the term "SE team" is used to refer to all personnel involved with the aforementioned system activities.

- Many of the SE factors discussed with interviewees are very closely related. Although all efforts have been made to separate interviewee comments into the topic areas they are most closely related to, there is some repetition between sections. (Some interviewee inputs are relative to many of the factors being discussed.)

- The identified issues, success factors, recommendations, benefits, and motivations summarized in this section are based on the data provided by a relatively small sample of Program B SE personnel, and are not meant to be declarative of program-wide applications (i.e. issues, lessons learned, motivations, etc.). However, the information discussed herein was provided by interviewees working in Company B, and on Program B, and is therefore an indication of Company B and Program B applications (i.e. issues, lessons learned, motivations, etc.).

5.1 Company and Program Overview

Company B is organized into five major business areas, with company headquarters located in New England. A program in Company B's "strategic" business area sponsored this case study. Company B employs over 1,000 personnel and has a total of five satellite office locations spread across the United States. Although there is a great deal of information publicly available about Company B and Program B, these details have been omitted to protect the company's identity and program-related technical data.

Company B is currently acting as the prime contractor and systems integrator for Program B. Program B involves the design and implementation of an upgraded subsystem to extend the lifetime of an existing United States defense system. In total, Program B employs over 350 personnel from Company B. The development of Product B is currently in the detailed design phase with a critical design review (CDR) scheduled for late 2007. The total projected product development lifecycle of Product B is approximately ten years. For this phase of Product B's development, there are six major United States aerospace and defense contractors collaborating across the United States (including Company B). To complete the overall Product B development, contractors, subcontractors, and suppliers, are currently collaborating across more than five states and the UK. For the CDSE efforts discussed in this research, collaboration takes place between companies located in Massachusetts, California, Florida, Indiana, Minnesota and in the UK.

Product B is unique in that it is an upgrade to an existing system, and therefore the Product B design must conform to the existing physical structure and interfaces, while using new technologies. The original defense system was constructed in the early 1980's, and since then the specific technology needed for Product B has come a long way. The Program B development team works not only with the final product customer, the United States government, but also with the developer of the overall defense system in which Product B resides. For this case study, the examination and analysis focuses only on the Product B SE team.

Company B is matrix organized, with each employee belonging in general to a discipline (such as SE) and a Program (such as Program B). The Program B SE teams are generally organized into several element teams, each with an overarching Program Manager (PM) and Technical Director (TD). Some of the SE element teams interviewed for this case study include: Systems Engineering, Sub-System Design, and System Integration. The element teams are further divided into tasks, each with a Task Leader. Some tasks organizations that participated in this case study include Requirements, Interfaces, and Virtual Systems. Additional SE support engineers and leaders that participated in this study were from the the Logistics and Information component team. For the most part, Company B manages the large program areas (the elements), but some of the task teams are led by contracting companies. Figure 5-1 summarizes the general organization of Program B, constructed from Program B interviews. (Note, Figure 5-1 does not contain the complete Program B organization.)

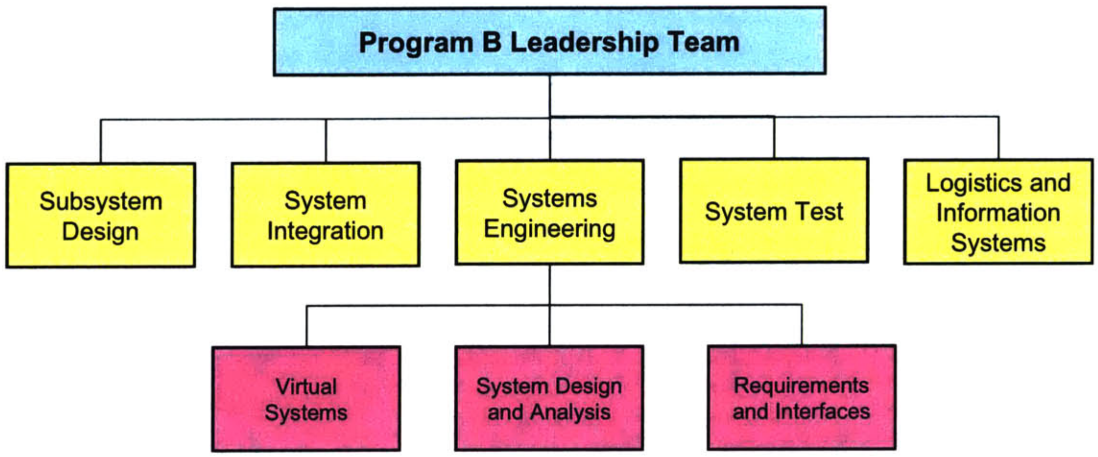


Figure 5-1: Summary of general Program B organization, constructed from interview data.

5.1.1 Scope of CDSE Research for Program B

As mentioned, this research focuses on the experiences of a sub-set of the Project B SE teams, although interviews were also conducted with SE support leadership to better understand how SE collaboration is being facilitated. One additional engineer was interviewed concerning a separate and different upgrade to the same legacy defense system. The inputs from this engineer are included in the Program B study, as the engineer's experiences are indistinguishable from the experiences of the other Program B engineers interviewed. At this phase in the SE development, engineers are completing detailed system design, while also gearing up for prototype integration and test.

As described in Section 3.4, the Program B personnel sample interviewed for this research is predominantly made up of experienced system engineers acting as SE component or task leads. All engineers interviewed were educated in engineering or science. Because Program B is part of an upgrade to an existing system, many of the engineers interviewed have been working on Program B for anywhere from 5-36 years, with the majority of interviewees having at least 5 years of Program B experience.

5.1.2 Program B Status

The Product B upgrade program has difficult and unique requirements - the resulting design must incorporate cutting-edge technology while maintaining existing system characteristics on the system interfaces, physical behavior, and overall system performance. Additionally, many of the existing components are being reused and the new components must be "maintainable, highly reliable, and cost effective over the remaining 30+ years of the system life." In essence, Program B involves a very large (non-traditional) modification to an existing system at a "traditional" price. To date, the SE and Program B leadership agree that the program has been very successful. There are no specific metrics in place to measure collaboration success, but the successfulness of the collaborations are evidenced by the fact that the teams have consistently delivered working products to the customer on schedule.

At the time this case study was conducted, Program B had successfully completed the system Preliminary Design Review (PDR), and is currently on target for a CDR in late 2007. One of the major deliverables for CDR is the development of a working Product B prototype.

Company B has an internal software and system development process that governs the development process, from innovation, to design, and proving system performance with analysis and simulation. Program B is currently performing subsystem design and simultaneously developing and using simulations to validate the preliminary system design and further develop the subsystems. To date, several major program tasks have been completed, including:

- Creation of a high-fidelity system modeling and simulation environment to be used to validate the preliminary system design and better define and evaluate the current subsystem design efforts.

- Selection of a modular, common functionality system architecture that would reduce development costs.
- Identification of commercial-off-the-shelf (COTS) parts that address both cost and maintenance constraints.

5.2 SE Collaboration Situation and Management

The CDSE collaboration situation is slightly different for the many collaborating engineers, depending on their position (leader, task leader, line engineer, etc.), the geographic distribution of their team, the team make-up, and the SE task or activity they are involved in. This section summarizes some of the collaboration situations, management techniques, and SE differences experienced by SE personnel interviewed from Company B working on Program B.

5.2.1 General SE Collaboration Situation

On a daily basis, collaboration for Program B is facilitated by a well-defined program organizational structure. Engineers and managers from subcontractors located off-site as well as engineers and managers from Company B are integrated into the overall program organization chart. There are supervisor-level positions at all sites, some filled by Program B employees and others filled by a subcontractor or supplier employee. In general, the overall program organization is broken down by task, each task being led by a program manager (PM) and technical director (TD). Within each task are technical and task leads, and below each task lead may be additional task leads or individual contributors. (Please refer to Figure 5-1.) Note that the organization changes with time to accommodate the current phase of development.

To coordinate engineering efforts at the top-level of program management, monthly program reviews are held. At the lower-levels, weekly meetings are held collectively for all task leaders within a discipline. Additionally, individual task leaders conduct weekly meetings (sometimes more often, as needed) with their individual contributors.

The Program B system design, simulation and analysis is performed by a combination of engineers whom are co-located and working distributively using collaboration tools - such as email, telephone, and teleconferences - to facilitate their communication. All interviewed personnel worked with engineers from other companies whom are located off-site.

Collaborations between Company B engineers and the contracting company engineers takes place in several different forms. Sometimes, several of the collaborating companies send "residents," or company representatives to work at Company B full-time on a temporary basis. Additionally, several Company B SE leaders, program managers, and engineers also travel often to off-site collaborating company locations to observe and assist in the development efforts of Product B. As a result of the "residents" and frequent travel, face-to-face meetings also occur frequently on Program B.

Many engineers felt they did not spend any additional time performing collaboration activities. However, several engineers and several SE leaders did feel that they spent 50% or more of their time performing collaboration tasks, such as writing emails, coordinating meetings, setting-up conferences, and disseminating information to the teams.

Although a classified program, in general, not a lot of classified information needs to be shared or transferred between collaboration sites, facilitating open communication and data transfer. However, if employees need to share or transfer classified information, there are specialized security personnel at Company B to handle classification issues.

Time differences have a mixed impact on Program B. Collaboration for Program B takes place over several different time zones, including the Western European Time Zone (WET) for the collaboration team in the UK (a five hour difference from EST) and the PST zone for the collaboration teams on the west coast. All of the interviewees were questioned as to whether time differences between collaborating sites had an effect on their work. The responses to this question were mixed, and varied to the extremes. Many interviewees, and especially SE leaders, replied that time differences have very little impact on their work. Although some interviewees relate that scheduling meetings is sometimes a nuisance, but the teams deal with it and often plan meetings for the "core hours" between 11AM and 3PM when most employees at all locations are in the office.

On the contrary, one engineer exclaimed that "time differences between teams is one of the biggest issues" that his team faces, and several other engineers agreed. Engineers related that because of the time differences, it is difficult to schedule longer meetings in particular. In the morning, when the east coast teams are ready to get to work, the west coast teams are not due to arrive for another three hours. By the time the west coast team is ready to work, the east coast team is out to lunch. After lunch, the east coast team is ready to work, but the west coast team is getting ready for lunch. After the west coast team returns from lunch, the east coast team is preparing to leave for the day. Another engineer explained that there are really only about two hours of the day (1-3PM EST or 10-12 PST) that works well for collaboration. Clearly there is a mismatch in useful collaboration time. Although this may sometimes be inconvenient when there are pressing issues to address, the time difference allows the east coast teams to work un-interrupted in the morning. Often, two hours is not enough time, and the Company B team often asks their counterparts on the west coast to come in earlier, or the Company B team needs to stay at work later to accommodate the schedule differences.

5.2.2 SE Collaboration Management

Collaboration on Program B is managed and coordinated by the team organization and structure and the agreed to development processes. The following bullets summarize the strategies and methods used by Company B to manage SE collaboration:

- Weekly and monthly team and program meetings take place at all levels of the SE organization to disseminate information, and to discuss issues and progress.
- To make better use of key stakeholders' time and availability, management often schedules similar virtual meetings around the same time.
- Cross-company formal processes are established and mandated. These processes were made with the input of all companies and the final, agreed-to processes are different from the processes of any given company or team.
- "In-house" developed Company B tools are shared with other companies and sites. These tools are searchable and provide a "firewall protected" and secure means by which distributed teams can communicate. The "in-house" tools are tailorable, have support teams, and can be used to access other tools.
- One group on the program keeps the master schedule, and each week an email summarizing the major scheduled meetings is sent out to the entire team.
- For one team, to work around the time differences and distributed-nature of the program, a teleconference is held each day between the east coast and west coast management teams to touch base and to address any immediate issues that may have transpired since the previous day. (Note - this approach is a great work-around to the three hour time difference.)
- Task leaders are responsible for the management, budget, deliverables, and team structure of their task teams. Therefore team organizations can be tailored for each task and structured to best meet the specific budgets and deliverables.
- Program progress is measured and tracked through the use of formal processes and tools, such as EVMS and an IMS (with a clearly defined critical path).

5.2.3 SE Collaboration Differences compared to Traditional SE Environment

Several engineers felt that there is very little difference between working in a CDSE environment on Program B and a more traditional co-located environment. One engineer explains that little change is felt because many of the engineers from other SE development locations travel to the Company B facility fairly often. The visiting engineers have temporary office space and access to the facilities that allow them to work as though they are in their home office. Additionally, the engineers mention that most of the collaboration tools in place do not really require extra set-up time or effort to make them work.

However, several SE personnel interviewed described several differences between working in a CDSE environment and working in a traditional SE environment. The following list summarizes some of the differences, as described by the engineers:

- Due to the collaborative nature of the team, all of the contractors are working toward the same final goal (a complete, functioning system). This is in contrast to more traditional sub-contractor awards, where a supplier is working solely on their confined aspect of the system, all contractors have a good understanding of the entire system, and can use that knowledge of the "big picture" to help them solve their "chunk" of the system.
- In general, communication is just more difficult - there is an impact of not seeing people "face-to-face," such as not being able to read people's body language.
- New and different processes had to be structured to accommodate the differences between different company-internal processes.
- When working with outside companies, it is often more difficult to allocate or re-allocate resources as changes occur. Since most interactions with outside companies are contracted, schedules and resources are allocated way in advance

and it is often more difficult to meet projections when resources are not under your direct control.

- The standardized design processes and tools in place on Program B are in general new, and some interviewees consider the standardization unorthodox for Company B.
- There are additional tool challenges that are encountered in this CDSE environment, for example, collaborative tool interactions must overcome corporate firewalls.
- The simulation environment and organization must be accessible to engineers at all locations. Therefore the simulations and simulation packages must have all of the required platform data, files, and compiling and build software to support multiple different operating platforms. Additionally, the simulations are modular in a the sense that user's can choose which simulations they need or want to use together and there is a separate interface that ties the simulation blocks together. This simulation architecture was chosen so the engineers at other companies or locations can add their own simulation software or use only the software models that are applicable to them.
- The simulation environment for Program B has additional obstacles that are not issues in a traditional environment, such as proprietary data sharing and non-disclosure agreements that must be addressed in order to seamlessly integrate all of the models and provide the entire simulation software set to all locations.
- There are more discussions and more frequent interactions between teams.
- There are several additional tools used to facilitate the collaboration environment.
- There is not as much brainstorming - meeting time is mostly comprised of an engineer describing work that he/she has completed for team review. Rarely

do engineers sit around the table and brainstorm better or other ways of doing things, as meeting time is scarce.

- A common database available to all teams has been put in place and as the result of several lessons learned at the beginning of the program.
- In general, for Program B, the structuring of the prime contract award is different compared to more traditional development environments. Specifically, in the past, the customer would act as the prime contractor, coordinating integration and design activities among subcontractors. More recently and as a result of budget cuts and a shrinking workforce, the customer awarded Company B the role of prime contractor. With that award comes a lot of additional responsibility and coordination between contractors. As a result, a lot of additional up-front work by Company B has gone into defining the system architecture and sub-contract awards.

5.3 SE Collaboration Tool Use

The discussions with SE personnel on collaboration tools was centered on their use in the daily practice of SE, potential issues and barriers personnel have encountered, and success factors that have been developed to improve collaboration tool use. The paragraphs in this section summarize the input from from Company B interviewees on these topics.

5.3.1 Collaboration Tool Use Overview

Company B coordinates and provides almost all the SE and collaboration tools for the development of Product B. The set of SE development tools used by and provided by Company B includes: Pro/Engineer, DOORS, Mentor Graphics, Cadence tools (used to emulate hardware), Computer Aided Drafting (CAD) tools, and virtual simulation tools for hardware and software development.

Collaboration for Program B is accomplished by using typical collaboration tools in addition to some specific tools for database management, configuration control, and requirements management. Typical tools include email (no standard email program), telephone, teleconferences, and net-meeting capabilities with tools such as Sametime, Netmeeting or WebEx.

To determine the tool set that would be used, Company B leveraged some of tools already in use at some of the contracting companies. Although a standard set of tools is mandated to ensure artifact consistency, one SE support leader relates that companies are allowed to use different tools for certain tasks, but the tools they use must be compatible with the core tool set (for consistency and to manage the final products sent to the customer).

The SE personnel interviewed consistently referred to the following list of tools that they used on a daily basis:

- **DOORS:** DOORS is a Commercial Off-the-Shelf (COTS) software tool distributed by Telelogic used specifically for requirements management and development. All sites have access to the DOORS requirements database. Note, DOORS is not used for requirements configuration management - configuration management is accomplished via document versions of the requirements.
- **Formal Data Management Tool (FDMT):** FDMT is a Company B custom developed database used program-wide to store and share configuration controlled data. It is essentially a document "vault" for released documents. All material on FDMT is unclassified.
- **Informal Data Management Tool (IDMT):** IDMT is a Company B developed and unique tool for the storage, management, and control of all Product B data. IDMT allows engineers from different locations to post working documents, data, and other products in draft form to share with engineers at any location. It is widely available to all developers through a secure network and

is the “home” of several additional tools. IDMT is unclassified.

- **Sametime:** Sametime is an IBM product which allows both voice and visuals to be used synchronously from multiple locations. It enables and coordinates virtual meetings and can also provide instant messaging capabilities. Company B uses Sametime purely for coordination of visuals.

- **Microsoft Office Suite:** The Microsoft Office Suite includes Microsoft Word, PowerPoint, and Excel. Program B also uses Microsoft Access as a database tool to record interface descriptions and controls. These tools are used mostly informally to support team communications and organization. Powerpoint is used often to support team communications (presentations, reviews, data sharing). Note, Microsoft Project is used for program management.

- **Problem Tracking System (PTS):** PTS is a home-grown tool developed and maintained by Company B used for tracking issues and managing the requirements. It is used to report and manage issues, their status, and resolution. The PTS tool is also distributed and linked to all of the distribution sites via IDMT.

- **Videoconferencing:** Company B has invested a great deal of time and money into setting up videoconference facilities. Some subteams utilize video conferencing, however the conferences are typically kept short and to the purpose to facilitate some of the awkward dynamics that can arise. Additionally, the videoconferences meetings require a great deal of set-up and not all sites have them (rendering them useless).

- **Email:** Although there is no specified email program to be used by all collaborators, many engineers and leaders referred to Microsoft Outlook as the predominant email program. Outlook is used for both email and calendar functionality. A new Outlook calendar functionality has recently been introduced to schedule virtual meetings. The Outlook tool has been integrated with the conference room booking system and teleconference system to allow one invitation to be sent to all meeting attendees, including the “conference room,” and a "call-in number" (therefore reserving the meeting room and number). This tool works great for meetings held within Company B, but unfortunately does not have the capability to cross organizational boundaries.

- **Teleconferencing System:** Although easily overlooked, the teleconference system is a key enabler for virtual meetings and CDSE. The ability to have a dedicated open phone line where teams from across the United States can call in and "conference" at the same time is critical to many of the weekly meetings.

- **Virtual Private Network (VPN):** VPN enables collaborating team members to communicate securely over a public internet or network. Company B provides VPN access to specific sections of their internal network through VPN, including IDMT, FDMT, the DOORS database, the modeling and simulation environment, and PTS. VPN access is secure and is enabled by the use of "RSA cards."

One engineer also jokingly mentioned the "airplane" as another collaboration tool, as several team members are often traveling for collaboration purposes.

Net-conference instant messaging services are used by some engineers to stay in touch real-time with employees when phone calls are not available (such as during the monthly status review meeting).

The IDMT and FDMT tools are also searchable and contain a lot of legacy data about the Product B program history.

For the Program B simulation environment, the Program B team typically uses Matlab and Simulink. The simulations are stored on a server located at Company B, and are accessible via the network (using VPN) by remote sites. A program called Dimensions is used to provide version control, allow for filtered views, and additionally provides the capability to run, compile, and build the software on a PC platform.

In order to use most tools, "seats" are purchased by Company B for all sites and re-evaluated yearly depending on the estimated number of engineers who need to use the tools. For other tools, site licenses are bought for each location through Company B.

All collaboration tools are unclassified. Although Company B has the tools and facilities for classified teleconferences, classified interactions are few and far between. The Program B SE leadership explained that this is mostly due to the long program history, since many of the engineers have memorized the classified values and do not need to discuss them (or they can be easily looked up). If classified discussions are needed, they are often very short.

Unlike most of the large companies Company B has subcontracted work to, Company B does not outsource their network infrastructure and maintenance. Although this is a lot of additional work for Company B, the positive aspect is much greater control over the network reliability, repairs, and outages.

Tool training is widely available and widely utilized by some. Training sessions for those interested (typically one week long) are held several times a year in person at the Company B facility. (Engineers from other locations typically travel to receive the necessary tool training.) There are essentially no online training courses offered, aside from those online help guides provided by COTS programs.

When new tools are acquired, Program B leadership usually coordinates large training sessions with the tool company representatives. The tools developed by Company B for managing and storing Product B data have associated training classes offered via Company B personnel periodically. Normally, one SE support team leader explains, when a new site/company is using a Company B developed tool, the leader will send out some of his managers to the new site to provide help in setting up the tool and demonstrate to the new users how to use it. Additionally, many of the Company

B-developed tools have help files and "FAQ's" to facilitate their use.

Currently, all interviewed personnel find the tools very useful. One engineer explains however, that when some tools were first introduced, they were not a big hit; however, once a critical mass of tool users was reached, there are now fewer complaints. However, the tools are sometimes unreliable (like Sametime) and some engineers find it simpler and easier to set-up a teleconference line and coordinate meetings through the use of PowerPoint presentations with slide numbers.

To ensure secure data access and transferal, the use of "SecureID" (RSA) cards and VPN capabilities are used. The VPN allows engineers from all over to access the Company B server from a remote location. Company B is in general very concerned with tool access and security. Each time contractors and collaborating engineers are accessing any of the shared databases, they are accessing the internal Company B databases. Company B takes extra precautions to ensure that contractors cannot access Company B proprietary or other Company B program information. To ensure that engineers are only able to view and modify the information they are privileged to access, rigorous authentication and authorization standards are implemented by Company B. The VPN allows the viewing and transferal of secure ITAR restricted information between sites, and also overcomes the issue of corporate firewalls. As a testament to the success of the VPN and authentication processes used at Company B, SE support staff leadership reports that the system has never been violated or hacked into in the fourteen years it has been in use.

The SE development and collaboration tool issues are reviewed daily. Each collaboration site has a local site representative to oversee the tool use. The SE support team meets and reviews the bugs and issues reported by the local site representatives. There is also a "bug reporting system" in place to record the issues encountered by engineers.

5.3.2 Identified Potential Collaboration Tool Use Issues

Although the Program B SE personnel express their satisfaction with the collaboration and SE development tools in use, several potential and engineer-experienced tool issues were discussed during interviews, including:

- **Configuration Control Inefficiency:** Currently, formal requirements management and configuration control is coordinated via "document" form of the DOORS database. Several engineers reported that this process is timely and inefficient for the engineers and results in tools (such as DOORS) that are not currently being used to their full capability.

- **Establishing and Acquiring Tools at all Sites:** Although all of the sites are currently using the same tools and the same versions of tools, getting all companies the same tools and tool versions was not an easy task. Getting certificates and tool licenses at all sites for use of the same tools can result in three-to-four month delays and litigation issues as companies determine tool management responsibilities and tool use guidelines.

- **License Expirations:** Since Company B purchases and maintains the tool licenses at all development sites they must monitor and re-purchase tool licenses as necessary. Unexpected license expirations can cause down-times or delays.

- **Multiple Accounts Necessary:** Having access to one of the collaboration tools, such as IDMT, does not automatically mean the user has access to the tools available via IDMT. Therefore additional accounts and passwords are necessary to access all collaboration tools (and employees need to remember several different passwords).

- **Network Bandwidth and Speed Issues:** Since all SE personnel are accessing the Company B internal network each time they access the shared collaboration tools and databases, the SE support team must contend with issues of network bandwidth and connection and data transfer speeds.

- **Transfer of Classified Information:** Sometimes it is necessary to share classified Program B information, but there are no classified data networks in place to provide instant data transfer. The teams must securely mail or courier classified material between sites. This does introduce some delays, and can be costly when you consider the delivery costs for overnight mailing, etc.

- **Tool Security:** As mentioned in Section 5.3.1, one of the major issues the SE support teams must address includes the security and access by outside engineers into the Company B Program B network and databases. One key issue that needs to be overcome is corporate firewalls.

- **Tool Stability:** Tools such as Sametime crash often, disrupting meetings and necessitating a computer restart. Collaboration tools are improving, but they are still not perfect.

- **Tool Usability and Learning Curve:** The collaborative tools have steep learning curves, especially DOORS and CAD tools (such as Pro/Engineer). Additionally, some tools are not very user friendly, or all the bugs haven't yet been worked out, such as importing pictures into DOORS.

- **Corporate Tool Support Responsibility Ambiguities:** Tools are typically purchased and managed by the individual companies. Since Company B provides the tool licenses and certificates, there are ambiguities as to whether Company B is responsible for the maintenance and support of the SE and collaboration tools being used. It is also difficult for Company B support teams to work across various companies and their program and company-wide information technology services.

- **Tool Version Compatibility:** Not all development sites use the same versions of tools for all tasks. Some companies, using "in-house" tools, have different upgrade versions, etc. that lead to issues with compatibility (not all upgrades are upward compatible).

5.3.3 Identified Collaboration Tool Use Success Factors

Several lessons learned and success factors have were discussed with Program B SE personnel, including:

- **Tool Training:** Tool training is widely available for Program B engineers, should they choose to participate. Additionally, Company B outsources tool training when they do not have the tool expertise in-house, such as when new tools are acquired. Large training sessions with tool company representatives are instead scheduled. Also, Company B provides tool training for the the tools developed in-house.

- **Create a Central Data Storage Location:** Although centralized databases are now being used, the program began with distributed databases kept separately at each location. After experiencing issues with configuration management and maintaining up-to-date information (delays with synchronizing databases), a centralized database system was put in place. Company B currently serves as the repository for the final electronic products and final tools used by all companies.

- **Modify Standard Tools for greater Functionality:** Internal to Company B, the Microsoft Outlook calendar functionality has recently been enhanced to schedule virtual meetings. The Outlook tool has been integrated with the Company B internal conference room booking system and teleconference system to allow one invitation to be sent to all meeting attendees, including the "conference room" and "call-in number" (thus reserving them). The enhanced tool works great for meetings held within Company B, but unfortunately does not yet have the capability to cross organizational boundaries.

- **Use a Single Version of Common Tool Versions:** One lesson Company B learned as a result of setting-up collaboration tools is that you should use the same version of tools. For example, several years ago many separate vendors had the same type of tools...and each location used a different version of say a design tool. Therefore, each different design tool may not be compatible. Later, each company was told to use the same version of database tools, allowing easier integration and cohesion from each location. When Company B began managing these tools, they worked with design engineers to chose one tool versus allowing several different vendor tools. This significantly cut down on the tool support infrastructure. Although some people needed to be retrained for one common tool, its worked out better in the end from the perspectives of integration and commonality.

- **Use Templates and Standards:** Standards are widely available and typically used on defense development projects. To provide artifact consistency, DOORS modules are standardized for different specifications, using templates such as MIL-STD-961-D (DoD Standard Practice for Defense Specifications).

- **Evaluate Tools on smaller Programs First:** Before deploying new tools on a large, distributed program, several of the tools in use were first tried out in smaller programs on a smaller scale to ensure that they meet the needs of the Company B and Program B.

- **Have Tool Coordinators at each Site:** Since the Company B internal tools were introduced at off-site locations, the Company B SE support team has had coordinators at each site. On a daily basis, the coordinators meet and review the bugs and issues reported by the local site representatives, and can often close the issues within minutes.

- **Maintain "Tool-Neutral Documentation":** Another lesson learned by the SE support team is the importance to maintain neutral tools and documentation formats. For example, not everyone has access to DOORS software, and therefore the support team archives and configuration manages requirements documents in "document" form. Additionally, sometimes tools are no longer made, or the tool company is bought out by a larger company, so the support team believes it necessary to store data in neutral formats (such as database or document forms), to allow for easier transformation into new tools.

- **Control Access to Tools:** Another lesson learned is the necessity to have limited "write access" to data as controlled by the tools. An engineer explained that previous to establishing read and write privileges, certain employees were able to modify/correct the products delivered from other engineers and contractors, which resulted in errors and protests from other engineers. It is therefore very important to determine ahead of time which engineers need what type of access to certain tools, software, requirements documents, etc. to prevent errant modifications.

- **Use Simple Tools where Possible:** Since tools such as Sametime and Netmeeting are not often reliable, engineers have adapted by using simpler tools: setting up a teleconference line, posting slides/documents on the shared network, and using page number to coordinate meetings at different locations.

5.4 SE Knowledge, Data and Decision Management Practices

Upon completion of all interviews and data review at Company B, as with the results from Company A, it was abundantly clear that SE and overall Program B knowledge and data management is in general facilitated by collaboration tool use. This section discusses how tools are used to support knowledge, data, and decision practices, in addition to other data management techniques on Program B. Also included in this section is a review of the issues and success factors associated with knowledge, data, and decision management.

5.4.1 Collaborative SE Knowledge and Data Management Overview

Knowledge and Data for Program B is shared and managed predominantly via unclassified tools and databases. The IDMT allows engineers to share information, data, and knowledge in draft form to all sites in a controlled setting via the internet. Sharing of formal data (released documents) is accomplished using the FDMT. Data can be found using search software in either the IDMT or FDMT database. To locate information, one can search by date, title, author, or subject to find information in the over 380,000 searchable entries in the IDMT database.

One key method by which knowledge and data is shared is through rotating "residents." These residents are part of the core team at Company B and either visit or work at off-site locations and then communicate the information they have gathered back to the core Company B team. However, this approach results in a great deal of travel still being done.

Another method by which knowledge and data are managed is by maintaining risk registers and action item listings. The risk register and action item listings are used to record specific decisions, knowledge and actions associated with an issue. The issues remain active until retired (and must be signed-off on) - but the history of a "risk" or "action" is maintained and is shareable.

Of course, knowledge and data is shared via meetings: including staff meetings, weekly program reviews, program meetings. There are also monthly program reviews where all of the key management personnel are briefed and updated on the progress made in each area. In addition, the large number of personnel on the program, especially those with such long histories of working on Product B, facilitate knowledge sharing and maintain a program history. Knowledge and data are also shared via email, telephone, teleconferences and meeting minutes.

A really interesting practice Company B has developed is emailing a weekly document all of the key meetings and reviews for the upcoming month. There is also a widely used program calendar in the IDMT docushare.

Configuration management processes are in place to control the released forms of documents. CM is done at all levels of the documentation to manage requirements knowledge and data. The configuration control processes is specified in the standardized design and development processes agreed to by all companies. CM on Program B consists of both paper document control and electronic document control (mostly in multi-platform friendly forms, as explained in Section 5.3.1).

Program B SE processes specify that teams are required to take meeting minutes, and document the requirements, design rationales, and models that they create. To assist in the taking of meeting minutes, one engineer explains that some of his co-workers actually use the conference room computers to take meeting minutes in real-time and in-front of the audience. This saves time (you don't have to go back and type them up) and also allows for real-time correction and buy-in by those present.

Unfortunately, since Program B is an update to an existing system, the engineers must also retro-actively create and update the existing Product B documentation, for CMMI compliance. Although sometimes difficult, timely, and cumbersome, the engineers believe this documentation process is important and necessary for knowledge sharing.

Most engineers do not find it very difficult to find the information they need when they need it, since the database tools are searchable. Additionally, Company B spent a great deal of time developing IDMT, including creating a document library of the old diagrams, specifications, and technical papers from earlier work on Product B.

In addition to IDMT and FDMT, design decisions and requirements changes are documented in the PTS, which allows a great deal of information to be captured about each change. Included in the PTS, in addition to the change documentation, is a summary of the process, key stakeholders, and recorded stakeholder buy-ins.

Classified data sharing is not a big issue, but its management can sometimes be difficult. Currently, classified data is stored separately and is often only referenced in the unclassified databases. Occasionally, the SE support team has to "strip" the unclassified data out of the classified databases to make it widely available on the unclassified network.

Feedback sessions have been utilized to capture issues and knowledge regarding the use of Program B tools and simulation software.

5.4.2 Collaborative SE Decision Making Overview

Currently, decisions are not formally documented on Program B. Most decisions are informally documented in the Risk Manager, Action Item tracking systems, IDMT and FDMT. Additional informal methods such as technical memos, presentations, and reviews are used to record and disseminate decisions and other design critical data. The recording of decisions and rationales are enforced by the Program B managers and technical directors.

One leader explained that issues that arise within an element can typically be dealt with at the level of organization they arise. In essence, managers and task leads are empowered to make their own decisions, unless the issue involves other program offices or other budgets. Most major decisions or changes are documented in emails to the entire team, or those affected. Last, meeting minutes are often recorded and shared to document discussions and decisions from team meetings. For example, one task leader explained that decisions on his team are typically made at the lowest level (within the lowest sub-group boundaries). Decisions that require cross subteam changes or input are escalated up the proper chain of command. Issues and decisions within this group are usually addressed during the weekly meetings.

When it comes to decisions about system requirements or interfaces, there is a design review board in place. The requirements issues and resolutions are tracked and recorded in PTS.

Many engineers relate that decisions are sometimes not very well documented on Program B.

Many engineers felt their decision making was unaffected by working in a CDSE environment; however some engineers feel that it is sometimes easy to be weary. Although there is a distinct one-team sentiment, when working with other companies, one is not always certain what a company's "ulterior motive" may be when it comes to design choices, etc. It is easy to wonder, "Are the decisions best for the final product or for the company creating the design?"

5.4.3 Identified Potential Collaborative SE Knowledge, Data and Decision Management Issues

During interviews, several issues were discussed with SE personnel regarding knowledge, data and decision management. These issues are:

- **Decision Making is Slower:** As there are no formal decision making processes or records, decision making occurs much more slowly, as there are significantly more players that need to weigh-in from all different companies and locations.
- **DOORS Requirements Central Repository and Development Time:** One engineer explains that there is not a central repository for documents and requirements that is easily accessible, up-to-date, and version controlled. For Program B, DOORS is predominantly used for requirements traceability - thus it is not being used to its full version control capabilities. Additionally, since

DOORS is not being used as the version controller, there is a very long turn-around time from the start of a document to the time it is available on the database.

- **Proprietary Data Development:** With development of systems and software being completed collaboratively, what products are proprietary? One clear example from the Company B experiences involves software developed by a contracting company. Company X may develop proprietary source code; however, once Company B compiles that code for integration and test, is the software still proprietary? These battles are often time consuming and difficult.

- **Company Proprietary Data Sharing:** The free flow of data, information, and especially knowledge between companies and locations is also impeded by company proprietary data. Because competitors are working together collaboratively on SE, alleged company proprietary data must be shared in order to understand the system design, interfaces, and integration. Company proprietary data sharing agreements are in place, however there is often tensions and disagreements over what constitutes company proprietary information, who has access to view the proprietary information, and what level of information can and will be shared. To complicate the issue, since not everyone has signed a proprietary data agreement, storing and transferring proprietary data on the open internet or in a collaborative databases is an issue.

- **Configuration Control of Requirements Process:** Configuration management (CM) processes are in place to control the released forms of documents. However, several engineers feel that it is a lot of additional work, however, to get the DOORS databases into the paper form specified for configuration control. Specifically, engineers need to make the DOORS requirements attributes into additional document appendices, rather than using the DOORS database layout itself for CM, for which it is designed to do.

- **Recording Design Rationale:** Several engineers related that design rationales are not always well-documented. Often engineers have to track down the right people to find out rationales. The recording of rationale is people-dependent and some feel that it is not always well enforced by management. This issue is exacerbated by company culture, where for instance, engineers explain that some of the older population of engineers believe knowledge is power, and therefore they prefer to keep some product information and design rationales to themselves.

- **Tools and Processes Needed for Meeting Minutes:** Currently, several engineers report that meeting minutes are not well recorded on Program B. Not only would minutes help in the documentation of decisions made at meetings, but it would also allow key personnel to remain informed. Everyone, unfortunately, cannot attend every meeting, and good meeting minutes would more easily allow the teams to remain aware of key issues, actions and other relevant information. For the most part, the writing of meeting minutes and design rationales is person-dependent. Some do it, some don't. Others take meeting minutes, but they are useless. Additionally, although recording meeting minutes is part of the process, engineers note that it is rarely enforced by management.

- **Meeting Attendance and Accountability:** During informal virtual meetings between companies, it is often difficult to take official, formal records and keep an accurate account of attendance and decisions. Since many decisions are made during meetings with key stakeholders, it is necessary to record and hold accountable those making and agreeing to decisions. Although many stakeholders participate in the meetings and serve as witnesses, it is often difficult to hold stakeholders accountable.

- **Collaboration Tool Navigation and Searchability:** Although most engineers feel that IDMT and FDMT are excellent knowledge and data management tools, some engineers report that they are difficult to navigate, as the library folder architecture is sometimes not very intuitive. This difficulty is somewhat overcome by a lot of cross-referencing. One engineer describes that the library is searchable, but the search engine does not generally return searches in a useable format - the search usually returns "all or nothing."

- **Distributed Decision Making:** SE leadership explained that this is in general a Program B issue, predominantly due to the size of the program and the complexity. Decisions are sometimes not very well documented, and instead are usually disseminated via word of mouth.

- **Person-Specific Fixes:** One engineer related that a lot of the tool fixes are person-specific - i.e., if the current tool administrator was to leave, several of the problem fixes Company B has derived would likely resurface.

- **Artifact Consistency:** One engineer related that consistency between the different DOORS modules is difficult to maintain, and there is much repetition between documents, which creates some protests from the engineers.

5.4.4 Identified Collaborative SE Knowledge, Data and Decision Management Success Factors

With such a long program history, several lessons learned and success factors for knowledge, data, and decision management have been developed and experienced by Company B, including:

- **Keep Record of Program History:** The legacy system design, design rationale, technical memos, architecture and specifications are critical to the system update. To facilitate sharing of the legacy system design details, Company B spent a great deal of time creating digital versions of the old paper documents they had in their possession. These materials were added to a searchable digital library in IDMT and shared with all of the developers via VPN.

- **Create Searchable Databases:** Many of the engineers and SE leaders were very satisfied with the search capabilities provided by the database management tools. The search tools, although not always successful, provided an excellent way to locate data.

- **Team Calendar Email:** Although there is also a widely used program calendar in the IDMT, many engineers do not take the time to review calendars. Emailing a weekly document all of the key meetings and reviews for the upcoming month has kept many engineers informed of the important ongoings for the program.

- **Use Tools to Enforce Process:** Tools such as PTS are used to support the requirements management processes. The introduction of a PTS was out of necessity - previously requirements and design comments were collected in Microsoft Excel, which provided no configuration management or history, etc. A much more successful tool, PTS is configuration managed and allows a great deal of information to be captured about each requirements change.

- **Use Feedback Sessions:** Feedback sessions have been used to collect the issues and obtain inputs for correcting problems with the software simulations.

- **Discussion Group Failure:** It is important to note efforts that have failed so they are not repeated. In an effort to retain a knowledge history, one group utilized the "Discussion Group" feature on FDMT. The team leader required all of his team members and all users of his teams' software to register on the group and monitor the discussion threads. Although an excellent idea, it did not succeed. The leader believes this idea may have failed for 2 reasons: 1) Engineering culture prevents engineers from publicly requesting help (especially for easy things); and 2) Perhaps the tool itself was too cumbersome.

- **No Proprietary Data** As proprietary data sharing is an issue, to facilitate sharing and integration, Company B does not encourage the development of proprietary information, but they respect that some businesses consider their core processes proprietary. The customer is the ultimate owner of products developed and currently the program has no company-proprietary data under development.

5.5 Collaborative SE Practices and Processes as they relate to Product Development

Although discovering how the SE practices and processes as they relate to the development of Product B, like Product A, is one of the key areas of interest of this research, many of the SE processes, methods, and technical artifacts are considered either proprietary, classified, or are export controlled and therefore their sharing is restricted. Consequently, this section discusses at a high-level the general SE practices and processes as they relate to Program B development.

5.5.1 Collaborative SE Practices and Processes Overview

Program B faces several unique challenges. Program B involves an update of an existing system, and therefore the system being re-designed must fit into the entire Product B system as though it were a "black box" - all interfaces and functions with the rest of the system must be transparent, despite the updates. With those particular constraints foremost in the design, the architecture and thus the SE efforts by Company B and Company B's contractors are constrained.

One engineer explained that, with the addition of ISO 9001, CMMI and other quality certifications, most of the major defense contractors have very similar product development processes. To facilitate a coherent design and uniform artifacts, engineering processes and tools were standardized across all the development sites. The tools chosen for use were selected to enable the processes. The standards and processes were developed with assistance and buy-in from the individual contractors and suppliers. In the end, the agreed to processes and tools were different from the traditional tools and processes used by each company. The SE leadership team explained that since contractors "bought-in" to the processes, there have been relatively few issues with enforcing the agreed to standards. The standard agreements are called out in contracts or formal plans, and therefore the sub-contractors must eventually come around. The SE leadership team notes that there was "pushback" on the processes from developers only if the standards were not clear up front. There have been arguments in the past, for example, about drafting guidelines. Standardization and formal processes on Program B were requested by the customer.

A modular system design has allowed several of the modules that comprise the system under development to be independently developed off-site. Independent, well-defined "chunks" (subsystems) of the system are given in entirety to off-site developers. Strict standardization is enforced at both the subsystem and common processes level to facilitate integration and interface development. For example, a common "process" may consist of common digital communications between subsystems, or a common power supply. Figure 5-2 illustrates the general concept of how the common processes enable the modular design. The great deal of standardization in processes and tools has enabled the independent, modular development. (Note that a modular design was not possible in the original Product B development program; modularity has been enabled by improvements in technology.)

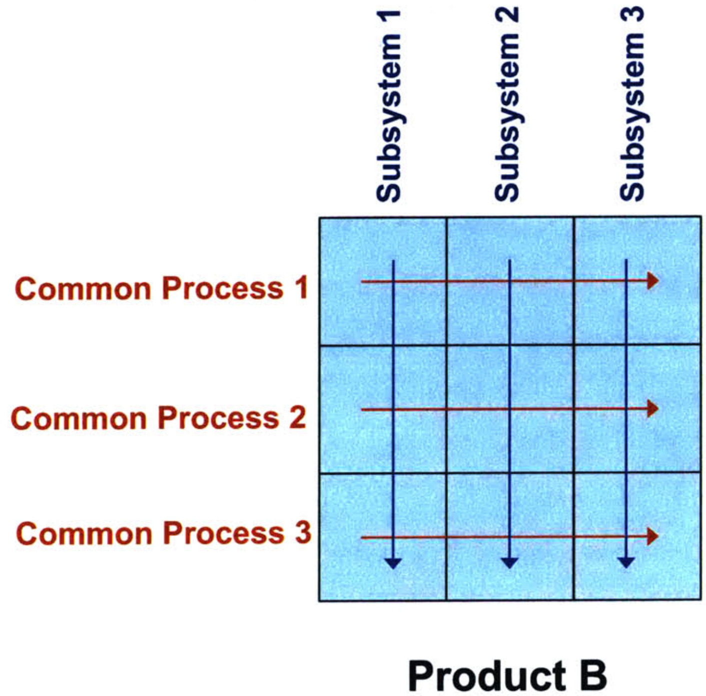


Figure 5-2: General concept of Product B common processes and relationship to modular design.

SE leadership exaplined that the modular, standardized system design is enabled further by a great deal of up-front program work. Up-front work consisted of defining the standardized common systems and processes, as well as defining the system architecture and infrastructure. A modular system design was not chosen solely because the system was being developed distributedly, but it was one of the motivating factors. Note that although the modular development approach, with well defined common systems and processes, allows a greater deal of independent development; it does have one major drawback - namely design changes are not independently contained within a module. For example, any changes originating in one module that require changing the common systems will likely require a change to all (or several) independent modules.

The team organization for Program B was based on the organization and system architecture of the predecessor Product B program, which is very similar. One engineer explained that keeping a similar team organization is both beneficial and presents an issue. It is beneficial because the organization is familiar and the team is familiar with the structure, but it is also an issue because the system architecture is different (i.e. modular) and therefore the organization may have been better optimized to serve the architecture.

A very interesting aspect of the collaboration efforts was the way the product architecture was assigned and awarded to contractors. Prior to awarding subcontracts, Company B determined which aspects of the system were very tightly coupled, and should therefore be developed in a more "traditional" engineering environment. Therefore work products were assigned based on the system coupling and complexity.

A very large aspect of Product B's design process involves the development of intense system simulations to facilitate system design, analysis, verification, and integration. The simulations developed by Company B are accessible to the entire Program B team. The simulations are in general a bit different from a simulation developed in a traditional SE environment. First, the simulations must be able to be used by any engineer, anywhere on the program. Therefore the simulations and simulation packages must have all of the required platform data, files, compiling and build software to support multiple different operating platforms. Additionally, the simulations are modular in a the sense that users can choose which simulations they need or want to use together and there is a separate interface that ties the simulation blocks together. This simulation architecture was chosen so the developers at at other companies or locations can add their own simulation software or use only the software models that are applicable to them. Last, there are issues such as proprietary data sharing and non-disclosure agreements that must be addressed in order to seamlessly integrate all of the models and provide that the entire simulation software set is available to all locations. The intense simulation environment in use by Company B is very similar to the SBA (Simulation Based Acquisition) environment described by Zittel. As described in detail in Section 2.7.2, Zittel explains that the SBA environment has innumerable SE benefits, if the issues described above can be overcome.[43]

5.5.2 Collaborative SE Final Product Integrity

All interviewees were asked how they believed the final product integrity would be impacted by CDSE, and all interviewees agreed that CDSE efforts would positively impact the final product. Some of their responses are summarized in the following list:

- The overall final product will be better, as the program is utilizing the expertise from off-site engineers and companies.
- Due to the collaborative nature of the team, all of the contractors are working toward the same final goal (a complete, functioning system). Instead of more traditional sub-contractor awards, where a supplier is working solely on their confined aspect of the system, all contractors have a good understanding of the entire system, and can use that knowledge of the "big picture" to help them solve their "chunk" of the system.
- Some companies can do things others cannot, and vice versa, therefore the collaboration utilizes the best of each company's skills.

5.5.3 Identified Collaborative SE Practices and Processes Lessons Learned Success Factors

As this is a little-explored area of research, the lessons learned and success factors relative to CDSE processes and practices are very important. The following list summarizes the key lessons learned and success factors related from the Program B interviewees:

- **Obtain Company Buy-In into the Processes:** The SE and design processes developed for Program B were based on the shared expertise and collective input of all major contracting companies. Therefore each organization had buy-in from the supporting organizations and were contractually obligated to follow the agreed to processes. Since contractors "bought-in" to the processes, there have also been relatively few issues with enforcing the agreed to standards.

- **Modular System Design:** A modular system design has allowed several of the modules that comprise Product B development to be independently developed off-site. Independent, well-defined "chunks" (subsystems) of the system are given in entirety to off-site developers.

- **Strict Product Standardization:** Strict standardization is enforced at both the subsystem and common processes level to facilitate integration and interface development.

- **Necessary Up-Front Work:** The modular, standardized system design is enabled further by a great deal of up-front program work. Up-front work consisted of defining the standardized common systems and processes, as well as defining the system architecture and infrastructure.

- **Be very Specific very Early:** A lack of specificity sometimes resulted in ideas being "run away with" at the beginning, only to create additional rework later on.

- **Widespread Availability of System Simulation Tools:** Several versions of simulation software are available and accessible to all sites. The availability of the simulations to all sites allows developers from all locations to test and simulate their design and development product with the rest (or other parts of) of the simulated system, and will likely facilitate integration (by identifying potential issues in the design stage and remedying them in phase).

- **Enforce the Common Development Processes:** At the start of the program, one engineer related that some companies continued to follow their own internal processes (such as internal design reviews), that often took additional time that was not necessary. After time, Company B intervened and a set of standard processed were more closely followed as lessons were learned.

- **Perform Analysis to Determine System Architecture Distribution:** A very interesting aspect of the collaboration efforts at Company B was the way the product architecture was assigned and awarded to contractors. Prior to awarding subcontracts, Company B determined which aspects of the system were very tightly coupled, and should therefore be done in a more "traditional" engineering environment. Therefore work products were assigned based on the system coupling and complexity. (This is a key area for future research, as described in Chapter 7.)

5.6 Social and Cultural Collaboration Experience

A variety of different social and cultural topics were discussed with the Program B interviewees. The following paragraphs summarize the findings from the topics discussed, as well as some of the issues and success factors that were identified.

5.6.1 Social and Cultural Collaboration Experience Overview

Company B SE leaders explained that Program B has over a half-century long history, and many engineers at Company B have been working on the system for a very long time. So long, in fact, that many of the interviewees stated that the program itself has its own "culture." For the most part, that is a great thing, since many workers from all over the globe know each other, have perhaps even met face-to-face, and have a shared vocabulary and history. This established history enables the team members to speak freely.

In working on Program B for so long, one engineer explained that he and his co-workers have an established history and a spirit of cooperation. Because of the ability to interact frequently over many years, this engineer has developed a great level of respect for many of his coworkers.

The following paragraphs summarize the responses on some of the key social and cultural topics discussed:

- **Job Satisfaction:** All engineers interviewed expressed that they were satisfied and enjoyed their job and the CDSE environment, although they found the environment very challenging. One interesting challenge reported by an interviewee was the additional responsibility to ensure that the employees he leads at other locations and companies have executable plans and are satisfied in their jobs, etc, even though they are technically employed by other companies.

- **Face-to-Face Meetings:** Every interviewee reported they met face-to-face with all of the engineers they have worked with. Many engineers also expressed that they prefer face-to-face meetings, specifically because getting the right people all together at the same time usually facilitates more work getting accomplished, permits the reading of body language, and is in general more comfortable. Another engineer explained that face-to-face meetings allow one to read the tone, words, looks, and body language of a fellow conversationalist; and many of these important non-verbal feedbacks are impossible to interpret in email and sometimes even difficult to tell by telephone (i.e body language).

- **SE Team Leadership:** The leadership team described some instances of company "philosophy" crashes. For example, different management structures and decision making cultures within companies sometimes make it difficult to work on the engineer to engineer level (e.g. needing management permission before acting). Another example is the inflexible nature of some companies, whereas other companies strive for flexibility.

- **Language and Interpretation:** Most likely due to the long, established history of the program, most engineers report never having experienced any language or interpretation issues. However, some misinterpretations have occurred and a Program B glossary has been developed to combat these issues.

- **Company Cultures:** Several engineers believe company culture differences to be one of the biggest Program B hurdles. Companies (and their engineers) are often set in their ways and getting everyone to work collaboratively on the same team is difficult - as egos and personalities often get in the way.

- **Trust:** Most engineers report having no issues with trust of their fellow engineers at different companies or locations, likely due to the established program history. However, a few engineers reported it is sometimes difficult to depend on the estimates and advice of people at other companies and locations for several reasons: there are some cultural differences between companies and the way they do business and therefore you cannot be sure what they really mean; and in the grand scheme of things, each company or corporation exists to make money, so they may unintentionally seek solutions that are in their best interest, but not necessarily in the best interest of the final product.

- **Team Communication:** Communication on a program of this size is in general very difficult, and having distributed development sites has complicated the communication issue. The longevity of Program B makes communication easier, since many team members have been on the program for a very long time and have worked together collaboratively for years. One engineer explains that once he has gotten to know the team members he works with, he can adapt to distribution and read their "body language" through the phone - picking up on signals from the tone of voice and the manner in which things are expressed.

- **Line Management and Career Development:** Company B interviewees reported that they had no issues with line management or career development, since most of their supervisors are co-located with them.

- **Collaboration Tool Use Comfort:** When asked if they were comfortable using the information technology and tools, most engineers reported that they were. Company B does also have and uses videoconferencing. One engineer that has partaken in a few videoconference meetings reports that they are great for getting to know what each team member looks like and to build rapport; however, they can be uncomfortable and often take a great deal of time to set-up. The lack of comfort and ease-of-use with videoconferencing is unfortunate. A study summarized by Harvey and Koubek found that the performance and product of collaborative design teams using high quality video was just as good as that of face-to-face teams.[23] (Refer to Section 2.6.3.)

5.6.2 Potential Social and Cultural Collaboration Experience Issues

Despite the fact that all SE personnel interviewed are satisfied working in the current distributed, collaborative work environment, there are some cultural and social issues that they have experienced. These issues are summarized in the following list:

- **Program B Culture Challenge:** For the upgrade to Product B, some of the partnerships, processes, and design philosophies have changed over the past several years. As described in Section 5.6.1, Program B has been around for so long, that it essentially has its own cultural. For the most part, this is a great benefit to the program, however, because it has been around so long, there are some cultural issues that have not adapted with the times. For example, one SE leader explains that there are several occurrences of the "Not Invented Here" syndrome, as engineers challenge the updates.

- **Company Culture/Business Challenges:** As many different companies are coming together to work on Program B, there is a mixing of companies cultures, processes, and business practices. Some engineers report that there have been several instances when other development teams assert, "This isn't how we do business," or symptoms of the "Not Invented Here" syndrome.

- **Engineers Experience Sensory Overload:** It is quite obvious that working in a CDSE environment is challenging, and several engineers report having to juggle many tasks, all the while the phone is ringing off the hook and there is a bunch of issue "fire-fighting" going on, which makes it difficult to keep sight of what is important.

- **Trust:** It is difficult for engineers to completely understand or trust their coworkers at other locations or companies, as described in Section 5.6.1. Development seems to take longer due to trust issues, since engineers spend time "double-checking" or verifying their coworkers deliverables and also "double-checking" or verifying their personal work before sharing it with their coworkers, as they do not want to be embarrassed. Another engineer relates that to combat mistrust, teams may devote resources to duplicating the efforts of their coworkers for fear they will not receive what they need when they need it from other teams. (Which is clearly a waste of precious resources.)

- **Company Philosophy Crashes:** The SE leadership team described some instances of company "philosophy" crashes. For example, different management structures and decision making cultures within companies sometimes make it difficult to work on the engineer to engineer level (e.g. needing management permission before acting). Another example is the inflexible nature of some companies, whereas other companies strive for flexibility.

- **Generational Differences:** One engineer explains that based on his experiences, there is often a discrepancy between the older and younger generations and their willingness to use the tools and electronics to collaborate. Also in some instances, one engineer explains that the older population of engineers believes knowledge is power, and therefore they prefer to keep some program information and rationales to themselves.

- **Responsibility of Off-Site Engineers' Job Satisfaction:** One interesting challenge reported by an interviewee was the additional responsibility to ensure that the employees he leads at other locations and companies have executable plans and are satisfied in their jobs, etc, even though they are technically employed by other companies.

- **Communication is Difficult:** Communication on a program of this size is in general very difficult, and distributed development sites have complicated the communication issue.

- **Organizational Obstacles:** Although most of the product-organizational related issues have been worked out, it was difficult at program start-up to collaborate at the engineer to engineer level across subsystems and interfaces.

- **Lack of Brainstorming:** An engineer relates that in this CDSE environment there is not as much brainstorming, as meeting time is mostly comprised of an engineer describing work that he/she has completed for team review. Rarely do engineers sit around the table and brainstorm better or other ways of doing things, as meeting time is scarce. These issues are likely not just due to collaboration, but also the large size of the program.

- **Language and Interpretation Issues:** Some engineers report experiencing language and interpretation issues, especially for tasks related to the interface definition and design (since it involves several of the different companies and cultures).

5.6.3 Social and Cultural Collaboration Experience Success Factors

Although there were some social and cultural issues noted by the Company B engineers, SE leaders, and support staff, the long Program B history has overtime self-corrected many of the issues that arise on shorter-term products. For example, issues of trust, company culture differences, and interpretation differences have been remedied by repeated interactions and the ability to build relationships over time. Despite the long program history, the Company B and Program B management has developed some key tactics to improve social and cultural interactions on Program B.

- **Create Team Glossary and Acronym Dictionary:** To combat issues of company or culture interpretation differences, a Program B team glossary has been developed to explicitly define all key terms and acronyms. All sites have access to the glossary.

- **Well-established Social Events:** One engineer describes that Program B has well-established social events. There are several team social outings every year, and the Program B management office and the customer have often recognized employee good work by throwing celebratory parties.

- **Dependency Matrix in Use:** To help overcome issues of trust and engineer reliability, a "dependency matrix" is in use. This matrix relates the dependencies between people and deliverables on the program. Since this matrix is viewed by top management across the companies, it incentivizes those who make agreements or who owe deliverables to perform the required task by the specified time.

5.7 CDSE Benefits and Motivation

All interviewees were asked what they believed were the benefits of and the motivations behind the CDSE efforts. The following bullets summarize their responses to what they believe are the CDSE benefits:

- Less travel (Or a lot less travel than there otherwise would have been without the addition of the collaboration tools).

- Although it may sometimes be inconvenient when there are pressing issues to address, the time difference allows the east coast teams to work "un-interrupted" by the west coast teams in the morning.

- The CDSE environment and the travel that sometimes goes along with it can be a boost to one's social life! When visiting off-site locations, you get to meet new people and go out to places/locations you wouldn't otherwise go - most of the time accompanied by the "tour guide" engineers who actually live there.

- The environment is highly beneficial in that it allows for several very different perspectives and proposed solutions to issues by having different companies and cultures work together.
- Cross-pollination of ideas across companies and cultures.
- Because employees can still live where they want, there is a larger pool of applicants, and therefore the program gets better qualified and happier engineers.
- The CDSE will result in a better product overall.
- This program is using the best people from everywhere for the betterment of the program.
- Resources are more widely available and can be better allocated and allocated more quickly.
- Exposure to other company cultures and operations. Company B engineers find it very interesting to see the different ways things are done elsewhere.
- The many perspectives offered from multiple companies.
- The collaboration tools allow real-time peer reviews.

Many of the motivations for the choice of CDSE environment can be summarized by the benefits described above. However, some additional responses from interviewees include:

- The motivation is for distributed collaboration is driven by the customer.
- Company B alone does not have all of the expertise or resources to complete the program alone - therefore it is necessary to collaborate.

Chapter 6

CDSE Case Study Analysis

As described in in Section 3.8, the research completed for this thesis has several limitations, and therefore there is not a large enough or diverse enough sample size to conclude with any CDSE theoretical findings. As explained in Chapter 3, the intent of this exploratory research is to document the current state of several United States aerospace and defense industries' CDSE practices, pinpoint issues where additional future research is needed, to record the CDSE lessons learned to date, and to identify indications of CDSE best practices. Consequently, to frame the discussion that follows, this chapter provides a comparison of some of the similarities and differences of the two case studies completed. The chapter further summarizes the key CDSE issues identified as well as the lessons learned and success factors concerning a variety of topics from both case studies, including: the collaboration situation and management; collaboration tool use; SE knowledge, data and decision management; the SE processes and practices; and the CDSE social and cultural environment. The chapter concludes with a consolidated summary of the interviewee provided CDSE benefits and motivations.

Note throughout this chapter "A and B" refers to Company/Program A and Company/Program B.

Note also, throughout this chapter tables are used to represent the consolidated data from A and B. In many of the tables, three columns labeled "A," "B," and "A and B" appear. A "*" is placed in each column corresponding to where the concept(issue, success factor, benefit, etc.) originated, Company A, Company B, or both A and B, as described in Section 3.7.

6.1 Comparison of CDSE Case Studies

In order to better understand the summary of issues and success factors discussed in this chapter, this section describes some of the general similarities and differences of the two case studies completed.

In general, both case studies have a very similar collaboration situation, it is thus no wonder why many issues encountered by each program are very similar. Table 6.1 summarizes the key similarities of A and B.

Characteristic	Company/Program A and B Similarities
SE Scenario	Both are performing CDSE across multiple sites and companies.
Customer	Both have same customer: US Government.
Company Role	Both are product prime systems integrator. Both are leading the management and coordination of SE efforts.
Development Timeline	Program timelines are similar (~ ten and twelve years).
Product Type	Both are creating a defense-natured product.
Lifecycle Phase	Both are currently completing detailed design of the product while preparing for integration.
Collaboration Management	Both use standard program management tools and processes, such as EVMS and IMS, team calendars, responsibility matrices, etc. to manage and track program efforts.
Communication Methods	Both use: collaboration tools, email, telephone, teleconferences, face-to-face communications, travel, reviews, weekly meetings.
Time Differences	Both companies work with development sites scattered in other time zones, and time differences have a similar impact on both teams - meetings have to be carefully scheduled and work hours must be shifted to accommodate.
Program Success	Both companies have been successful thus far in their product development, completing major milestones on time and nearly within budget.
Collaboration Tools	Both use collaboration tools extensively to facilitate SE coordination, communication, SE processes and product development.
Collaboration Tool Types	Both programs use many of the same types of tools: email software; database tools for informal and formal program and technical documentation; DOORS for requirements development; teleconference systems; Sametime; Netmeeting; trouble reporting systems; and version control software.
Tool Training	Both offer collaboration tool training to program personnel at all sites.
Tool Support	Both have a variety of program tool support engineers to facilitate tool use, development and integration.
Tool Usefulness	All interviewed personnel on both programs find tools very useful.
Knowledge and Data Storage	Both have learned that knowledge, data, and artifacts for the program must be stored in a central database.
Knowledge and Data Management and Tools	Both companies support knowledge and data management with the use of collaboration tools, including formal and informal databases, email, teleconferences, etc.
Disseminating Information	Both use email, meetings, reviews, and the collaboration tools to disseminate information.
Configuration Control	Both have processes in place to formally control documentation and program artifacts.
Decision Making	Decisions are not formally controlled on either program. Both programs foster decision making at the lowest level possible, unless decisions involve different budgets or impact other system components.
SE Processes	Both have formal, standard SE processes in place to govern SE development. All development companies and sites are under contractual obligation to follow the processes.
Job Satisfaction	All interviewees at both companies are satisfied with their jobs.

Table 6.1: Summary of general A and B similarities.

However, A and B have different development approaches, process maturities, and program histories, all of which affect the issues they encounter, the lessons they have learned, and the success factors they have developed to overcome barriers. Table 6.2 summarizes the key differences between A and B.

Characteristic	Company/Program A and B Differences
Contract Set-up	Company B Prime Contractor on Program B; Customer Prime Contractor on Program A
Program Size	Program A is ~6 times larger than the size of Program B (considering the number of personnel on the program employed at each of the companies).
Company Size	The defense business unit of Company A is ~13 times greater than the total company size of Company B.
Collaboration Size	There are many more companies and sites involved in the development of Product A, compared to Product B.
Product Development Phase	Program A has successfully completed CDR and is finishing detailed design; Program B has successfully completed PDR and is preparing for CDR.
Product Development	Product A involves the design and development of a complete, new system; Product B development is an update of a legacy subsystem, which is a major component of a legacy defense system currently in use.
Travel	Program B personnel travel often; Program A personnel travel seldom, relatively speaking.
Company Residents	Company B often has other company "residents" located on-site; Company A has a collaboration center near the customer where company representatives work together.
Classified Communications	Program A communications often involve classified media, including a separate classified network; whereas Program B communications are mostly unclassified.
Process Improvement Projects	Company A supports and relies on process improvement projects to improve performance; Company B does not have a formal process improvement program currently in place.
Process Maturity	Company A has more mature SE processes, as rated by CMMI (level 4); Company B is working toward achieving its first CMMI certification (Level 3).
Network Infrastructure	Company B is responsible for their network infrastructure and maintenance; Company A outsources their network infrastructure and maintenance (resulting in different methods to access program data: VPN).
Requirements Management	Company A uses the DOORS tool for requirements management (i.e. requirements formal configuration control); Company B uses non-specific document formats for formal requirements management.
DOORS Use	Company B uses DOORS predominantly for requirements development and traceability; Company A uses DOORS and the DXL extension for configuration management, metric collection, artifact consistency, and automated requirements maintenance.
Database Searchability	Program B databases have more mature search capabilities; many Program A databases do not have mature search capabilities.
System Architecture	Program B has a modular product system architecture with common processes; Program A has an integrated system architecture with distributed processes.
SE Development Tools	Program B coordinates the simulations and SE analysis tools at a program level; Program A allows sites/developers freedom to use different development tools for analysis and simulations.
Program History	Program A is relatively new, and many companies and sites are working together for the first time; whereas Program B has over a 50 year history and many companies and sites have been working together for many years.
Face-to-Face Meetings	All Program B interviewees have met face-to-face with all engineers they work with at all locations; Program A interviewees have met only some of the engineers they work with face-to-face.

Table 6.2: Summary of general A and B differences.

6.2 Compilation of Experienced SE Collaboration Differences compared to Traditional SE Environment

Throughout the interviews conducted for the two case studies, the engineers and leaders described several key differences between working in a collaborative, distributed SE environment and working in a traditional SE environment. For a more detailed discussion of environmental differences, refer to Section 4.2.4, which summarizes several key differences as related from interviewees in Company A, and Section 5.2.3, which summarizes several key differences as described by interviewees from Company B. Table 6.3 summarizes the consolidated key environmental differences between CDSE and Traditional SE.

	CDSE vs. Traditional SE Environment	A	B	A and B
1	There is a great deal more "up-front" work to coordinate SE efforts, teams, resources, etc.			X
2	Communications in a CDSE environment are in general more difficult and facilitated by the introduction of and reliance on collaboration tools.			X
3	CDSE meetings are more formal, thus there is less brainstorming and social interactions amongst teams.			X
4	New and different processes are standardized, mandated and followed.			X
5	There are additional obstacles and complexities: company proprietary data sharing, corporate fire-walls, non-disclosure agreements, classified data transfer.			X
6	Untraditional organizational channels are used to enforce all developers to use the agreed upon processes.			X
7	Centrally collected raw data metrics are used to measure relative company performance.	X		
8	New and different SE management positions are created to coordinate efforts.	X		
9	Collaboration creates a "one team" or "one goal" work arrangement, where all contractors are working toward the same final, integrated product.		X	
10	It is more difficult to allocate or re-allocate resources as changes occur, since formal contracts with schedules and resource allocations are typically done way in advance of program execution.		X	
11	There are more discussions and more frequent interactions among teams.		X	

Table 6.3: Summary of CDSE vs. traditional SE environmental differences.

6.3 CDSE Barriers and Issues Encountered

CDSE issues and barriers were discussed at length with interviewees on a variety of topics. This section provides a consolidated summary of the key issues discussed with interviewees from A and B on: the collaboration situation and management; collaboration tools; knowledge, data, and decision management; SE processes and practices; and the CDSE social and cultural environment. Note that there is not a large enough sample to support that the issues discussed herein with interviewees are widespread program issues. The issues discussed in this thesis are indicators of current possible issues and potential future issues on other programs, and their further examination is an are of future research.

Note, where provided, interviewee proposed recommendations to specific issues are given throughout the tables in this section.

Also note that many of the issues and barriers discussed by interviewees are consistent with the CDSE issues and barriers discovered in the literature review. Refer to Section 2.9 to review a summary of the proposed CDSE issues and barriers discussed in the literature.

6.3.1 Collaboration Situation and Management Issues

Several of the key collaboration situation and management issues discussed with interviewees are common to both programs, likely due to the very similar collaboration situation faced by both A and B. Table 6.4 summarizes the consolidated collaboration situation and management issues from A and B.

Please refer to Section 4.2, which describes collaboration situation and management issues as related from interviewees in Company A, and Section 5.2, which summarizes collaboration situation and management issues as related from interviewees in Company B.

Table 6.4: Summary of consolidated collaboration situation and management issues from A and B.

	Collaboration Situation and Management Issues and Potential Barriers	A	B	A and B
1	It is difficult to get all team members and development sites on board with team-wide/management created policies and agreements.			X
2	The CDSE programs are not short-lived; the SE infrastructure, organization, processes and tools must be able to support long development lifecycles.			X
3	For long contracts, the contractor and contractor relationships may change. Oftentimes poor precedents or tool selections started by one company, must often be continued in a later phase			X
	Interviewee Recommendation: Additional ``upfront" work may help to establish universal tools that will be useful for all teams over the program lifecycle.			
4	People at other sites are sometimes difficult to locate and time is wasted tracking them down.			X
5	There is a general lack of previous experience on how to estimate cost and schedule for CDSE. Management tends to underestimate the costs and schedule slips associated with collaboration. There is a learning curve.			X
	Interviewee Recommendation: Document current CDSE program data, lessons learned, and successes for use on future projects.			
6	Time differences between collaborating sites makes it difficult to schedule meetings, particularly long ones.			X
	Interviewee Recommendation: Schedule daily short meetings at a set time to review status, progress, and issues during core business hours.			
7	Additional time is spent on collaboration-related activities, such as tracking down people, transferring data, preparing tools, disseminating information, setting up meetings, and answering emails.			X
8	It is more difficult to allocate or re-allocate resources as changes occur, since formal contracts with schedules and resource allocations are typically done way in advance of program execution.		X	

6.3.2 Collaboration Tool Use Issues

Most interviewees related that they were very satisfied with the collaboration tools used on A and B. However, several potential and engineer-experienced collaboration tool issues were discussed during interviews. Table 6.5 summarizes the consolidated collaboration tool issues from A and B.

Please refer to Section 4.3.2, which describes collaboration tool issues as related from interviewees in Company A, and Section 5.3.2, which summarizes collaboration tool issues as related from interviewees in Company B.

Table 6.5: Summary of consolidated collaboration tool issues from A and B.

	Collaboration Tool Issues and Potential Barriers	A	B	A and B
1	Not all sites use or have access to the same tools or versions of tools, resulting in compatibility issues. Interviewee Recommendation: Make the tools more widely available as a way to improve collaboration by improving the account request process.			X
2	Not all development facilities have access to or are accustomed to using classified data systems, resulting in delays and potential data contamination as information is transferred from classified to unclassified systems.			X
3	Several collaborative tools have steep learning curves. Interviewee Recommendation: Tailor engineer tool training to teach only the necessary tool functionality to complete job tasks.			X
4	Collaborative tools and networks are often unreliable, unstable, and prone to crash.			X
5	To use all the different collaboration and SE development tools require that engineers and other users have several different accounts, passwords, and privileges.			X
6	Many different versions of email software is used across different distribution sites. Therefore, address book and calendar functions, which would greatly aid communications and meeting support, are unavailable.			X
7	Establishing classified data networks is a difficult task - not all sites have access or equipment to support encryption, including the customer. (Necessitating the exchange of classified data through different means: courier service, US mail, etc.)			X
8	There are significant delays to obtain accounts for collaboration tools.	X		
9	Data often needs to be transferred between classified and unclassified systems and internal and external company systems, resulting in delays and wasted time. Interviewee Recommendation: In an effort to speed up the inevitable data transfer, place the two different systems side-by-side to facilitate transfer by minimizing delays and transportation time.	X		
10	The processes governing tool usage are sometimes ambiguous or non-existent. Therefore, artifact development may not be consistent.	X		
11	Internet connectivity and a protected network are necessary for most CDSE work, making it difficult to work in several traditional places (while traveling, airplanes, at home, etc.)	X		
12	Getting all companies the same tools and tool versions was not an easy task - there are 3-4 month delays and litigation issues as companies determine tool management responsibilities and tool use guidelines. Interviewee Recommendation: Have a central point of contact at each company when dealing with collaboration tools.		X	
13	All collaboration and development tools are not being used to their full potential, resulting in the necessity for engineers to perform additional work to get specifications into the correct form for configuration control.		X	
14	There are ambiguities as to whether one company is responsible for the maintenance and support of the SE and collaboration tools being used program-wide, since tools are typically purchased and managed by the individual companies.		X	

Note, specific DOORS related issues are described in Section 4.3.2, and have not been included in the consolidated data in Table 6.5.

6.3.3 Knowledge, Data and Decision Making Issues

Several issues have arisen for knowledge, data and decision management, predominantly due to the size and distributed nature of A and B. Table 6.6 summarizes the consolidated knowledge, data and decision making issues from A and B.

Please refer to Section 4.4.3, which describes knowledge, data and decision making issues as related from interviewees in Company A, and Section 5.4.3, which summarizes knowledge, data and decision making issues as related from interviewees in Company B.

Table 6.6: Summary of consolidated knowledge, data and decision making issues from A and B.

	Knowledge, Data and Decision Management Issues and Potential Barriers	A	B	A and B
1	CDSE decision making is slower and difficult. It is difficult to record, track and disseminate decisions, as they are often informally recorded.			X
2	Establishing and enforcing boundaries between the different industries and companies for who controls, specifies, completes and documents work.			X
3	There is not enough time and resources to thoroughly document knowledge, decisions, decision rationales, design rationales, change rationales, etc. Interviewee Recommendation: Management enforcement of the processes in place - make documentation a high priority.			X
4	It is very difficult to disseminate information, changes, decisions, etc to all personnel at all locations in a timely manner. Interviewee Recommendation: A well-defined responsibility matrix helps to know who must be informed of changes, etc. and who must weigh-in on decisions.			X
5	The free flow of data, information, and especially knowledge, between companies and locations is impeded by company proprietary data, resulting in delays, tensions, and even mistrust.			X
6	There are currently no easy-to-use, readily available tools to document meetings, meeting minutes, and actions for virtual meetings. Interviewee Recommendation: Create a meeting minute template and have management enforcement of the process.			X
7	Recording meeting attendance during virtual meetings is difficult and needed for accountability. Interviewee Recommendation: Developing a new tool or enhancing an exiting tool (such as the teleconference system) to keep meeting attendance.			X
8	Collaboration tools and databases are difficult to navigate, resulting in user frustration, lack of use, and wasted time searching for information. Interviewee Recommendation: Better database search tools are needed to navigate the large amounts of data available on collaborative tools.			X
9	A great deal of program information and tool "fixes" are tacit and person-specific. If these key personnel leave/retire, that information will be lost. Interviewee Recommendation: Capture the "fixes" and knowledge in explicit form for use later in this program or on other programs.			X
10	With development of systems and software being completed collaboratively, what products are/can be labeled company proprietary? Interviewee Recommendation: It is very important early on to determine what information and products will be proprietary from the start of the program.			X
11	The free flow of data, information, and knowledge between companies and locations is impeded by data classification, resulting in delays, misunderstandings, and lack of data.	X		
12	Configuration control of requirements process does not fully utilize development tools, is not optimal, and results in a lot of additional work on the part of engineers.		X	
13	It is difficult to maintain artifact consistency and reduce repetition between artifacts. Interviewee Recommendation: Create and enforce processes to make artifacts consistent.		X	

Note that several of the knowledge, data, and decision making issues brought up by interviewees are consistent with the collaborative, distributed knowledge management literature. It is interesting to note that Fagerstrom and Olsson found that: 1) "Engineers spend as much of 30% of their time searching for and accessing engineering design information;" and 2) The KM support tools that currently exist for multidisciplinary product design are directed primarily toward the storage and exchange of explicit knowledge concerning processes or projects."[18] The literature findings are consistent with the A and B issues that suggest tacit knowledge is not being adequately captured (Item #9 in Table 6.6) and that engineers are spending a great deal of their time searching for information (Item #8 in Table 6.6). (Refer to Section 2.6.2.1.)

One interviewee recommendation suggests the creation of a "responsibility matrix" (Item #4 in Table 6.6) to clearly define those personnel who must be informed of changes, etc. and who must weigh-in on decisions. This concept is very similar in nature to the "decision rights matrix" approach used at Duke Power to assist with decision responsibilities.[19] As decision making and dissemination is a widespread CDSE issue, it is a recommended success factor that CDSE programs formulate and widely distribute a "decision rights matrix," thereby requiring that key personnel weigh-in and be informed of decisions. (Refer to Section 2.6.2.2.)

6.3.4 SE Process and Practice Issues as they Relate to Collaboration

There are relatively few SE process and practice issues, since interviewees could not discuss the specific technical process and product issues encountered due to the proprietary, classification, or export controlled nature of SE specific technical data. Table 6.7 summarizes the SE process and practice issues from A and B.

Please refer to Section 4.5.1, which gives an overview of the SE processes and practices in use in Company A, and Section 5.5.1, which summarizes SE processes and practices in Company B.

Table 6.7: Summary of consolidated SE process and practice issues from A and B.

	SE Product and Process Related Issues and Potential Barriers	A	B	A and B
1	Product integrity may sometimes be negatively impacted due to lack of total system visibility and different company priorities.			X
2	There are differences in company design philosophies that make development more difficult, such as requirements traceability and the level of design specification. Interviewee Recommendation: Establish "good requirements" guidelines to facilitate all developers having the same concept of what the requirements should contain.	X		
3	System architecture mirrors that of program and contract organization, thereby not taking advantage of possible common service architectures.	X		
4	One drawback to the modular design approach is that design changes may not be independently contained within a module.		X	

6.3.5 Social and Cultural Collaboration Experience Issues

Despite the fact that all interviewees expressed their contentment and satisfaction with working in the current CDSE environment, several cultural and social issues were discussed. Table 6.8 summarizes the social and cultural collaboration experience issues from A and B.

Please refer to Section 4.6.2, which describes social and cultural collaboration experience issues as related from interviewees in Company A, and Section 5.6.2, which summarizes social and cultural collaboration experience issues as related from interviewees in Company B.

Table 6.8: Summary of consolidated social and cultural collaboration experience issues from A and B.

	Social and Cultural Issues and Potential Barriers	A	B	A and B
1	There are tensions and differences in the way different companies and cultures do business, as ``it is hard to see eye-to-eye." Some engineers report that there have been instances when other development teams assert, ``This isn't how we do business."			X
2	Working in a CDSE environment often requires that developers juggle many tasks, sometimes resulting in sensory overload.			X
3	It is difficult for some to trust CDSE teammates and coworkers at other companies due to lack of visibility into system development (proprietary data sharing restrictions), sometimes resulting in duplicated efforts, and delayed decision making.			X
4	Different management structures and decision making cultures within companies sometimes make it difficult to work on the engineer-to-engineer level. Interviewee Recommendation: After several years of experience and relationship building, it is now easier to communicate with and work with the right people from different organizations. Perhaps these experiences lend themselves to the need for additional initial organizational planning and team-building meetings.			X
5	There is sometimes a discrepancy between the older and younger generations and their willingness to use the tools and technologies available to collaborate.			X
6	Communication is in general more difficult and requires additional effort to overcome poor connections, lack of visual communication feedback, and time differences.			X
7	There appears to be less brainstorming and informal meeting discussions as CDSE meetings are often formally scheduled with strict agendas and limited meeting time available.			X
8	Some engineers report experiencing language and interpretation issues between companies and development sites, resulting in misunderstandings and rework.			X
9	There are ``some things you just can't do" in a CDSE (such as having impromptu meetings or walking down the hall to ask a question).	X		
10	There are cultural differences between different geographical regions (north/south) and the different cultural attitudes results in different ideas about accountability and responsiveness. Interviewee Recommendation: Have social gatherings or team building sessions to improve collaboration relationships	X		
11	Sometimes engineers are isolated, being the only immediate team member working at a development location. Social interaction is necessary for relationship building and employee happiness. Interviewee Recommendation: Have social gatherings or team building sessions to improve collaboration relationships	X		
12	Relationship building itself has been brought up as being more difficult in this CDSE environment, which is a big issue since relationship building is necessary to establish trust in co-workers. Interviewee Recommendation: Have social gatherings or team building sessions to improve collaboration relationships	X		
13	One interesting challenge is the additional responsibility for some managers to ensure that the employees at other locations and companies have executable plans and are satisfied in their jobs, even though they are technically employed by other companies.		X	

Note that the interviewee expressed feeling of a more formal social environment is consistent with the findings of Harvey, Koubek and Hammond. They concluded that distributed communication consists of: "Fewer messages, with greater task orientation and less spontaneity..."[20]. (Refer to Section 2.6.1.1.)

On the contrary to the findings of Harvey, Koubek and Hammond , some interviewees felt there was less brainstorming in ; whereas the authors found that distributed design teams tend to consider more alternatives with a greater degree of clarity than co-located teams.[20] Perhaps more alternatives are considered, but not in the traditional "brainstorming" sense. (Refer to Section 2.6.1.1.)

6.4 CDSE Lessons Learned and Success Factors

All interviewees were asked to provide information about lessons they have learned, success factors they have experienced, and recommendations for improvement. This section provides a consolidated summary of the key lessons learned and developed success factors discussed with interviewees from A and B on: the collaboration situation and management; collaboration tools; knowledge, data, and decision management; SE processes and practices; and the CDSE social and cultural environment. Note that in this analysis, all lessons learned are considered success factors, since it is an achievement to avoid making the same mistake twice. (Note that this consideration is consistent with Huang's "Framework to Evaluate Collaborative Systems for Distributed Collaboration."[24])

Note that there is not a large enough sample to support that the lessons learned and success factors described herein are successful in all CDSE contexts. They are believed to be indications of widely applicable CDSE success factors and early indications of CDSE "best practices." Further examination of these indications is a key area of future research.

Also note that many of the success factors and lessons learned by interviewees are consistent with the proposed CDSE success factors discovered in the literature review. Refer to Section 2.8 to review a summary of the proposed CDSE success factors discussed in the literature.

6.4.1 Collaboration Situation and Management Success Factors

With time and experience, several success factors have been developed to coordinate CDSE. Table 6.9 summarizes the collaboration situation and management success factors from A and B.

Please refer to Section 4.2.3, which describes the collaboration situation and management success factors as related from interviewees in Company A, and Section 5.2.2, which summarizes the collaboration situation and management success factors as related from interviewees in Company B.

Table 6.9: Summary of consolidated collaboration situation and management success factors from A and B.

	Collaboration Situation and Management Lessons and Success Factors	A	B	A and B
1	Share "in-house" developed tools with other sites and mandated their use. Tools were therefore tailorable, the learning curve is lessesned for some, and there is an established tool support system to ensure smooth operation.			X
2	Allow each company/site to have input into standardized processes and development plans, creating management buy-in and company accountability.			X
3	Contractually obligate all teams to follow all SE process, practice and development documentation.			X
4	Provide docushares and other online "firewall protected" collaboration tools to share information and maintain configuration control. These tools should be available and used at all sites and provide both classified and unclassified data storage and transfer.			X
5	Write formal development plans and standardized document templates with buy-in from all key stakeholders.			X
6	Up-front work is necessary, including: defining the management role, defining SE processes and tools, understanding the contract (scope), determining the resources needed for collaboration, setting out the key milestones, determining the metrics, and defining the responsible parties.			X
7	Implement the lessons learned/recommendations collected from previous program phases.			X
8	Measure and track program progress through the use of formal processes and tools, such as EVMS and an IMS, and selected key metrics.			X
9	Hold meetings weekly and monthly at all levels of the SE organizations to disseminate information, discuss issues, and coordinate efforts.			X
10	Conduct a short meeting daily between the east coast and west coast management teams to touch base and to address any immediate issues that may have transpired since the previous day (A work around for time differences and the distributed-nature of the program.)			X
11	Closely monitor key system metrics to determine performance of each team. Provide metric alignment and traceability so teams understand importance.	X		
12	Create a "responsibility matrix" for deliverables and system components, which explicitly calls out the responsible personnel for decision making authority and who must be informed of all decisions or changes to the system.	X		
13	Create a collaboration center near the customer, thereby increasing customer satisfaction and participation, allowing barriers to be "let down," and encouraging a "one team" mentality.	X		
14	Create additional overarching SE positions to coordinate SE work, processes, and artifacts across companies.	X		
15	Dedicate resources to SE collaboration: including tools, tool support, tool training, teambuilding sessions, etc.	X		
16	Ensure that collaboration resources that are needed to support SE efforts should be under the control of SE management (both people and money) . (These resources were problematic when not directly controlled by SE - misalignment of priorities/funds.)	X		
17	Hold "process days" to assist with dissemination of SE information and SE processes.	X		
18	Identify key SME's to weigh-in on decisions, to provide additional support and information to the development teams, and to ensure consistency in the system with the customer's needs.	X		
19	Include the customer as a key team member - they can weigh in on decisions, keep teams focused, and arbitrate. Customer participation and integration into work teams is integral to maintaining customer (not company) driven focus.	X		
20	Make tool/collaboration training a trackable metric to ensure it is a priority and show management backing.	X		
21	Require 100% attendance by all SE team leads (if a lead cannot attend, he/she must send a representative) in order to have accountability and equal representation at meetings.	X		
22	Rotate face to face meetings (such as reviews) between companies/locations to allow each company to take an active role and also to share the burden of cost/traveling.	X		
23	Use formal, team-wide process improvement initiatives to improve program performance, better allocate resources, cut costs, and solve problems.	X		
24	Empower task leaders to be responsible for the management, budget, deliverables, and team structure of their task teams. Therefore team organizations are tailored for each task and structured to best meet the specific budgets and deliverables.		X	
25	Use a well-defined program organizational structure to facilitate collaboration, provide accountability, and monitor issues.		X	

6.4.2 Collaboration Tool Use Success Factors

Collaboration tools are used often by A and B and are necessary for almost all facets of SE development. With many users, managers and tool support personnel involved, a great deal of collaboration tool use success factors have emerged. Table 6.10 summarizes the collaboration tool use success factors from A and B.

Please refer to Section 4.3.3, which describes the collaboration tool use success factors as related from interviewees in Company A, and Section 5.3.3, which summarizes the collaboration tool use success factors as related from interviewees in Company B.

Table 6.10: Summary of consolidated collaboration tool use success factors from A and B.

	Collaboration Tool Lessons Learned and Success Factors	A	B	A and B
1	Offer a wide variety of collaboration tool training, outsourcing the training to COTS representatives if necessary. Include online tool courses, lunchtime training sessions, FAQ's and online help guides.			X
2	Create a central data storage location. After experiencing issues with configuration management and maintaining up-to-date information (delays with synchronizing databases), a centralized database system was put in place at both companies.			X
3	Use the simplest tools where possible. (Ex. hands-free headsets, or teleconferences with page-numbered presentation slides.)			X
4	Implement tool lessons learned and suggestions from tool users to improve tools over the course of product development.			X
5	It is very important to determine ahead of time which engineers need what type of access to certain tools, software, requirements documents, etc. to prevent errant modifications.			X
6	Tailor tool training to target specific skills for specific tools.	X		
7	Limit the tool features available to users to create consistent artifacts and efficient development.	X		
8	Establish guidelines to determine when/how tools should be updated. (Just like the data in the tools, the tools themselves can often be altered.)	X		
9	Establish guidelines and processes for tool usage in formal documentation.	X		
10	Management backing of the tools and the processes that support them is essential to their widespread success.	X		
11	Automate where possible. Some automation ideas: tool maintenance, metrics collection, artifact consistency checking, database synchronization, etc.	X		
12	Evaluate tools on smaller programs first before deploying new tools on a large, distributed program to ensure that they meet the needs of the program.		X	
13	Provide dedicated collaboration tool support staff, including having tool coordinators at each site.		X	

6.4.3 Project Knowledge, Data and Decision Management Success Factors

Successful knowledge, data, and decision making practices are key to coordinating SE development in a CDSE environment. Being able to find the information one needs when needed and sharing data, simulations, and designs across development sites in real-time facilitate smooth product development. Table 6.11 summarizes the knowledge, data, and decision making success factors from A and B.

Please refer to Section 4.4.4, which describes the knowledge, data, and decision making success factors as related from interviewees in Company A, and Section 5.4.4, which summarizes the knowledge, data, and decision making success factors as related from interviewees in Company B.

Table 6.11: Summary of consolidated knowledge, data, and decision making success factors from A and B.

	Knowledge, Data and Decision Management Lessons Learned and Success Factors	A	B	A and B
1	Define a team glossary/dictionary to foster uniform interpretation and to minimize misunderstandings.			X
2	Define an intuitive, searchable, central database to store and share all program data, knowledge and decisions.			X
3	Explicitly capture the knowledge, data, and experience from the current and past programs for use on future programs.			X
4	Record and share meeting minutes to document discussions, rationales, and decisions.			X
5	Use document standards and templates to provide artifact consistency.			X
6	Use the accessibility feature of tools to provide an accountability mechanism and track decisions.			X
7	Write meeting minutes, action items, and agreements during meetings in real-time for all to see. Doing so allows engineers to agree to wording and accept accountability.			X
8	Allow each team member to be heard/weigh-in on decisions so their opinion is known and recorded - but only those with authority actually have bearing on decisions.	X		
9	Appoint SMEs for each subsystem and hold them accountable for sharing information with their teams and for weighing in on decisions/changes that effect or impact their product.	X		
10	Define the relationships between subteams and have an arbitrator/coordinator to work out the issues/overlap between each team and ensure that the right people are interfacing.	X		
11	In order to use SME time efficiently, most of the work for meetings must be done outside of the formal meeting time. Formal meetings are thus efficiently run, and there purely to formally record decisions, positions, and data.	X		
12	It is necessary to have a centralized, consistent method to collect and represent SE metrics from all sites.	X		
13	Keep track of and publish meeting attendance, ensuring decision accountability and awareness for a decision is traceable.	X		
14	Tailor the data and metrics collected by tools to see the important trends, new requests, changes, and status.	X		
15	Use the processes and tools to your advantage to create and manage data and knowledge.	X		
16	Use the work environment to facilitate knowledge and data sharing, such as having "team rooms" and collaboration centers.	X		

Fagerstrom and Olsson suggest the following additional success factors found from their case study experiences: 1) The creation of uniform definitions, processes and models is essential to collaboration success in knowledge management; and 2)Have formal meetings to exchange experiences and transfer knowledge in cross-functional teams.[18]. (Refer to Section 2.6.2.1.)

6.4.4 Collaborative SE Processes and Practices Success Factors

As this is a little-explored area of research, the lessons learned and success factors relative to CDSE processes and practices are very important to the CDSE field. Table 6.12 summarizes the collaborative SE process and practice success factors from A and B.

Please refer to Section 4.5.3, which describes the collaborative SE process and practice success factors as related from interviewees in Company A, and Section 5.5.3, which summarizes the collaborative SE process and practice success factors as related from interviewees in Company B.

Table 6.12: Summary of consolidated collaborative SE process and practice success factors from A and B.

	SE Product and Process Related Lessons Learned and Success Factors	A	B	A and B
1	Coordinate SE development efforts with formal SE processes.			X
2	Enforce the common SE development processes.			X
3	Update the collaboration and product development tools to support the SE processes (like configuration management) and system architecture (like the simulation library).			X
4	Use integrated modeling and design approaches, such as EMT, mission threading, and virtual system simulations.			X
5	Be very specific early-on about; vague definitions and requests can result in re-work, poor interface development, and integration issues.			X
6	Ensure that SE development tools, such as simulation software and DOORS databases are widely available to all teams.			X
7	Perform process improvement initiatives to improve SE processes, solve issues, and optimize team performance.	X		
8	Define a process to raise SE issues for SE-related artifacts (processes, design documents, etc.).	X		
9	Define a process to determine how system changes get flowed down to subsystems or contractors.	X		
10	Require strict standardization of both the subsystem and the common processes to facilitate integration and interface development.		X	

6.4.5 Social and Cultural Collaboration Experience Success Factors

Although often not a top program priority, the social and cultural collaboration experiences in a CDSE environment have a big effect on many interviewees. Despite its low priority, several success factors have emerged to improve social and cultural-related interactions on A and B. Table 6.13 summarizes the social and cultural success factors from A and B.

Please refer to Section 4.6.3, which describes the social and cultural success factors as related from interviewees in Company A, and Section 5.6.3, which summarizes the social and cultural success factors as related from interviewees in Company B.

Table 6.13: Summary of consolidated social and cultural success factors from A and B.

	Social and Cultural Lessons Learned and Success Factors	A	B	A and B
1	Create a team glossary and acronym dictionary to help minimize misunderstandings and bring new personnel up to speed more quickly.			X
2	Sponsor social events to foster relationship building and reward teams for good performance.			X
3	Once metric collection methods are verified, rely on SE metrics to indicate trends, team performance, and problem areas that need to be addressed.	X		
4	Make team members more comfortable with tool use by tailoring tool training, lessening the number of tools needed and having a tools specialist on the team to answer questions.	X		
5	Create and enforce ``Team Rules" or guidelines all team members must abide by concerning the treatment of all companies and teammates.	X		
6	Provide team paraphernalia (coffee mugs, t-shirts, pens, etc.) to foster team camaraderie and a "one team" identity.	X		
7	Create a dependency matrix that is visible to the management team relating the dependencies between people and deliverables on the program to help overcome issues of trust.		X	

Creation of a team glossary and acronym dictionary to help minimize misunderstandings is a key success factor consistent with the findings of Fagerstrom and Olsson, who found that a shared communication language is essential if knowledge sharing is to take place efficiently.[18] (Refer to Section 2.6.2.) Further, the findings of Harvey and Koubek relate that in a collaborative, distributed design environment, where engineers and designers from many different disciplines must communicate, a common vocabulary and communication schema is vital to effective communication.[23] (Refer to Section 2.6.3.)

6.5 Compilation of CDSE Interviewee Provided Motivations and Benefits

All interviewees were asked what they believed were the benefits of and the motivations behind the CDSE efforts. The following tables summarize the consolidated responses from interviewees at A and B: Table 6.14 summarizes the interviewee provided CDSE benefits and Table 6.15 summarizes the interviewee provided motivations.

For a more detailed listing of CDSE benefits and motivations as described by interviewees, refer to Section 4.7 for Company A responses and Section 5.7 for Company B responses.

Table 6.14: Consolidated summary of interviewee provided CDSE benefits.

	CDSE Benefits Provided by Interviewees	A	B	A and B
1	Less travel (or a lot less travel than there otherwise would have been without the addition of the collaboration tools).			X
2	Time differences allow the east coast teams to work ``un-interrupted" by the west coast teams during the morning hours.			X
3	Different industrial and experiential backgrounds allows the program to take advantage of the expertise of the national defense industry.			X
4	Team members get to experience diversity in many things: companies, cultures, people, ideas, etc.			X
5	Technology enables engineers to not have to travel, allowing the engineers to save time and remain with their families. It also saves the company and the program time and money.			X
6	By having the collaboration tools in place, impromptu or emergency meetings can be called on short notice; whereas in a traditional environment, an entire day of travel may have been needed to have a face-to-face meeting with the customer, etc.			X
7	Ability to have a challenging and rewarding job position.			X
8	This environment forces us to enforce the processes, standards, and documents.			X
9	There is a greater level of predictability (in people and products), since the processes are well-followed.			X
10	Because employees can still live where they want, there is a larger pool of applicants, and therefore the program gets better qualified and happier engineers.			X
12	Get to work with customer as a partner.	X		
12	Exposure to a broad range of information and personnel. The increased exposure leads to a great deal of interactions (different processes, ideas, practices, cultures, etc.), which overall leads to improvements in SE.	X		
13	Resources are more widely available and can be better allocated and allocated more quickly.		X	

Table 6.15: Consolidated summary of interviewee provided CDSE motivations.

	CDSE Motivations Provided by Interviewees	A	B	A and B
1	All of the potential benefits.			X
2	The motivation for CDSE is customer driven.			X
3	A single company alone does not have all of the expertise or resources to complete the program singlehandedly - therefore it is necessary to collaborate.			X
4	The job gets done to the maximum product capability.	X		
5	By increasing capabilities and experience in this type of defense system, there is the hope to achieve additional future business with this customer.	X		
6	Increase the customer's satisfaction since with collaboration, a better system at the best value can be created.	X		
7	CDSE is a smarter way to do business, allowing programs to be bigger, better and easier to complete. This method is less expensive and the technology that results from it is a huge benefit. Since more work can be done, these types of programs are likely more profitable as well.	X		

In examining the CDSE benefits and motivations related by interviewees, t is interesting to note that interviewee responses encompassed in some way **all** of the proposed benefits and motivations described in the literature examined, as summarized in Section 2.3.1 and Section 2.3.3.

Chapter 7

CDSE Research: Recommended and Proposed Future Work

One of the major aims of this research is to identify future areas of work. The exploratory research summarized in this thesis has touched upon many different facets of CDSE, and there is clearly a wealth of additional research that would be beneficial to this emerging field. This chapter summarizes many of the obvious and emergent areas of future work on the topic of CDSE, including research aimed at addressing the limitations of the case studies described herein. The following bullets summarize several topics and questions brought up by this research and recommended for future investigation:

- **Address Existing CDSE Issues Revealed by Case Studies:**
  First, and foremost, where existing issues have been identified by interviewees, future work is needed to better the current CDSE practices and find solutions to the existing problems experienced by the participating companies. All of the issues related by interviewees in Section 6.3, are real, current issues being experienced on a daily basis by SE personnel. It is likely that the common issues revealed by the case study participants are issues for many other CDSE programs as well. An obvious first step for future work in CDSE is to address these issues.

- **Expand Current Research:**

  As described in Chapter 3, and specifically in Section 3.8, there are several limitations of the research summarized in this thesis, particularly because data for each case study was collected from only one collaborating company, sometimes only within one major subteam of SE, and with predominantly senior engineers and leaders. An obvious area of future research would be to expand the CDSE research sample to include:

  - Perspectives from multiple companies and the customer.
  - Interviews/Input from all levels of systems engineers and support staff (including individual contributors).
  - Interviews/Input from all SE subteams and and SE-related disciplines.
  - Expand the interviewee sample size to include enough data for the execution of statistical analysis of the responses.

  With a large enough sample size and the input of multiple companies, disciplines, and levels of engineers, enough data will be obtained to determine if the proposed lessons learned and success factors discussed in Chapter 6 are indeed best practices.

- **Is there a preferred system architecture to support CDSE?**

  Two different product system architectures were employed by the two CDSE case study teams. Company B developed a modular system architecture supported by common systems; whereas Company A selected an integrated system architecture supported by distributed systems. There was really not enough evidence to suggest that one architecture was better or preferred for a CDSE environment, although both teams mentioned that distributed collaboration was one of the factors considered when selecting the architecture.

- **Is the CDSE environment for everyone?**

  A common theme for future research that surfaced in many of the interviews at both Company A and B, was to determine if there is a certain type of person that is cut out for a CDSE position. As many engineers describe, some people thrive in the CDSE environment, most likely those that really enjoy communicating and being challenged. The CDSE environment really requires somewhat more effort to communicate, a greater amount of teamwork, and a lot of patience, which some people are uncomfortable with. An idea for future research is the investigation of what type of people, if any, are better-suited to work in a CDSE environment. Are there certain personality traits that CDSE engineers need to have to do well? Are there necessary communication skills or level of experience?

- **How is CDSE organization or program performance measured?**
  In general, there are metrics in place to measure program performance (such as cost, schedule, EVMS, etc.). In addition, there are typical SE metrics employed to measure SE team performance, (such as # of TBD's in requirements, the # of action items open, staffing needs, CDR/PDR performance, etc.) We lack metrics to specifically target the performance of CDSE teams. Are the metrics currently in use sufficient to indicate collaboration team performance?

- **Is there a way to determine which aspects of the system must be developed in a traditional environment?**
  A very interesting aspect of the collaboration efforts at Company B was the way the product architecture was assigned and awarded to contractors. Prior to awarding subcontracts, Company B explained that they determined which aspects of the system were very tightly coupled, and should therefore be done in a more "traditional" engineering environment. Therefore work products were assigned based on the system coupling and complexity. How did they decide this? What criteria of a system can be developed to determine when a system should be developed in a "traditional" or CDSE environment?

- **What types of organizations foster successful CDSE? or What Types of Organizations prevent successful CDSE?**
  CDSE team organization surfaced many times in these studies and was stated

to foster CDSE. However, the structures of both teams investigated in this research are different. Do both organizational structures foster CDSE? Is one organizational type better than another in this type of environment?

- **When can a product be developed in a CDSE environment and when must it remain "in house"?**

  Are there characteristics of a team or system that require all SE work to be done "in house." Perhaps the system cannot be broken into defined architectural chunks, due to complexity, classification, or system coupling. Or perhaps the specialized development team organization cannot support distributed collaboration.

- **Does the size of a system, the geographic locations of expertise, or the team size determine when it is best to perform SE more traditionally or distributively?**

  The size of the programs in the two case studies described in this research varied greatly. It was clear from the interviewee responses that varying levels of travel and engineer residency took place. In Company A, where the program is very large and very distributed, relatively less travel occurred and SE was more distributed. In Company B, where the program is relatively smaller and there are fewer development sites, a great deal more travel has taken place and residents from other companies were frequently on location at Company B. One could infer that the SE development on Program B may be more similar to traditional SE development. Therefore, it is important to the structuring of the team and SE processes to understand how the program size, product size, and geographic distribution affects the SE collaboration. At what level/program size is CDSE better than more traditional SE (i.e. - moving all specialized systems engineers to be co-located)?

- **What is the necessary relationship between system architecture and team organization for successful CDSE?**

  The two organizations studied in this thesis had very different team organizational structures; however both organizational structures were closely linked to the system architecture. What types of organizations foster successful CDSE?

- **Do CDSE team organizations need to be different than traditional SE organizations to be successful?**

  The preliminary data from the two case studies indicates that yes, CDSE team organizations need to be different. For example, Company A needed to create overarching SE coordination positions to oversee SE development across distribution sites. Company A also expressed that the customer needed to be a key stakeholder on many subteams. Company B has integrated their partner companies into the team organizational structure. Determining the type of organization and how the organization must be adapted from traditional SE team structures is an important CDSE future research topic.

- **How can systems "engineer to engineer" relationships and communication be improved across different sites and companies?**

  Many engineers expressed their frustration at not being able to successfully establish working relationships at the engineer-to-engineer level due to organizational and management boundaries. Establishing better ways to foster systems engineer communication across organizations and sites will likely foster a better, more integrated final product.

- **Are there ways to expedite relationship building, thus enhancing trust and foster better collaborations?**

  One aspect that was repeatedly brought up by interviewees at Company A was the difficulty in establishing relationships with other engineers working from other sites. The lack of a developed relationship resulted in a lack of trust, and other issues with collaborations. However, the engineers at Company B have worked together for many years, and have had an opportunity to develop relationships and build trust. Are there methods, training, tools, or social events that could be employed to expedite relationship building? One idea is to perhaps train all SE personnel in the processes, language, cultre, company product line, history, etc. of the other companies and engineers they are working with.

- **Creation of an "SE Collaboration Maturity (SECM)" Factor:**
The results from the two case studies indicate that there is a company or program "collaboration maturity" for many of the key CDSE topics analyzed (collaboration tool use, process, organization, culture, knowledge and data management, training, decision making, etc.) that can either foster or impede CDSE. The differences in maturity is evident by comparing the issues and success factors/lessons learned for each CDSE key topic for each case study. Similar to CMMI, I propose future work towards creation of SECM appraisal criteria, to include the maturity and integration of:

  - **Collaboration Situation:** Do engineers need to travel often? What is the percentage of "traveling residents" on the program? How often are face-to-face meetings held? What percent of budget is devoted to travel? How is risk managed? Are there metrics in place to measure collaboration success?, etc.
  - **Collaboration Tools:** Are the tools integrated? Are they supported by SE processes? Are they useful? Do they support team communications? Are there personnel to support their use? Do all sites have them? Do they use the same versions of tools at all locations?, etc.
  - **SE Processes:** Are SE processes formalized/codified? Are they agreed to by all stakeholders? Are they enforced? Are they configuration controlled? Do all sites follow the same processes? Are the processes successful? Are process improvement initiatives supported?, etc.
  - **Knowledge and Data Management:** Are knowledge and data formally managed? Are there tools in place to facilitate knowledge and data management? Are databases searchable? Are meeting minutes formally captured? Are there configuration control processes in place? Are design rationales formally captured? Are there document templates?, etc.

- **Decision Making:** Are decisions formally managed? Are decisions recorded? Are decisions widely disseminated? Is there a responsibility matrix to define roles and boundaries?, etc.
- **Culture:** Is the culture integrated across sites? Do cultural misunderstandings occur? Are there language interpretation issues? Is there a team glossary? Is there a "one team" mentality?, etc.
- **Social:** Are employees satisfied? Are there trust issues between companies or sites? Is line management an issue? Are there team social events? Are employees across companies and sites recognized for good work? Are employees incentivized to work together?, etc.
- **Organization:** Does the organization support SE activities? Are all sites and companies integrated and/or represented by the organization? Are there mechanisms in place to raise issues? Can engineer-to-engineer communications take place?, etc.
- **Training:**Is there training offered to support collaboration tool use? Is training tailored to specific program/organization functions? Is there training offered to support collaboration (team building, etc.)? Is training enforced by management, etc.

Calculated SECM factors can be used similarly to CMMI ratings. For example, contracts could be awarded to those who have an SECM (arbitrary rating) of "X", to ensure that there are the proper mechanisms in place to support successful CDSE activities.

Chapter 8

Summary and Conclusions

This chapter presents a summary of the research presented herein and concludes with several key CDSE research themes.

8.1 CDSE Research Summary

Companies in the aerospace and defense industry have been working collaboratively and distributedly to develop systems for many years, often traveling, relocating employees and using the few collaboration tools that were available. Now enabled by advances in information technology, and motivated by a dynamic global environment, the complexities of today's aerospace and defense systems, and increasingly limited resources (shrinking budgets and a lack of experience systems engineers), CDSE is an emerging practice in need of research and examination to develop successful practices.

When first reading the hypothetical CDSE meeting presented in Section 1.1, I am sure the reader was skeptical, and believed the meeting to be a gross over-exaggeration. Although exaggerated, the issues faced by SE personnel in a CDSE environment are real, and must be addressed. Due to the increasing complexity of systems and the increasingly limited United States defense resources, CDSE will likely become the future and the norm of aerospace and defense industry engineering.

With these motivations in mind, this research set out to document the current state of CDSE practices in industry, pinpoint issues where additional future research is needed, to record the CDSE lessons learned to date, and to identify indications of CDSE best practices. The research questions are summarized for reader convenience in Figure 8-1.

CDSE Research Questions

1. **How can distributed enterprises successfully collaborate to perform systems engineering?**
   - Use IT and Collaboration Tools
   - Schedule and Conduct Meetings
   - Communicate
   - Train Engineers
   - Overcome Social and Cultural Differences
   - Make Decisions
   - Adapt the Product
   - Overcome Issues and Barriers
   - Determine or Measure CDSE Benefits
   - Manage Knowledge and Data
   - Coordinate Processes
2. **What lessons can be learned and success factors developed from enterprises currently performing CDSE?**
3. **What are the key areas for future CDSE research based on exploratory research?**

Figure 8-1: Summary of the CDSE research questions addressed by this thesis.

To address these research questions, two case studies were carried out at two United States aerospace and defense companies. Semi-structured interviews were completed with 21 SE personnel, including systems engineers, SE and program managers, and SE support personnel (such as process or tool experts). Chapter 3 describes in detail how data was collected and analyzed for this study.

Chapter 4 and Chapter 5 directly address Research Question 1, by summarizing how two companies have successfully performed CDSE practices relative to a variety of key topics. (See Figure 8-1.)

Chapter 6 directly addresses Research Question 2, by presenting the issues encountered, lessons learned, and success factors developed by two enterprises performing CDSE. (See Figure 8-1.)

Chapter 7 directly addresses Research Question 3, by summarizing many critical topics and questions for future work in the field of CDSE. (See Figure 8-1.)

Despite several research limitations, this research has explored the current CDSE practices in industry, summarized key issues plaguing current CDSE activities, and has documented several CDSE lessons learned and success factors which are soft indicators of CDSE "best practices." (There are several limitations of this research, due to the relatively small sample size and restricted access to data. Research limitations are summarized in Section 3.8.)

As demonstrated by the lengthy "Recommended Future Work" list in Chapter 7, CDSE is an emerging practice, and there is still a great deal of CDSE research to be done.

8.2 CDSE Conclusions: Identified Successful CDSE Themes

Although the primary purpose of this research was to document current CDSE practices in industry, including issues encountered, lessons learned, and success factors, several emerging CDSE-related "success themes" have been identified based on this CDSE research. Table 8.1 summarizes the proposed successful CDSE themes.

Table 8.1: Summary of proposed successful CDSE themes.

CDSE Topic	Proposed Successful CDSE Theme
Collaboration Situation and Management	It is necessary to have a well-defined SE and program organizational structure, with additional middle-management to coordinate efforts across development sites and companies.
Collaboration Situation and Management	Management buy-in and wide-spread enforcement of the processes is necessary for consistency and for capturing knowledge, data, and decisions.
Collaboration Tools	Collaboration tools are critical to CDSE work; the better the tools and the processes in place for their use, the more "distributed" the work can be and the less resources that are wasted.
Collaboration Tools	For SE, not only are the tools used to collaborate important (such as meeting resources, teleconference lines, email), but widely available collaborative product development tools are also needed to support successful SE and development, such as DOORS, Rational Rose, and modeling and simulation environments.
Knowledge, Data, and Decision Management	Transference and sharing of classified data is an issue that affects almost all aspects of CDSE, when applicable. Improving classified data transfer and sharing mechanisms would be a smart place to dedicate resources to improve CDSE.
Knowledge, Data, and Decision Management	Company proprietary data development and sharing is an issue that affects almost all aspects of CDSE, including trust between teams, product integration, system cohesiveness, and information dissemination. Determining how to deal with and overcome company proprietary data barriers is an area where resources should be focused immediately to facilitate CDSE.
Knowledge, Data, and Decision Management	Better methods are needed to support and facilitate dissemination of program information and decisions to teams. Engineers tend to be overwhelmed by emails and meetingss, and therefore are not able to keep up with important program data. Improving communications between CDSE personnel is another area where resources should be dedicated. One possibility: include digital screens relaying program information in collaborative work areas.
Knowledge, Data, and Decision Management	Well-established and supported tools and processes are needed for successful knowledge, data, and decision management practices. There are many issues to overcome for successful knowledge, data, and decision management, including: corporate firewalls; proprietary, ITAR, and classified data restrictions, different management decision making philosophies; time differences, database locations, etc.
SE Processes and Practices	Formal, contractually obligated, and widely-publicized SE processes are needed to control all aspects of systems development (tool use, knowledge management, decision making, configuration management, etc.) to ensure artifact consistency and smooth integration.
SE Processes and Practices	Widely available and platform independent system modeling and simulation tools are needed to confine system defects in phase and facilitate system-wide integration.
SE Processes and Practices	System interfaces (mechanical, logical, electrical), especially those that cross company boundaries, are a problem area due to the many issues discussed in this thesis. Recognize this issue and the importance of system interfaces, and dedicate resources early on to better define and monitor interface development in a CDSE environment.
Social and Cultural Environment	Program kick-off face-to-face, and regularly scheduled face-to-face meetings are necessary to build and maintain relationships and trust between teams. For example, issues of mistrust, company cultural differences, and misunderstandings have been remedied by repeated interactions and the ability to build relationships over time.
Social and Cultural Environment	It is important to have team-building activities or social events in a CDSE environment - the social environment is believed by many to be more formal, and almost all interviewees suggested team social events as a mechanism to improve team relationships.

8.3 Identified Variation in CDSE Practices

This exploratory study confirmed that there are variations in how CDSE is performed and thus reinforces the case that additional rigorous research is needed to understand what processes and organizations, under what conditions, will yield the best results. As Chapter 7 describes in greater detail, further research is needed to determine the SE processes, system architectures, team organizational structures, knowledge management practices, etc. that bolster CDSE success. The variations in how CDSE is performed also indicates a maturity factor may be necessary to distinguish between less mature and more mature CDSE capabilities, similar to program and organizational maturity factors (such as CMMI) as described in Chapter 7.

Appendix A

Standard Systems Engineer Interview Template

Company:

Project:

Name:

Date:

Read the following:

This interview is being conducted to gather your perspective on how collaborative, distributed systems engineering (CDSE) is currently being done and to hear your recommendations for improving CDSE efforts in the future.

Interviewee is advised of the following:

"Please note that your participation in this interview is voluntary; you many decline to answer any or all questions; you may decline further participation in this interview at any time without adverse consequences; and your confidentiality and/or anonymity are assured."

Standard SE Interview Questions:

Section 1: Basic Information about Interviewee

1. What is your current role/function?
2. How long have you been in this position?
3. How long have you been on this project?
4. How long have you been a systems engineer?
5. What is your disciplinary background? (if not always in SE)?

Section 2: Current Collaboration Situation

1. On a daily basis, do you collaborate/communicate with engineers at other sites (via email/telephone/Instant message)? If so, how often, with whom?
2. Do you attend virtual meetings? (via the web, teleconferences?)
3. Are your collaborative, distributed development efforts influenced or affected by time differences? How?
4. Can you give a rough estimate of how much of your time you spend performing "collaboration activities" you wouldn't otherwise spend time on? (Preparing collaboration tools, writing emails instead of talking to people, etc.)
5. How is this collaboration environment different from a "traditional" SE environment? What are you doing differently?

Section 3: Collaboration Tools and IT

1. What tools, if any, do you use for collaboration? (email, IM, telephone, version control, conferencing?)
2. Did you receive training on how to use all/any collaboration tools? If so, what type of training (class, online, 1-on-1, etc.)?
3. How useful are the tools you use?

4. Does everyone use the same tools/same versions of tools?
5. Do you have now or have you previously had any issues with the tools you use?
6. Do you have any concerns about identification/security (clearances, log-ons, passwords, etc.) or tool access?
7. Do you have any recommendations for improving CDSE tools/ IT?

Section 4: Knowledge and Data Management

1. Do you find it difficult to find the information you need when you need it? Explain:
2. How are knowledge and data currently managed on your project?
3. How are decisions are made and documented (architecture, processes, etc.)?
4. Are decisions, meeting minutes, design rationales, etc. well documented across CDSE collaboration sites?
5. From your experiences, what would you consider to be the Knowledge/Data Management issues for CDSE?
6. Do you have any recommendations for improving how knowledge/data is managed for CDSE environments?

Section 5: Technical Product Issues

1. Can you describe the current product design process?
2. If you are aware, have you had to modify traditional SE processes to accommodate for collaboration or distribution?
3. Have you had to modify the product/architecture in any way to accommodate for CDSE? How?
4. Do all CDSE locations use the same processes and procedures for SE product development?

5. If not (Answer N to above Q), does this present a problem?
6. Do you believe the integrity of the final product is affected by CDSE? How?
7. Do you have any suggestions for improving the product/architecture when working in a CDSE environment?

Section 6: Social and Cultural Effects

1. How do you feel about working in a distributed, collaborative environment?
2. Have you met "face-to-face" the engineers you are working with at other locations? If so, why?
3. Have you encountered language/interpretation issues? If so, what are they?
4. Do you have trouble "trusting" your CDSE counterparts at other locations (their estimates, advice, and recommendations)?
5. Is your decision making affected by CDSE efforts? How?
6. Is your supervisor/manager co-located?
7. If not (Answer N to above Q), do you feel that you are at a disadvantage? (lack of promotion/raise opportunities, less professional advice, etc.)
8. If you have had experience working in a non-CDSE environment, how does this CDSE social experience compare (can't make jokes, issues with communication, etc.)?
9. Are you uncomfortable with using collaboration tools and information technology?
10. What would you consider to be the social/cultural issues for CDSE?
11. Do you have any suggestions for improving social/cultural relations with CDSE counterparts at other locations?

Section 7: CDSE Benefits

1. What would you consider to be the benefits of distributed collaboration?

Section 8: CDSE Motivation, Success, and Future Work

1. What do you believe to be the motivation behind distributed collaboration?

2. Do you believe your project/product/service to be successful? Why or why not?

3. Do you believe collaboration efforts to be successful? Why or why not?

4. Can you identify any future work you would like to see on this topic?

Appendix B

SE Leadership Interview Template

Company:

Project:

Name:

Date:

Read the following:

This interview is being conducted to gather your perspective on how collaborative, distributed systems engineering (CDSE) is currently being done and to hear your recommendations for improving CDSE efforts in the future.

Interviewee is advised of the following:

"Please note that your participation in this interview is voluntary; you many decline to answer any or all questions; you may decline further participation in this interview at any time without adverse consequences; and your confidentiality and/or anonymity are assured."

SE Leadership Interview Questions:

Section 1: Basic Information about Interviewee

- What is your current role/function?
- How long have you been in this position?
- How long have you been on this project?
- How long have you been a systems engineer?
- What is your disciplinary background? (if not always in SE)?

Section 2: Program-Specific Information

1. What is the nature of the collaborative, distributed project (partnership, sub-contract, prime)?
2. Who/which location does what?
3. How is the team "legally" structured? (contract)
4. What is the motivation behind establishing a collaborative, distributed systems team?
5. In general, how is the distributed collaboration managed and coordinated?(Who orchestrates meetings?Do you hold weekly telecoms? Are there site-reps at each location?)

Section 3: Current Collaboration Situation

1. On a daily basis, do you collaborate/communicate with engineers at other sites (via email/telephone/Instant message)? If so, how often, with whom?
2. Do you attend virtual meetings? (via the web, teleconferences?)
3. Are your collaborative, distributed development efforts influenced or affected by time differences? How?
4. Can you give a rough estimate of how much of your time you spend performing "collaboration activities" you wouldn't otherwise spend time on? (Preparing collaboration tools, writing emails instead of talking to people, etc.)

5. How is this collaboration environment different from a “traditional” SE environment? What are you doing differently?

6. Do you visit each site your team works at? If so, how often?

7. Do you schedule and/or allow periodic “face-to-face” meetings?

Section 4: Programmatic Issues

1. Do you believe this program/product/service is successful (relatively speaking)?

2. Do you have any metrics in place to measure the success of your team? Any specific metrics to measure collaboration effectiveness? If so, what are they?

3. Is your team performing/completing tasks on schedule? If not, why not?

4. Is your team performing/completing tasks within budget? If not, why not?

5. Do you allocate resources specifically for collaboration? (space, money, time?)? If so what and how much?

6. If applicable, how does the success of your CDSE project/team compare to that of other “traditional” projects/teams you have worked on or lead?

7. Does your team utilize process/product improvement resources to save money, reduce efforts, or cut cycle-time? If so, how?

Section 5: Collaboration Tools and IT

1. Who dictates which collaboration tools and tool versions should be used?

2. What tools, if any, do you use for collaboration? (email, IM, telephone, version control, conferencing?)

3. Did you receive training on how to use all/any collaboration tools? If so, what type of training (class, online, 1-on-1, etc.)?

4. How useful are the tools you use?

5. Does everyone use the same tools/same versions of tools?

6. Do you have now or have you previously had any issues with the tools you use?

7. Do you have any concerns about identification/security (clearances, log-ons, passwords, etc.) or tool access?

8. Do you have any recommendations for improving CDSE tools/ IT?

Section 6: Knowledge and Data Management

1. Do you find it difficult to find the information you need when you need it? Explain:

2. How are knowledge and data currently managed on your project?

3. How are decisions are made and documented (architecture, processes, etc.)?

4. Are decisions, meeting minutes, design rationales, etc. well documented across CDSE collaboration sites?

5. From your experiences, what would you consider to be the Knowledge/Data Management issues for CDSE?

6. Do you have any recommendations for improving how knowledge/data is managed for CDSE environments?

Section 7: Technical Product Issues

1. Can you describe the current product design process?

2. If you are aware, have you had to modify traditional SE processes to accommodate for collaboration or distribution?

3. Have you had to modify the product/architecture in any way to accommodate for CDSE? How?

4. Do all CDSE locations use the same processes and procedures for SE product development?

5. If not (Answer N to above Q), does this present a problem?

6. Do you believe the integrity of the final product is affected by CDSE? How?

7. Do you have any suggestions for improving the product/architecture when working in a CDSE environment?

Section 8: Social and Cultural Effects

1. How do you feel about working in/leading in a distributed, collaborative environment?

2. Have you met "face-to-face" the engineers you are working with at other locations? If so, why?

3. Have you encountered language/interpretation issues? If so, what are they?

4. Do you have trouble "trusting" your CDSE counterparts at other locations (their estimates, advice, and recommendations)?

5. Is your decision making affected by CDSE efforts? How?

6. Is all of your personal team co-located?

7. If not (Answer N to above Q), do you feel that there are issues with evaluating employee professional development or providing adequate supervision? (lack of promotion/raise opportunities, less professional advice, etc.)

8. Is your supervisor/manager co-located?

9. If not (Answer N to above Q), do you feel that you are at a disadvantage? (lack of promotion/raise opportunities, less professional advice, etc.)

10. If you have had experience working in a non-CDSE environment, how does this CDSE social experience compare (can't make jokes, issues with communication, etc.)?

11. Are you uncomfortable with using collaboration tools and information technology?
12. Do you train any of your team members on working in distributed, collaborative environments (teamwork training, cultural awareness, etc.)? If so, how?
13. What would you consider to be the social/cultural issues for CDSE?
14. Do you have any suggestions for improving social/cultural relations with CDSE counterparts at other locations?

Section 9: CDSE Benefits

1. What would you consider to be the benefits of distributed collaboration?

Section 10: CDSE Motivation, Success, and Future Work

1. What do you believe to be the motivation behind distributed collaboration?
2. Do you believe your project/product/service to be successful? Why or why not?
3. Do you believe collaboration efforts to be successful? Why or why not?
4. Can you identify any future work you would like to see on this topic?

Appendix C

Company B Group Leadership Interview Template

¡p align="center"¿ **Questions for Company B Program B Leadership** ¡/p¿

1. Can you describe how the distributed collaboration of systems engineering related activities are managed and coordinated?

2. What collaboration methods and tools are you using and what issues have you encountered with them?

3. What approaches do you use to manage and share knowledge and data between collaborating locations, and how effective do you think these are?

4. Have you had to modify any of your traditional design processes, products, interfaces, architectures, or decision making processes to accommodate distributed collaboration?

5. Do you believe the systems engineering collaboration efforts have been successful, and what "metrics" or indicators do you use to judge this?

6. What barriers, both social and technical, have you encountered in this distributed collaborative environment and how have you dealt with them?

7. What impact do you believe the distributed collaboration effort had on this program as a whole - positive, neutral, or negative?

Appendix D

Interview Protocol

Interview Checklist - Collaborative, Distributed Systems Engineering (CDSE)

Darlene Utter, MIT ESD

Darlene@mit.edu

☐ 1. Give Interviewee Contact Information for follow-up, questions
(Business Card with phone/email)

☐ 2. Provide Interviewee with Research Background Overview
(1-page summary of key issues, 1 minute)

☐ 3. Ask Interviewee for Informed Consent - COUHES form
(Explain form, ask for signature, sign as witness)

☐ 4. Read Interviewee their Rights as a Participant

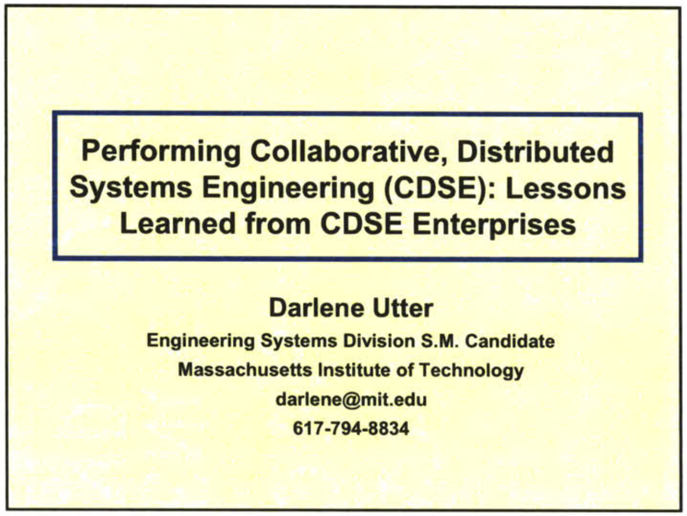
☐ 5. Explain Interview Process and Post-Interview Process
(I will be taking notes during the interview to document your responses to my questions. After the interview I will transcribe my notes into a summary document, capturing all of the key issues we discussed. I will email you the transcribed summary for your approval and feedback.)

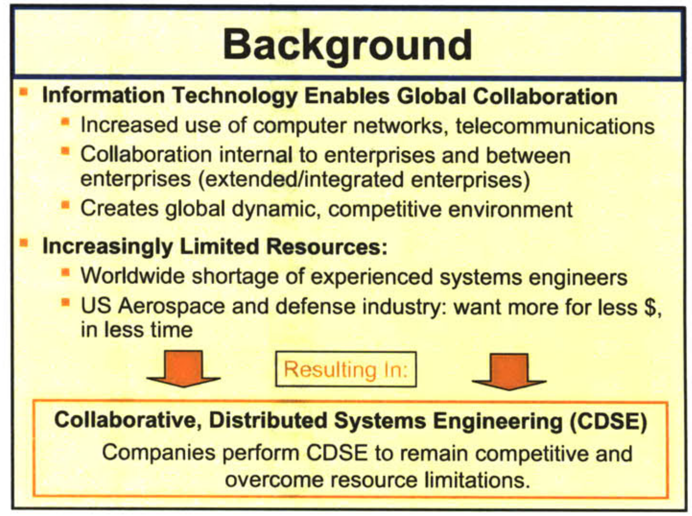
☐ 6. Interview
(Will last 45 minutes)

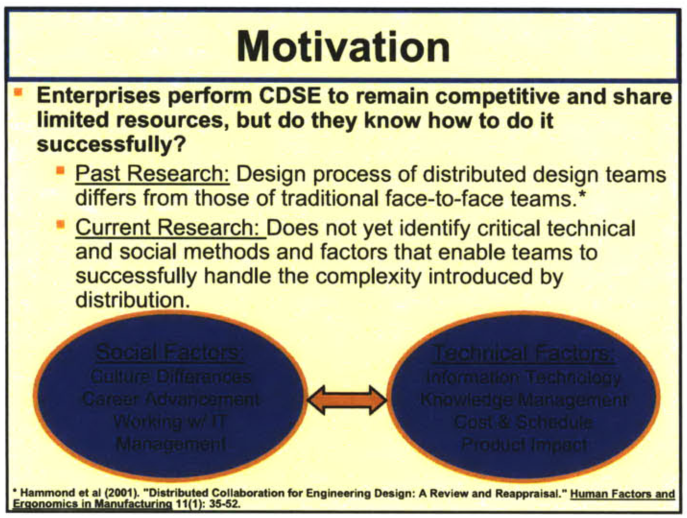
Appendix E

Short Summary of CDSE Research

A copy of the short summary of the CDSE research motivation, background and objectives is located on the next page.







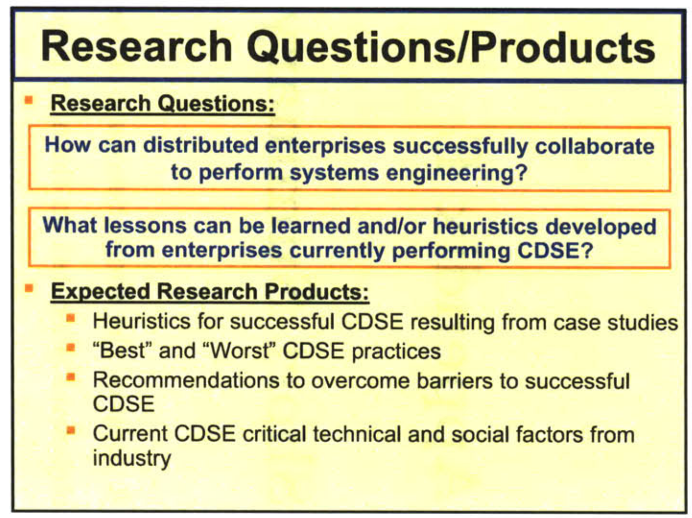


Figure E-1: Short CDSE research summary provided to interviewees.

Appendix F

COUHES Consent Form

A copy of the original COUHES form all interviewees were require to sign is on the following page.

MAY 26 '06

CONSENT TO PARTICIPATE IN INTERVIEW

Performing Successful, Collaborative, Distributed Systems Engineering (CDSE)

You have been asked to participate in a research study conducted by Darlene Utter from the Engineering Systems Department at the Massachusetts Institute of Technology (M.I.T.). The purpose of the study is to gather information about the key social and technical aspects of collaborative, distributed, systems engineering in large US aerospace and defense companies. The results of this study will be included in Darlene Utter's Masters Thesis. You were selected as a possible participant in this study because you are involved/ have been involved in a large aerospace or defense systems engineering project. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

• This interview is voluntary. You have the right not to answer any question, and to stop the interview at any time. We expect that the interview will take about 1 hour.

• You will not be compensated for this interview.

• Unless you give us permission to use your name, title, and / or quote you in any publications that may result from this research, the information you tell us will be confidential.

This project will be completed by January 2007.

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

(*Please check all that apply*)
[] I give permission for the following information to be included in publications resulting from this study:
[] my name [] my title [] direct quotes from this interview

Name of Subject

Signature of Subject ______________________________ Date __________

Signature of Investigator ____________________ Date ________

Please contact Darlene Utter, by email at darlene@mit.edu, or by phone at 617-794-8834 with any questions or concerns.

If you feel you have been treated unfairly, or you have questions regarding your rights as a research subject, you may contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T., Room E32-335, 77 Massachusetts Ave, Cambridge, MA 02139, phone 1-617-253 6787.

APPLICATION FOR APPROVAL TO USE HUMANS AS EXPERIMENTAL SUBJECTS
(EXEMPT FORM) – revised 8/5/2003
- 1 -

Figure F-1: Copy of Original COUHES Informed Consent form required from each interviewee.

Bibliography

[1] Boeing-uq initiative tackles key engineering skills shortage. 2003. Available from: `http://www.uq.edu.au/news/index.html?article=4949` [Accessed March 28, 2006].

[2] Tiranee Achalakul, Booncharoen Sirinaovakul, and Nion Nuttaworakul. Virtual laboratory: A distributed collaborative environment. *Computer Applications in Engineering Education*, 12(1):44 – 53, April 23 2004.

[3] Jack D. Becker, Roger D. Ballentine, Angelique Lee, and Carole Townsley. Collaborative technology tools for virtual teaming. In *Proceedings of the 5th Americas Conference on Information Systems*, Milwaukee, WI., 1999.

[4] Lyn Berry-Helmlinger. Need for systems engineers growing in lots of industries. 2003. Available from: `http://www.bizjournals.com/denver/stories/2003/11/24/newscolumn2.html` [Accessed November 27, 2006].

[5] Defense Science Board. Report of the defense science board task force on future strategic strike skills. March 2006.

[6] Dennis M. Buede. *The Engineering Design of Systems: Models and Methods.* John Wiley and Sons, Inc., New York, 1999.

[7] Linda Candy and David Harris. Evaluating model data exchange between systems engineering tools. *Systems Engineering*, 4(1):13–23, February 13 2001.

[8] John Cleveland. Toyota's other system - this one for product development. *Automotive Design and Production*, (February 2006):18–22, February 2006.

[9] Harry E. Crisp and Pin Chen. Coalition collaborative engineering environment. *INCOSE Insight*, 5(3):13–15, October 2002.

[10] Thomas H. Davenport and Laurence Prusak. *Working Knowledge: How Organizations Manage What They Know.* Harvard Business School Press, USA, 1998.

[11] Leonard David. Experts grapple with exploration agenda. 2005. Available from: `http://www.space.com/spacenews/050407_nss_exploration.html` [Accessed 11/27/2006].

[12] Heidi L. Davidz, Deborah J. Nightingale, and Donna H. Rhodes. Accelerating the development of senior systems engineers. In *Fifteenth Annual International Symposium: International Council on Systems Engineering 2005*, Systems Engineering: Bridging Industry, Government, and Academia", Rochester, NY, 2005.

[13] Department of Defense. Military standard, engineering management. 1974. Section 3.3. Available from: `http://sepo.spawar.navy.mil/MI-STD-499A_Engr_Management.pdf`.

[14] Grover L. (SES USAF) Dunn. Memorandum for almajcom-foa/lg: Product data management information technology. 2004.

[15] Jeffrey H Dyer and Nile W. Hatch. Using supplier networks to learn faster. *MIT Sloan Management Review*, (Spring):57–63, 2004.

[16] Jose Martin Molina Espinosa and Khalil Drira. A multi-modal coordination service for cooperative distributed systems engineering. In *International Conference on Systems, Man and Cybernetics*, volume 6, Tunisia, 2002. IEEE.

[17] Jose Martin Molina Espinosa, Khalil Drira, and Thierry Villemur. The responsibility management system for collaborative meetings scheduling in the distributed system engineering project. In *Workshop on Knowledge Media Networking*. IEEE, 2002.

[18] Bjorn Fagerstrom and Lars-Erik Olsson. Knowledge management in collaborative product development. *Systems Engineering*, 5(4):274–285, May 4 2002.

[19] Michael Hammer and Steven Stanton. How process enterprises really work. *Harvard Business Review*, 77(6):108–112, November/December 1999.

[20] Janeen Hammond, Richard J. Koubek, and Craig M. Harvey. Distributed collaboration for engineering design: A review and reappraisal. *Human Factors and Ergonomics in Manufacturing*, 11(1):35–52, 2001.

[21] David Harris. Standardised model data exchange for dispersed systems engineering design teams. In *Proceedings of the IFIP TC5 WG5.3/5.7/5.12 Fourth International Conference on the Design of Information Infrastructure Systems for Manufacturing*, pages 340–351. Kluwer, B.V., 2001.

[22] David Harris. Supporting human communication in network-based systems engineering. *Systems Engineering*, 4(3):213 – 221, August 9 2001.

[23] Craig M. Harvey and Richard J. Koubek. Cognitive, social, and environmental attributes of distributed engineering collaboration: A review and proposed model of collaboration. *Human Factors and Ergonomics in Manufacturing*, 10(4):369–393, September 14 2000.

[24] Josie P.H. Huang. A conceptual framework for understanding collaborative systems evaluation. In *Workshop on Evaluation of Collaborative Information Systems and Support for Virtual Enterprises (ECE)*, Sweden, 2005.

[25] INCOSE. What is systems engineering? 2004. Available from: `http://www.incose.org/practice/whatissystemseng.aspx`.

[26] Sieja Komi-Sirvio and Maarit Tihinen. Lessons learned by participants of distributed software development. *Knowledge and Process Management*, 12(2):108–122, 2005.

[27] Patrick Lencioni. *The Five Dysfunctions of a Team, A Leadership Fable.* Jossey-Bass, San Fransisco, 2002.

[28] Jessica Lipnack and Jeffrey Stamps. *Virtual Teams: People Working Across Boundaries with Technology.* John Wiley and Sons, New York, NY, 2nd edition, 2000.

[29] Azad Madni, Weiwen Lin, and Carla C. Madni. Ideon: An extensible ontology for designing, integrating, and managing collaborative distributed enerprises. *Systems Engineering*, 4(1):35–48, September 28 2004.

[30] Mark W. Maier. Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284, November 3 1998.

[31] Earll Murman. *Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative.* The Lean Enterprise Value Foundation, Inc., New York, 2002.

[32] Fred Niederman and Catherine M. Beise. Defining the "virtualness" of groups, teams, and meetings. In *SIGCPR '99: Proceedings of the 1999 ACM SIGCPR Conference on Computer Personnel Research*, pages 14–18, New Orleans, Louisiana, United States, 1999. ACM Press.

[33] I. Nonaka. A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1):14–37, 1994.

[34] M. Polanyi. *The Tacit Dimension.* Doubleday, New York, 1966.

[35] Andrew P. Sage and Christopher D. Cuppan. Information, knowledge, systems management. 2(4):325 - 345, 2001.

[36] Aaron Shenhar. Systems engineering management: A framework for the development of a multidisciplinary discipline. *IEEE Transactions on Systems, Man, and Cybernetics*, Volume 24(2):327–332, February 1994.

[37] Deborah Sole and Lynda Applegate. Knowledge sharing practices and technology use norms in dispersed development teams. In *ICIS '00: Proceedings of the*

Twenty First International Conference on Information Systems, pages 581–587, Brisbane, Queensland, Australia, 2000. Association for Information Systems.

[38] R.A. Stebbins. *Exploratory Research in the Social Sciences.* Sage Publications, Thousand Oaks, California, 2001.

[39] International Council on Systems Engineering Tools Database Working Group (TDWG). Tools database. 2006. Available from: `http://www.incose.org/ProductsPubs/products/toolsdatabase.aspx` [Accessed October 11, 2006].

[40] John Hopkins University. Technical management part time programs in engineering and applied sciences: Origins and approach. 2006. Available from: `http://www.jhuapl.edu/education/techmgt/origins.html` [Accessed November 27, 2006].

[41] Amanda Vaughn and Ed Fleming. Leadership of a geographically diverse team: The sbss ground segment example. In *Lean Aerospace Initiative Plenary Conference*, San Antonio, TX, 2006. Lean Aerospace Initiative.

[42] Aurora Vizcano, Mario Piattini, Manuel Martnez, and Gabriela Aranda. Evaluating collaborative applications from a knowledge management approach. In *14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05)*, pages 221–225. IEEE Computer Society, 2005.

[43] Randy C. Zittel. Simulation based (systems) engineering or collaborative based acquisition. In *Ninth Annual International Symposium of the International Council on Systems Engineering*, Systems Engineering: Sharing the Future, Brighton, England, 1999.