NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

THE CASE FOR A KNOWLEDGE BASED DOD SOFTWARE ENTERPRISE: AN EXPLORATORY STUDY USING SYSTEM DYNAMICS

by

Richard J. Dixon, Jr.

September 2007

Thesis Advisor: Tarek Abdel-Hamid
Co-Advisor: Daniel Dolk

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: The Case for a Knowledge Based DoD Software Enterprise: An Exploratory Study using System Dynamics	**5. FUNDING NUMBERS**
6. AUTHOR(S) LCDR Richard Jerome Dixon, Jr.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This research project will examine the DoD Software Acquisition process utilizing Jay Forrester's System Dynamics methodology. Well known acquisition issues and policies will be examined with specific focus on oversight, process integration, process discipline, and knowledge management. These issues will be examined for causality and dependent relationships. Additionally, a proof of concept systems dynamics model will be developed to simulate the system and test possible interventions for organizational structure and policy.

DISCLAIMER: This paper contains judgments and conclusions of the authors. It does not necessarily reflect any policy or position held by the Departments of Navy or Defense.

14. SUBJECT TERMS Management of Technology, Strategic Decision Making, Software Acquisition, System Dynamics Modeling, Product-Line Architecture, Learning Curve Theory	**15. NUMBER OF PAGES** 77
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

THE CASE FOR A KNOWLEDGE BASED DOD SOFTWARE ENTERPRISE: AN EXPLORATORY STUDY USING SYSTEM DYNAMICS

Richard J. Dixon, Jr.
Lieutenant Commander, United States Navy
B.S., U.S. Naval Academy, 1997

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2007

Authors: Richard J. Dixon, Jr.

Approved by: Tarek Abdel-Hamid
Thesis Advisor

Daniel Dolk
Co-Advisor

Daniel Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This research project will examine the DoD Software Acquisition process utilizing Jay Forrester's System Dynamics methodology. Well known acquisition issues and policies will be examined with specific focus on oversight, process integration, process discipline, and knowledge management. These issues will be examined for causality and dependent relationships. Additionally, a proof of concept systems dynamics model will be developed to simulate the system and test possible interventions for organizational structure and policy.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Dr. Tarek Abdel-Hamid and Professor Daniel Dolk, for their guidance and assistance in performing this study. Their time and attention both guided and challenged me through this process. Their efforts ensured this path of research and discovery was a valuable learning experience that I will always remember and cherish.

I also wish to thank Regina Dixon, Charles Dixon, and Grace Dixon. Regina, your fortitude, tolerance and understanding was needed and greatly appreciated. You managed our home, added a new addition to the family, home schooled a highly energetic boy, and raised both of our children while I was away at the library or locked in the study. All I had to do was go to school and occasionally walk the dog. Without your support and discipline, this thesis would not have happened. Charles, your big smile and Spiderman versus Wolverine scenarios were put on hold during my studies. I owe you big little man. Autobots – Roll out! And remember with great power comes great responsibility. Gracie girl, I left you to the musings of your big brother. Thank you for your patience and your beautiful little smile.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

DoD systems are increasingly software dependent. Software acquisition has become increasingly more complex from a technical perspective as well as from a management perspective. The increased reliance on Industry and the increased complexity of systems creates challenges for the current acquisition structure and system.

These challenges have existed for decades but seem amplified by software. Some common issue areas are oversight, senior leadership, process discipline, mature technology, and knowledge management.

Systematic problems are typically a result of system structure. The current software acquisition model is platform-centric where programs compete for funds and resources. This paper will examine a domain-centric model where the respective domain manages a portfolio of investments with an enterprise product-line architecture.

Not only are there technical challenges, but there are resource challenges as well. A RAND study predicts .4 percent growth in the labor force over the next decade with .3 percent the following decade. Questions also abound over how strong the labor force will be for science and math skills with current outputs of the U.S. education system. A reduced labor force for skilled workers means higher competition with Industry for qualified personnel. Consider these trends with the upcoming transfer of the Baby Boom generation into retirement and a scary picture emerges.

The goal of this research is to examine these issue areas and determine dependencies and relationships for possible interventions. We will use System Dynamics methodology to develop a proof of concept model to simulate the software acquisition process for policy and analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

> The current situation is characterized by massively accelerated cost growth in many major defense programs, lack of confidence by senior leaders, and no appreciable improvement in the defense acquisition system in the past two decades -- DAPA Project Team Report December 2005

DoD's programs for acquiring major weapon systems have taken longer, cost more, and have delivered fewer quantities and capabilities than planned. GAO has documented these problems for decades. Most recently, GAO reported that 27 major weapon programs assessed since they began product development have experienced cost increases of nearly 34 percent over their original research, development, test, and evaluation (RDT&E) estimates, and increases of almost 24 percent in acquisition cycle time.

Although the military services fight together on the battlefield as a joint force, they do not identify warfighting needs and make weapon system investment decisions in the same manner. DoD has taken steps to identify warfighting needs from a more joint requirements perspective, but the department's service-centric structure and fragmented decision-making processes are at odds with the integrated, portfolio management approach used by successful commercial companies to make enterprise-level investment decisions. (GAO, 2007)

A. BACKGROUND AND PROBLEM STATEMENT

Systems are becoming more and more software intensive. According to a Defense Science Board's Task Force on Defense Software report (2000), military aircraft dependency on software increased from approximately 10 percent functionality on the F-4 to 80 percent functionality on the F-22—equivalent to a 2 percent per year increase (1960-1995). Though hardware is following Moore's Law and gaining in performance and capacity at an exponential rate, DoD software acquisitions seem to be following Murphy's Law with schedule and cost overruns and less than desirable delivered capability.

The Software Engineering Life Cycle is a complex system driven by interactions and feedback loops across various stages. New software development must undergo the rigors of a six phase process: concept development, requirements development, design, software development, test, and release. A program manager seeking to field a new software application can normally expect a multiyear effort from concept to deployment. For ready-made commercial off the self (COTS) products, the concept and development phase is reduced, but the product must still undergo the scrutiny of testing and certification through an external release process.

The objective of this study is to develop a high level acquisition model for DoD to simulate knowledge-based polices and constructs in order to improve the software acquisition process. The model can be used as a high level starting point for policy makers, domain managers, or program managers to run policy what-if scenarios or analyze current problems for future decision making.

B. RESEARCH METHODOLOGY

Our research methodology is based on the System Dynamics (SD) methodology proposed by Jay Forrester (1961), shown below in Figure 1.

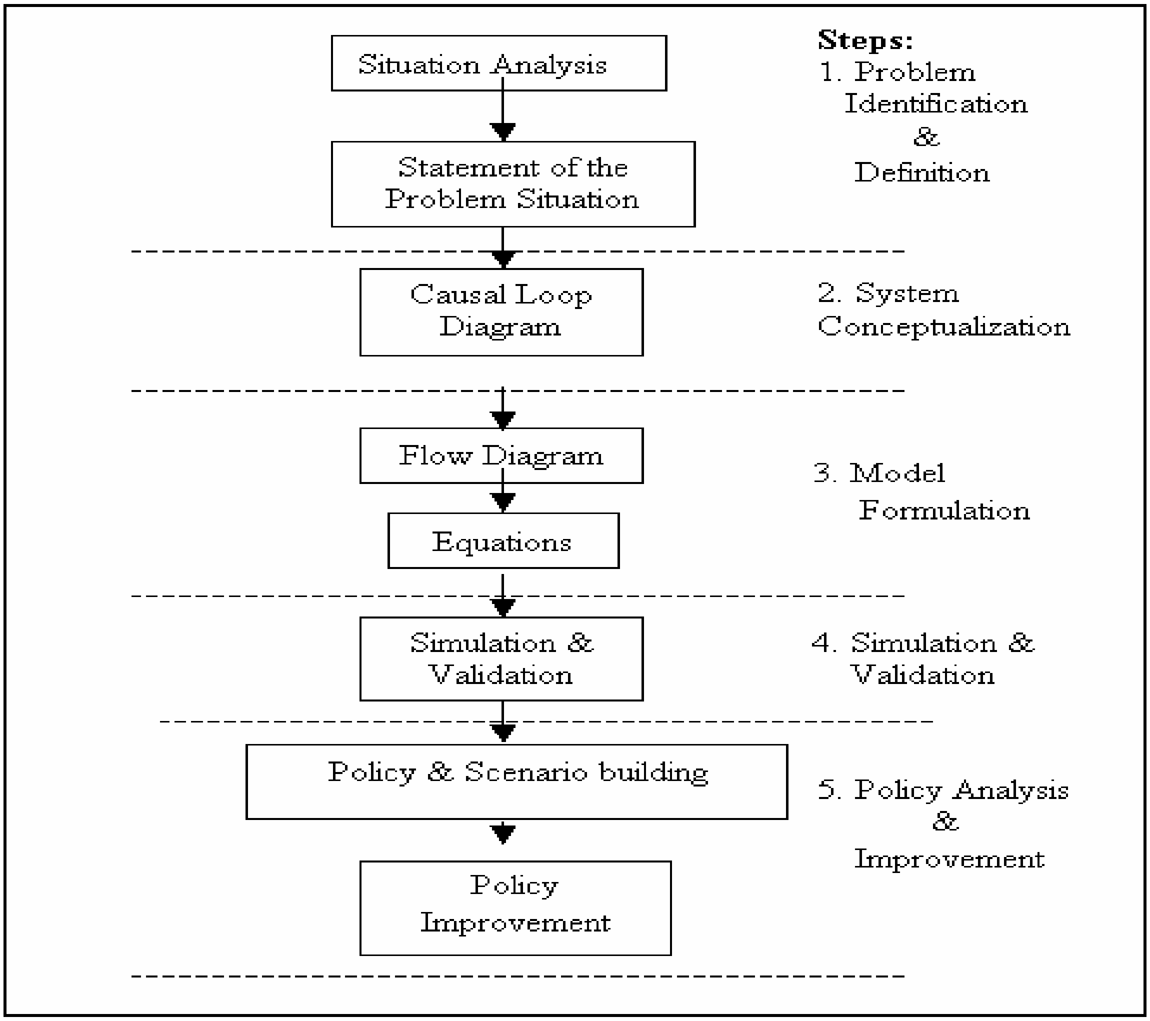

Figure 1. System Dynamics Methodology

To develop situational awareness, we conducted a literary review of various GAO and DoD assessments, held various interviews with former and current program managers, and reviewed OSD/ASN congressional testimony and policy documents. Chapter II will provide a brief discussion of the Acquisition Process and the various embedded relationships. Chapter III will introduce the model hypothesis with a supporting industry case study. Chapter IV will introduce a brief summary of System Dynamics and our causal representation of the system based on relationships from Chapter II. Chapter V will introduce system dynamics model and describe formulations representing the acquisition system. Chapter VI will baseline model to current environment and simulate various policy what-if scenarios. Finally, Chapter VII will offer conclusions and areas for further research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. RECURRING THEMES

We begin by first documenting the existing acquisition process for software intensive systems. After a brief overview of the process we will discuss common areas of concern from various sources and DoD assessments. These areas of concern will provide the basis for constructing our system dynamics model. Based on these concerns we will then examine causality and linkage between the different issue areas. Once we have identified causality and linkage, we will then propose a model to simulate the system. Once system is simulated, we will propose possible interventions to address areas of concern.

A. SLOW AND CUMBERSOME ACQUISITION PROCESS

DOD has three major processes involved in making weapon system investment decisions. These processes, depicted in Figure 2, are the Joint Capabilities Integration and Development System (JCIDS), the Planning, Programming, Budgeting and Execution (PPBE) system, and the Defense Acquisition System (DAS).

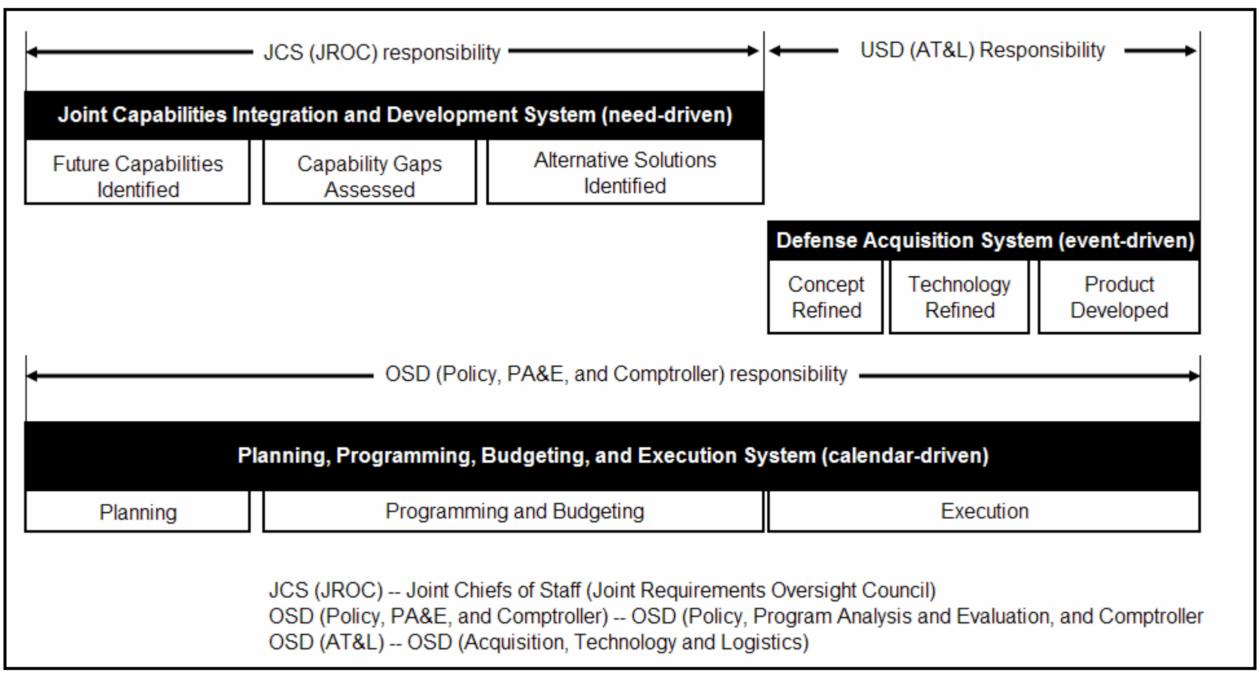

Figure 2. DoD Requirements, Budgeting, and Acquisition Processes

These processes can be described as service-centric, program centric, extremely complex, and slow. This is due to the fact that they are designed and organized around the competition, control, and flow of funds.

This competitive service-centric structure leads to a fragmented decision-making process that impedes DOD's ability to prioritize warfighting needs or leverage existing knowledge and information across the enterprise (DoD, 2006). The military services identify warfighting needs individually, and department-level organizations are not optimized to integrate the services' results or to evaluate their fiscal implications early in the process. Historically, this approach has contributed to duplication in weapon systems and equipment that does not interoperate. At the department level, Functional Capability Boards oversee each of the eight functional areas, reviewing the services' assessments, and providing recommendations to the Joint Requirements Oversight Council (JROC), which leads the JCIDS process. However, defense experts and DOD officials report that the Functional Capability Boards do not have the staff or analytical resources required to effectively evaluate service assessments within the context of the broader capability portfolio and assess whether the department can afford to address a particular capability gap. Several recent studies have recommended that DOD increase joint analytical resources for a less insular understanding of warfighting needs. The boards also lack the authority to make or enforce decisions divesting their capability area of existing programs to pay for new ones. This leaves DoD with ongoing programs that are scheduled to deliver capability that is either obsolete or no longer useful.

Resource allocation decisions are made through a separate process, the Planning, Programming, Budgeting, and Execution system (PPBE), which hinders the department's ability to weigh the relative costs, benefits, and risks of investing in new weapon systems. Within the PPBE system, the individual military services are responsible for budgeting and allocating resources under authority that is commonly understood to be based on Title 10 of the United States Code. PPBE is structured by military service and defense program, although the department integrates data on the services' current and projected budget requests under 11 crosscutting mission areas called Major Force Programs. The cross-cutting view provided by the Major Force Program structure is intended to facilitate a strategic basis for resource allocation, allowing the Secretary of Defense to more easily see where the greatest mission needs are and to re-allocate funds to meet those needs regardless of which service stands to gain or lose. However, GAO has reported in the past that the Major Force Program structure has not provided sufficient visibility in certain mission areas. Moreover, although they cut across the services, the program mission areas are not consistent with the more recently established capability areas used in the JCIDS process, and as a result, it is difficult to relate resources to capabilities (GAO, 2007).

The Defense Acquisition System is based on four phases. Requirements, Design, Develop, and Field. Testing can also be considered a phase but Testing is more effective if integrated early in the development process. Our model will concentrate on the DoD Software Acquisition Process. It is our belief that the Software Acquisition Process can be structured in a way to complement and improve the JCIDS process while also mitigating the financial instability caused by the PPBE process. If DoD can develop software faster and cheaper, the problems associated with the PPBE process are lessened. If DoD can develop software products at a higher quality and with better integration, the requirements process also becomes a little easier to navigate.

B. RECURRING THEMES WITHIN SOFTWARE ACQUISITION

1. Workforce Capacity

One of the most alarming issues with regard to workforce capacity is the low ratio of in-house support to outside contractor support. This is a function of the resident knowledge within the program per actual number of personnel. The DoD acquisition workforce experienced a 55% decrease from 1987-2006 (DoD, 2007). Coupled with senior acquisition officials reaching retirement age, and the increasing dynamic complexity of software, there is now a significant knowledge shortfall concerning the acquisition of software intensive systems. DoD has developed the Lead System Integrator model as a solution but this model only works if there is a certain level of knowledge resident in the program for oversight and control. From this relationship we see that the lower the level of knowledge and experience results in higher dependence upon outside contractor support.

The Acquisition community is presently at a low level and expected to proceed even lower once the Baby Boom generation enters the retirement window. Furthermore, the U.S. labor force is estimated to grow at only a level of .4 percent from 2010-2020 followed by .3 percent from 2020-2030 (RAND, 2006). A cursory look at the available talent pool shows our current education system is producing fewer scientists and engineers while Asia and India are producing more and more. We are witnessing a technical shift in capacity from the United States to Asia and India. Such trends as globalization and outsourcing will work to reinforce this dynamic even further. Government will not be able to compete with Industry for the remaining "knowledge workers" that will exist in the U.S. workforce. This dynamic will create a significant knowledge shortfall in the Acquisition community. The Acquisition community is preparing for this eventual dip in capacity by identifying critical skill sets and certifications with plans to minimize the baby boomer effect by formal training methods. (OSD, 2006). Formal training methods account for only 10-20% of the knowledge growth within an organization. Informal learning or experience accounts for the breadth of knowledge and builds the basis or data set for formal training. Assuming a retirement age of 66, 76% of the acquisition workforce will enter the retirement window in 2012. This significant drop in knowledge will require not only formal training and critical identification of skills, but structures to build domain experience as quickly as possible.

2. Oversight

Oversight is the ability to monitor and give direction for the effective management of a project. Based on current literature, current DoD oversight processes and procedures do not appear to be adequate or effective. According to 97% of the DAPA survey respondents, the current oversight and leadership process is deficient. In accordance with statute, the ASD(NII) and the DoD Component Acquisition Executives (CAEs) are responsible for acquisition oversight of IT acquisition programs, including serving as the Milestone Decision Authority for the programs under their cognizance (GAO, 2006). This is a relationship primarily between the Program Manager (Component Representative) and the ASD(NII)/CAE oversight team. OSD has made significant efforts to limit required reporting and reduce layers of oversight, but again, due to the nature of software intangibility, dynamic complexity of programs, and the lack of comparable data across domains, oversight is hard to capture and enforce.

There is an interesting dynamic to oversight. DoD has reduced the layers of oversight to no more than two levels between PM and OSD, however, due to increased reliance on industry, the issue is largely oversight of the contractor through to the PM. There is an optimal structure for a knowledge-based organization as well as an optimum size. In Malcolm Gladwell's book, "The Tipping Point", he discusses the idea that for a group of up to 150 members, there is a "community memory" that works such that either you know how to find something out, you do it yourself, or you know the person who does. Beyond this threshold of 150, he argues, an individual no longer knows whom to turn to, and must go through intermediaries, resulting in a communication breakdown. Research in cognition supports this argument, as documented in Robin Dunbar's *Grooming, Gossip, and The Evolution of Language*. Professor Dunbar discovered a relationship among primates between the relative size of their cortex and the number of primates inside a social group. He further concludes that language evolved as the basis for bonding to a social group, and there is an inherent limit to the amount of social interaction that we can handle based on the size of our cortex. As human communities grew too large for each member to personally associate through gestures, we developed language. The structure of language, including the words that we use, is designed for the purpose of advancing the social and sociological goals of the communities and the individuals who use them. This theory also integrates well with the problem of data semantics and different communities of interest. In summary, we can conclude that it is likely that for large scale projects involving more than 150 personnel, it may very well be more difficult to provide meaningful oversight. The current trend in DoD is larger and larger programs that deal with not just one system but families of systems. The Army's Future Combat System (FCS) is a prominent example as well as the Navy's ERP system for business modernization. Structured processes and metrics can alleviate this but today's acquisition environment with numerous contractors and subcontractors means that most processes and metrics are contractor dependent and not standard from project to project.

So the dynamic of reduced layers of oversight from PM to MDA, increased layers of oversight from contractor to PM, increased industry dependence, and increasingly larger and more complex programs, has made it worse for managers to socialize with subordinates and vendors to effectively capture what is going on in their respective programs.

One issue is the sheer size and number of programs within DoD. The second issue is the lack of mature processes with normalized metrics to compare scope, team composition, funding profiles, and performance results. Every program is different, and unique, however, one would expect that the program makeup for one DoD avionics system should be fairly similar to the program makeup of another DoD avionics system. In other words, it would be desirable if domains developed open architectures across the services horizontally rather than being service-specific. Possibly a Joint SPO can engineer the open architecture with standard processes, data semantics, and metrics for a respective domain. The Joint SPO would act as the domain lead and be responsible for oversight of their domain to the ASD(NII)/CAE oversight team. Services can then be responsible for their portfolio of investments within this respective domain. (See possible structure in Figure 3)

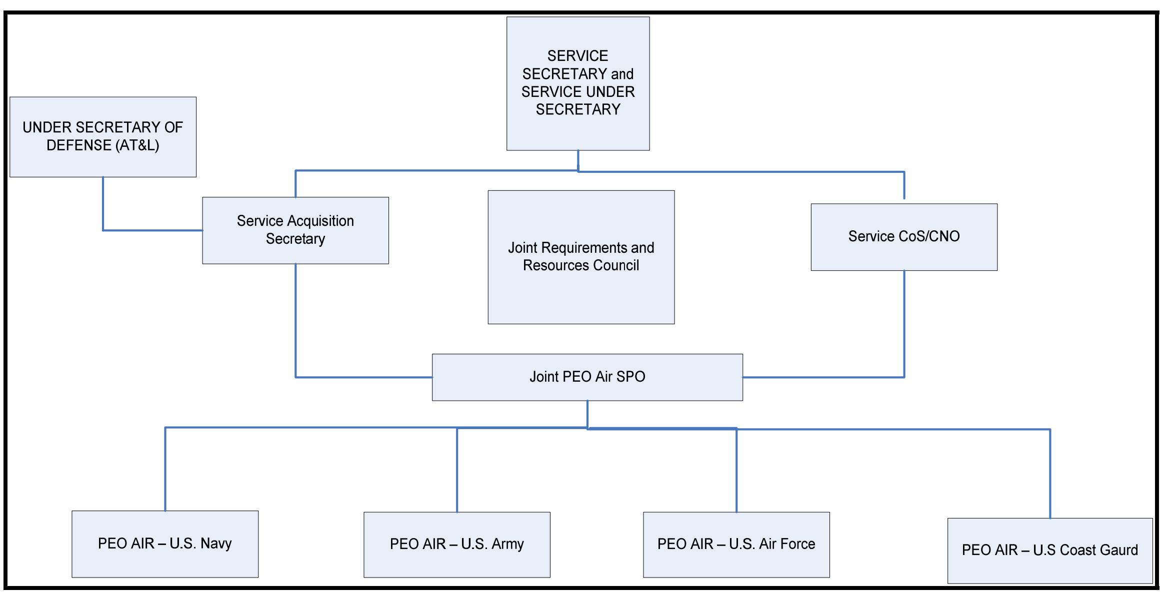

Figure 3. Possible Knowledge Based Domain Centric Organizational Structure

In summary, the key to achieving effective oversight is an integrated structure that allows social interaction and knowledge transfer among all key members. The organization should also possess and continually nurture mature and standardized processes with valuable metrics for feedback and optimization.

3. Integration

Integration can be assessed by the actual hands on process time and the addition of any non value added delays - internal or external to the process. The goal of integration is to reduce delays and create value. Value creation depends on the focus of the organization and the mission it supports. This focus on value is sometimes referred to as the value chain and requires a high level enterprise effort to align all members in support of the value chain. DoD does not appear to have a value chain for software. We would like to change that by highlighting the benefits of such a change, however, we cannot ignore the fact that intra-agency integration for software is an issue as well. Below is a notional timeline for SPAWAR San Diego's Software External Release Process.

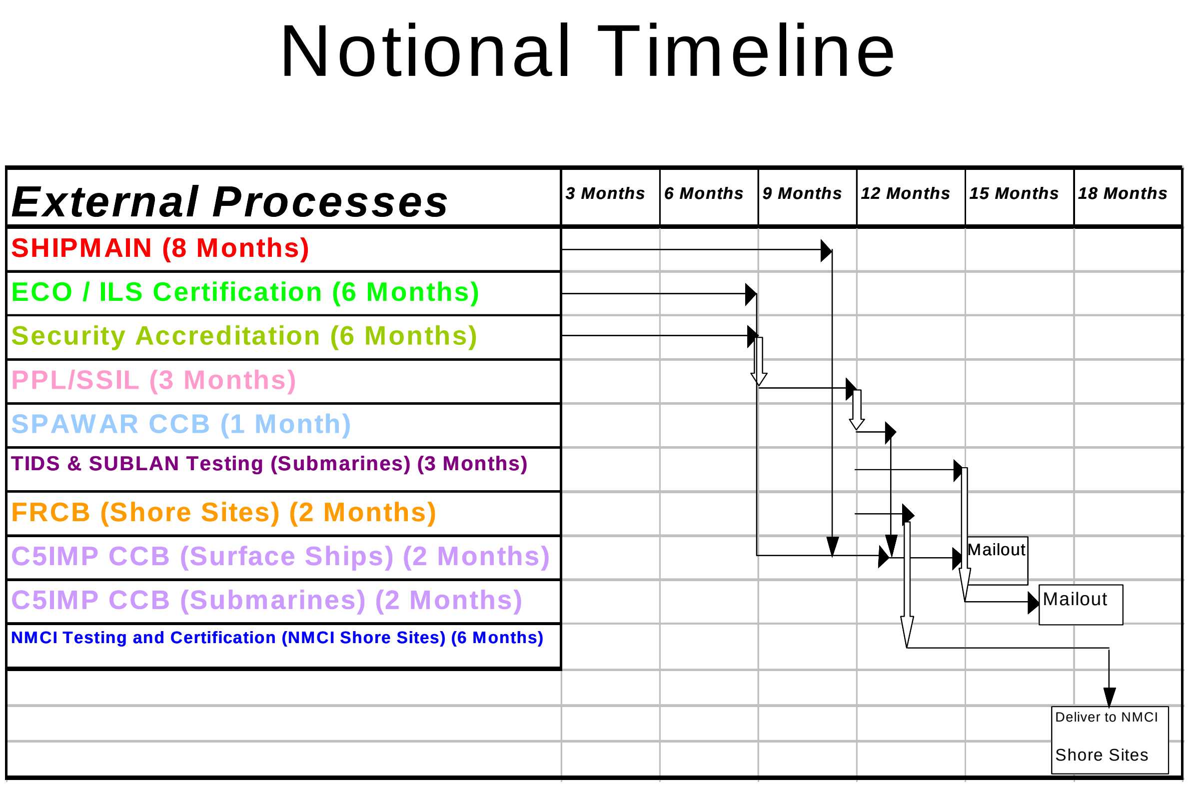

Figure 4. SPAWAR External Release Process Notional Timeline

The key actors in the External Software Release Process are SPAWAR, NAVSEA, and COMNETWARCOM. SPAWAR is the lead agent and responsible actor for getting quality systems to the fleet. As such, the processes or services that require the most time are external to SPAWAR. One process is the DoD Information Technology Security & Accreditation Process (DITSCAP) or the pending update to DITSCAP – the DoD Information Assurance Certification and Accreditation Process (DIACAP). This process deals with the certification and accreditation of information technology (IT) systems. The second process is the Surface Ship and Aircraft Carrier Modernization Program (SHIPMAIN). SHIPMAIN was developed to focus on the early decision process regarding which alterations are to be accomplished and add discipline into the accountability and adherence to the Fleet Modernization Plan associated processes. The SHIPMAIN process ensures Configuration Management, Certification and Accreditation, other required certifications, and funding are aligned and in place.

SHIPMAIN, through the use of the Navy Data Environment (NDE) and the one master electronic document for maintenance – the Ship Change Document (SCD), adds much needed discipline and oversight to ensuring certifications are met before fielding systems to the fleet. The SHIPMAIN process has three key phases that are repeated three times similar to the milestone decisions in the Defense Acquisition Cycle. (Appendix A). These three phases, however, do not happen in parallel with the acquisition cycle but at the end. Key areas like Technical Assessment have large non value added delay due to resource constraints. For software and hardware maintenance actions, the technical assessment primarily consists of a ship drawings check and confirmation by the Ship Platform Manager (SPM) that they approve of the system. The delay is simply due to the large amount of actions and a small office. A second source of non value added delay is the voting system at the three different decision points. Once Technical Assessment, Cost Benefit Analysis, and Armed Figure of Merit are complete, the SCD is then emailed out to approximately 50-60 voting members, mostly on the East Coast. If someone departs on leave or does not vote right away, the process stops. The submitter must wait until that person returns from leave or contact a dissenting voter to resolve the issue. Common concerns were cases of people on leave with no identified relief who could move the SCD through the decision point. A second concern was random office codes stopping the SCD due to misunderstanding of roles and responsibilities.

SHIPMAIN is a relatively new process and has shown positive benefits for the prioritization of funds and maintenance effort. Unfortunately, most software and hardware upgrade cycles are 6 months to a year in duration. The added certifications for processes such as SHIPMAIN and DITSCAP take at least a year to eighteen months. This means the fleet will always be behind in already mature technology. Better integration and coordination of processes are needed.

One possible solution is to regionalize software similar to the maintenance regionalization effort, and identify software as an enterprise value chain with SPAWAR as the lead regional agent with collocated NAVSEA and COMNETWARCOM cross functional teams. The objective of the cross functional teams would be to provide DITSCAP and SHIPMAIN services at a more integrated level with current software processes.

4. Funding Instability

Congressional, OSD, and service related program assessments, nonrecurring decrements, and other budget reductions, have an impact on acquisition programs. These reductions require PEO's to prioritize their requirements and reprogram resources to meet mission requirements. PEOs and PMs attempt to mitigate the impact of budget reductions through internal reprogramming and other actions to keep all programs healthy.

Stabilized Funding is the largest concern among project managers and is identified as the strongest enabler for accurately predicting program cost, schedule and performance. The lack of stable funding creates discontinuity in the system which leads to a new learning curve and startup time. This new learning curve, in turn, creates an unpredictable cost and schedule, and degrades the ability of management to develop an effective Acquisition Strategy. An ineffective Acquisition Strategy leads to a loss of confidence by Senior Leadership, who then applies more intervention and more oversight. Budget and schedule adjustments are made, with the staffing profile adjusted and set according to requirements, schedule, and budget. Funding profile changes once again and the cycle of instability continues. The best way to minimize this effect is by programs or increments of capabilities that can be delivered quickly, that is, in less than five years (GAO, 2006).

It should be noted that stabilized funding is not suitable for all programs. If this were the case, programs that do not meet desired objectives would still acquire funding and be fielded with no added capabilities. Stabilized funding should only be applied to programs that meet the desired measures of performance for desired capability objectives. If the desired capability changes, then subsequent funding should also change. DoD is moving in this direction with portfolio management, but it is still hard to find and end programs that don't meet desired capability objectives. This is due to the platform centric nature of acquisition and program competition to stay alive. Competition is good for nurturing performance but creates barriers for information sharing and visibility into the program. A domain structure with measures of effectiveness for capabilities and performance can alleviate this problem as can incentives for domain performance vice program performance. Programs that meet goals and have the highest measure of effectiveness for the DoD enterprise or component are ensured stable funding. Programs that don't meet these criteria must compete for remaining funds based on actual program performance.

5. Industry Behavior

Industry behavior is motivated largely by quarterly earnings and market share. DoD must take this into account for setting acquisition strategy and contracting policies. Firms will pull and push personnel to other projects in order to meet quarterly goals and maintain market share. This can have serious implications for large scale, long term DoD projects. A key attribute is a good staffing profile with the appropriate mix of experienced developers identified early and for the life of the effort. *The Mythical Man Month* by Frederick Brooks, discusses the dynamics of adding people late to an ongoing software development effort. Adding people late makes the project later due to increased learning time and its dampening effect on productivity. The experienced developers lose productivity due to the transfer of knowledge to inexperienced developers. The delay due to inexperienced developers getting up to speed, and the reduced productivity by the experienced developers combines to slow the project even more. If Industry quarterly earnings behavior and market goals do not align with the short and long term goals of the program, this dynamic will occur and costs will grow.

We have identified Workforce Capacity, Oversight, Process Integration, Funding Instability, and Industry Behavior as common areas of concern in the Software Acquisition Process. We have also discussed the various factors and trends of the system that influence these issues. Our next step is to determine common linkages among these areas and develop a model to simulate the system for further analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DYNAMIC HYPOTHESIS

In Chapter II we discussed workforce capacity, oversight, process integration, funding instability, and industry behavior as common themes of concern in the Software Acquisition Process. We now take these key themes and formulate a mental model of how these themes relate and interact.

A. MENTAL MODEL OF SOFTWARE ACQUISITION

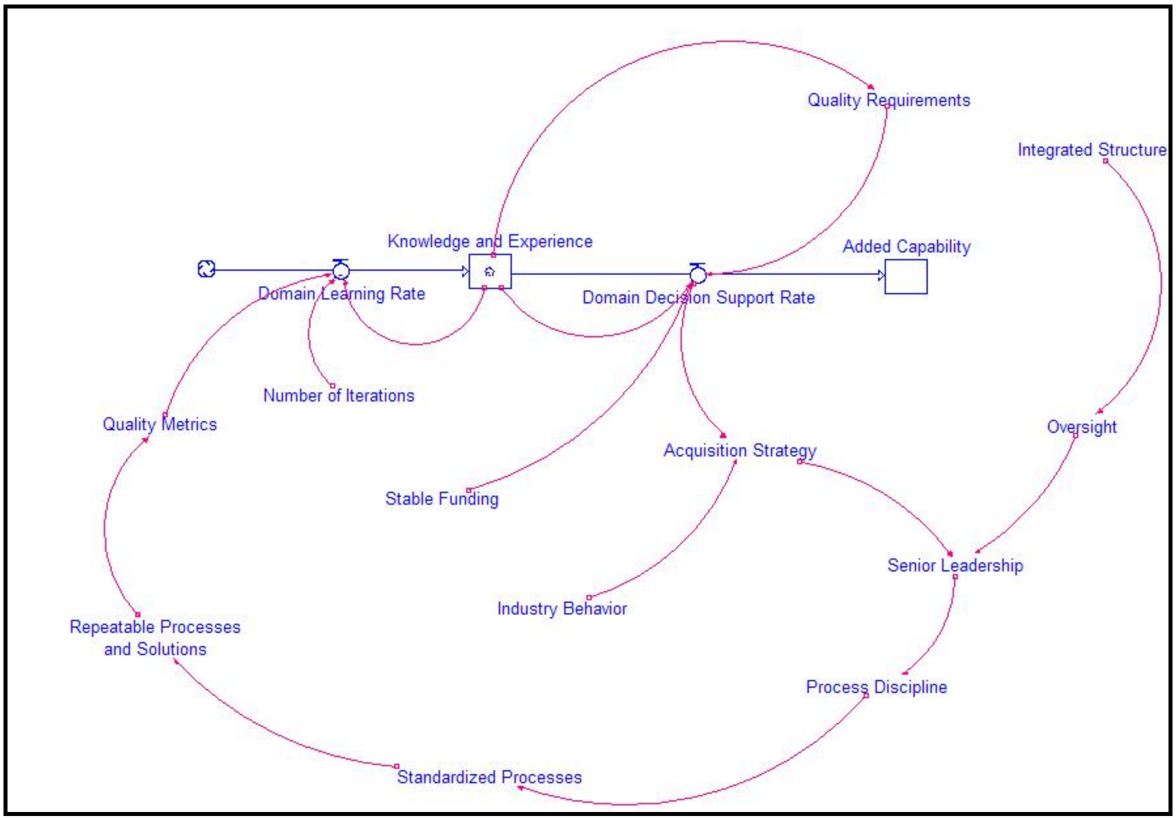

Figure 5. Mental Model of Software Acquisition

In our mental model shown in Figure 5, the following cause-effect relationships are apparent:

1) An integrated organization improves oversight. The organization can be integrated through structure, information sharing, and or social interaction. The ideal organization possesses all three of these characteristics.

2) Effective oversight leads to effective senior leadership. This relationship is primarily between OSD and the respective Program Manager (PM). However, there is both a Component Acquisition Executive and a Program Executive Officer in the hierarchy between them. Direct communication between OSD and PM is infrequent. OSD's ability to provide oversight of a large number of DoD programs is essential for ensuring organizational goals and responsibilities are communicated and carried out effectively by senior leadership.

3) Effective senior leadership ensures process discipline and standardization. Once process discipline and process standardization are achieved, processes can be repeated, measured, and optimized.

4) Optimization improves the learning rate and adds knowledge to the system for decision makers. The number of times a process is repeated, the better the learning rate and the higher the level of knowledge.

5) As domain knowledge and experience increases, decision making and formulation of future requirements improves.

6) Stable funding improves the decision process by adding certainty, reducing delays, and ensuring the timely utilization of resources.

7) Industry behavior and the organization's level of knowledge influences the chosen acquisition strategy. Two examples of acquisition strategies are Spiral Development and the Lead Systems Integrator (LSI) concept. Both of these strategies depend on industry behavior and performance. Both of these strategies are more effective if the organization has a certain level of resident knowledge to ensure oversight and to ensure requirements are met.

8) The chosen acquisition strategy influences how effective senior leadership can monitor a program. The LSI concept is an example of an acquisition strategy that makes it more difficult for senior leadership to oversee and monitor a program due to the increased responsibilities of contractor and the lower level of involvement by government officials. The lower level of involvement is due to the fact that DoD does not have the level of knowledge or capacity to oversee all aspects of the program.

The key issues in these relationships center around:

1) Domain and technical knowledge, and

2) Integration and coordination.

What policy or organizational constructs will develop positive self reinforcing mechanisms to counteract the risk inherent in rapid technological change, funding instability, decreasing workforce capacity, and uncertain or changing requirements?

One possible answer is to leverage experience curves or learning curves to shorten production times, build experience, improve schedule and budget forecasting. Learning or improvement curves were first pioneered in 1936 by T.P. Wright, who published his theory in the Journal of Aeronautical Sciences as part of an article entitled “Factors Affecting the Cost of Airplanes.” Wright’s findings showed that, as the number of aircraft produced in sequence increased, the direct labor input per airplane decreased in a regular pattern that could be estimated mathematically. (NASA, 2007)

Learning curves are typically used for estimating production costs but the same concepts can be applied for knowledge management. As the process for creating a product is repeated, people learn and develop experience with the product that leads to improvement and more knowledge gained by the system. This process will continue in a self reinforcing fashion until the product is fully mature and the maximum capacity for improvement has been reached.

In software, this can be compared to product line architectures with reusable components. For a product line architecture to be successful, there must be effective oversight and management of the domain to which the architecture will apply.

Knowledge management is a social phenomenon based upon communities of interest, shared goals, human interaction, and informal learning. Knowledge is typically categorized as either tacit or explicit knowledge. Explicit knowledge can be captured and measured in artifacts such as spreadsheets, lookup tables, and IF..THEN statements. Tacit knowledge is based on individual or group mental models and experiences. Tacit knowledge might be your personalized decision model of what to do once you’ve analyzed the spreadsheet data. Developing tacit knowledge is key for any organization but even more important for acquisition and software intensive programs, since metrics may mean different things to different programs. The metrics are important, but understanding how various metrics such as funding profile, development rate, staffing profile, schedule pressure, and technological maturity all relate to one another is even more important. This tacit knowledge is developed through a social network of peers, colleagues and personal experience. In order to fully exploit and develop knowledge based processes, organizations must either organize around knowledge (become domain centric) or develop policies to create reinforcing mechanisms for knowledge building and retention. To capitalize on the explosion of information and the new capabilities that advances in technology will bring, we must become domain centric vice program centric across DoD. The following case study demonstrates the advantages and benefits of a domain centric approach.

B. DOMAIN-CENTRIC CASE STUDY - CELSIUSTECH

CelsiusTech Systems AB is a Swedish naval defense contractor that has successfully developed a product line approach for building large, complex, software intensive systems. CelsiusTech was chosen due to the time duration and extent of research. The initial assessment was performed in 1996 by the Carnegie Mellon Software Engineering Institute with a follow on assessment in 2006.

CelsiusTech migrated to a product line architecture for a number of reasons, the principal of which was survival. In 1985, CelsiusTech (then Philips) was awarded two contracts simultaneously—one for the Swedish navy and one for the Danish navy. Both ships' requirements indicated the need for systems larger and more sophisticated than CelsiusTech's previous system. (ASWEC, 1996) Looking at past experience and the need to build two even larger systems in parallel, management and senior technical staff reevaluated their business and technical approaches. In its analysis, CelsiusTech determined that the continuation of the development technologies and practices applied on their Mk2.5 system were insufficient to produce the new systems with any reasonable certainty of schedule, cost, and required functionality. Staffing requirements alone would most likely have been prohibitive. This situation provided the genesis of a new business strategy: recognizing the potential business opportunity of selling and building a series, or family, of related systems rather than relying upon a single monolithic system. This strategic realignment resulted in the creation of the SS2000 product line.

Another business driver was the recognition of a 20- to 30-year life span for naval systems. During that time, changes in threat requirements and technology advances would have to be addressed, thus the more flexible and extendable the product line, the greater the business opportunities. These business drivers or requirements forged the product line strategy which resulted in CelsiusTech reducing their staffing profile and achieving the gains in schedule performance shown in Figure 6.

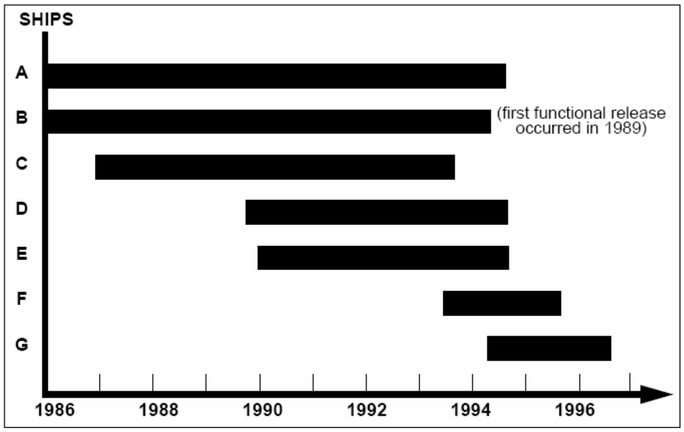

Figure 6. Celsius Tech Schedule Performance

Moving to a product line approach also required changes to the organizational structure. CelsiusTech did not shift immediately to a domain based organization, but instead went through a transition period. The following figures represent the change in organization from a project centric organization to a domain centric organization.

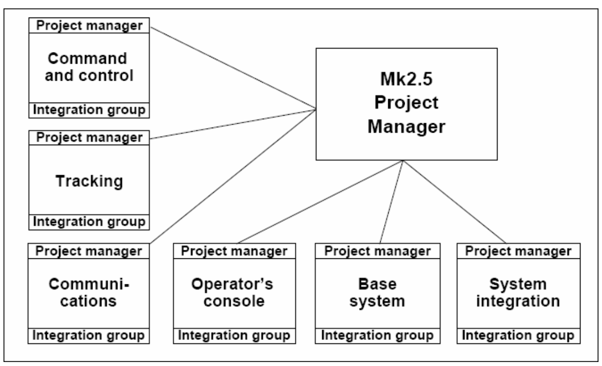

Figure 7. Celsius Tech Project Organization 1980-1985

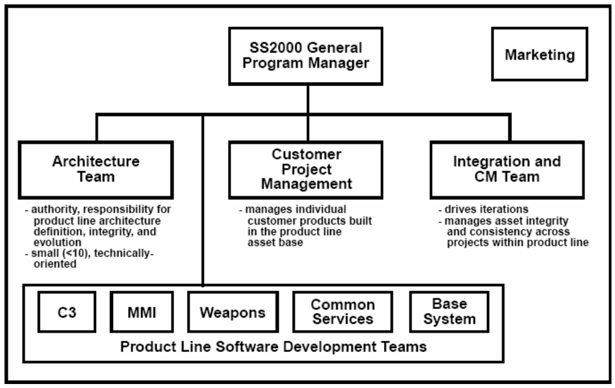

Figure 8. CelsiusTech Organization 1987-1991

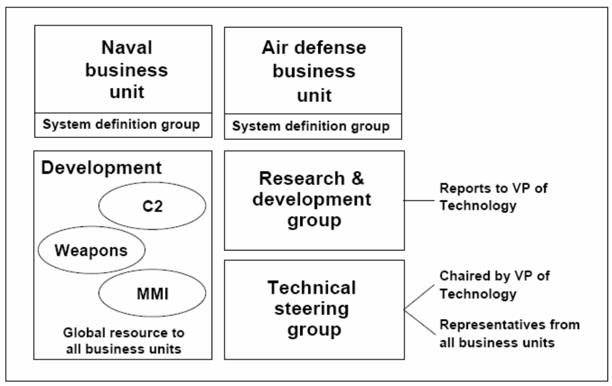

Figure 9. CelsiusTech Organization 1994-Present

As of 2006 CelsiusTech has produced the following metrics (ASWEC, 2006):

i. A family of 55 Ship Systems

ii. Integration test of 1-1.5 million SLOC requires 1-2 people.

iii. Rehosting to a new platform or OS takes 3 months.

iv. Cost and schedule targets are predictably met.

v. Performance and distribution behavior is known in advance.

vi. Customer satisfaction is high.

vii. Hardware-to-software cost ratio changed from 35:65 to 80:20.

CelsiusTech's experience suggests a viable strategy for software intensive systems. DoD has realized that the service-centric approach is inefficient and has shifted to a "capabilities" concept where requirements are "born joint". (DoD, 2004) Our solutions and systems should leverage knowledge and resources across domains and be "joint" as well.

We have given an example of a successful domain-centric organization. We will now apply these concepts to DoD's Software Acquisition Process.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SYSTEM DYNAMICS MODEL

System Dynamics is an approach to understanding the behavior of complex systems that exhibit nonlinear behaviors over time. It was developed by Professor Jay Forrester at MIT for industrial dynamics and focused on such management problems as instability in production, inventory, and employment. Since then, the field of System Dynamics has been expanded to other applications such as urban growth, economic, and ecological systems. It captures internal feedback loops and time delays that affect the behavior of the entire system. The system uses feedback loops, stocks and flows to assist in describing nonlinear, dynamically complex systems. There is no intrinsic limit on the size or complexity of systems that can be modeled; they may have thousands of elements or actors. The computer software creates an environment to run "what if" simulations to test policies and their effects as they change over time.

Developing a systems dynamics model starts with identifying the problem of a system and the underlying cause and effect relationships. These cause and effect relationships are then transformed into a mental model of the system. These mental models are described and documented as causal loop diagrams that represent the system with feedback loops. From the causal loop diagram, equations or graphical relationships are formulated that put the model into consistent units and scale. Now with the tweak of a sensitivity level, a user can learn more about the stocks that impact the problem domain's big picture. Given the appropriate data, it is a useful tool to simulate and predict a stock's aggregate interactions in a specific problem domain.

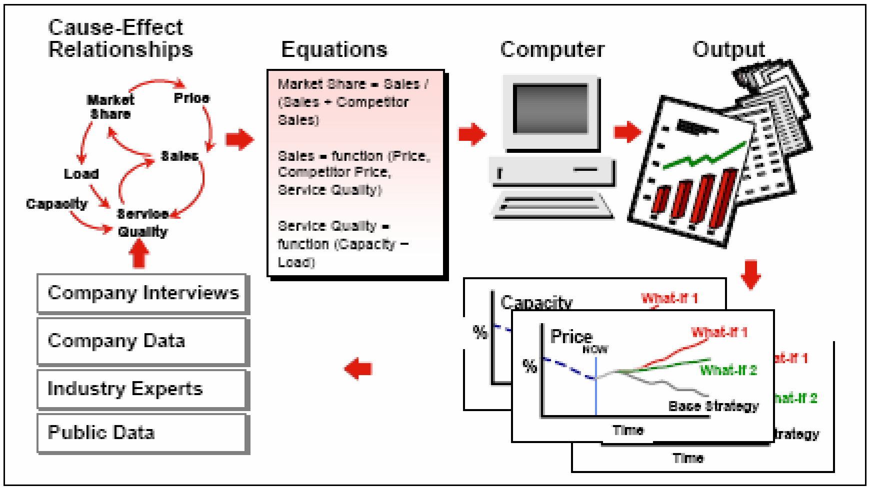

Figure 10. The Basic Steps for System Dynamics Modeling

A. MODEL FORMULATIONS

In Chapter III, we discussed our mental model of the system. We now want to apply our mental model to the actual Software Acquisition Process. We will do this by first developing a submodel to represent domain knowledge building. We will then apply our knowledge management submodel to the knowledge points of the acquisition cycle. To track cost performance, we will also create a submodel to track budgeted costs versus actual costs of the work flowing through the system.

B. THE KNOWLEDGE MANAGEMENT SUBMODEL

The system dynamics model we are developing represents organizational learning throughout the Software Acquisition Process. People and organizations learn to a certain carrying capacity or ceiling. The carrying capacity depends on the nature of the community and the maturity of the domain. For our model, we will use Technology Readiness Levels (TRL) as a measure of knowledge for each domain. A TRL value of 9 represents a system proven in successful mission operations. A final TRL value of 10, which is one level beyond Figure 11, will be used to represent a stable technology with high levels of organizational knowledge due to the fact the system has been in operation for some time and optimized with feedback from users. We will set the domain's carrying capacity and knowledge ceiling to this value of 10.

Technology Readiness Level is a measure used by some United States government agencies and many major world's companies (and agencies) to assess the maturity of evolving technologies (materials, components, devices, etc.) prior to incorporating that technology into a system or subsystem. Generally speaking, when a new technology is first invented or conceptualized, it is not suitable for immediate application. Instead, new technologies are usually subjected to experimentation, refinement, and increasingly realistic testing. Once the technology is sufficiently proven, it can be incorporated into a system or subsystem. Our model will proceed with the assumption that there is a level of knowledge associated with each TRL. As the level of knowledge reaches our carrying capacity of 10, the technology is mature and our knowledge about the technology is fully mature and optimized.

Technology Readiness Levels in the Department of Defense (DOD)
(Source: DOD (2006), *Defense Acquisition Guidebook*)

Technology Readiness Level	Description
1. Basic principles observed and reported	Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Example might include paper studies of a technology's basic properties.
2. Technology concept and/or application formulated	Invention begins. Once basic principles are observed, practical applications can be invented. The application is speculative and there is no proof or detailed analysis to support the assumption. Examples are still limited to paper studies.
3. Analytical and experimental critical function and/or characteristic proof of concept	Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.
4. Component and/or breadboard validation in laboratory environment	Basic technological components are integrated to establish that the pieces will work together. This is relatively "low fidelity" compared to the eventual system. Examples include integration of 'ad hoc' hardware in a laboratory.
5. Component and/or breadboard validation in relevant environment	Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so that the technology can be tested in a simulated environment. Examples include 'high fidelity' laboratory integration of components.
6. System/subsystem model or prototype demonstration in a relevant environment	Representative model or prototype system, which is well beyond the breadboard tested for TRL 5, is tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high fidelity laboratory environment or in simulated operational environment.
7. System prototype demonstration in an operational environment	Prototype near or at planned operational system. Represents a major step up from TRL 6, requiring the demonstration of an actual system prototype in an operational environment, such as in an aircraft, vehicle or space. Examples include testing the prototype in a test bed aircraft.
8. Actual system completed and 'flight qualified' through test and demonstration	Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications.
9. Actual system 'flight proven' through successful mission operations	Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation. In almost all cases, this is the end of the last "bug fixing" aspects of true system development. Examples include using the system under operational mission conditions.

Figure 11. Technology Readiness Level Definitions

As an organization goes through multiple iterations or production cycles learning increases. (Chapter III). As experience or level of knowledge accumulates with the number of iterations, the production costs and completion time often decrease with higher quality output. It is recognized that repetition of the same operation results in less time or effort expended on that operation. For the Wright learning curve, the underlying hypothesis is that the direct labor man-hours necessary to complete a unit of production will decrease by a constant percentage each time the production quantity is doubled. If the rate of improvement is 20% between doubled quantities, then the learning percent would be 80% (100-20=80). While the learning curve emphasizes time, it can be extended to cost as well. A NASA study has found the following learning curves for different domains (NASA, 2007):

INDUSTRY	LEARNING CURVE
Aerospace	85%
Shipbuilding	80-85%
Complex machine tools for new models	75-85%
Repetitive electronics manufacturing	90-95%
Repetitive machining or punch-press operations	90-95%
Repetitive electrical operations	75-85%
Repetitive welding operations	90%
Raw materials	93-96%
Purchased Parts	85-88%

Table 1. Industry Learning Curves

We will use the logistics growth model to simulate this interaction of knowledge. The initial stage of growth in the logistics growth model is approximately exponential; then, as maturity begins, the growth slows, and at maturity, growth stops. Essentially, growth (knowledge building) will occur at an exponential rate until reaching its carrying capacity. For our purposes, the carrying capacity will be the required Technology Readiness Level (TRL).

We will further scale our model to capture the process maturity of the organization. In order to accomplish this, we must have a measure of how much value these processes create. The Software Engineering Institute Carnegie Mellon Maturity and Integration (CMMI) model provides a good tool for value assessment within an organization's processes. As an organization's CMMI level improves, productivity and quality increase, and uncertainty and risk decrease. (See Figure 12)

Maturity Level	Rating Definition	SEI CMM Definition[2]	KPAs[1]
1.	Initial	The processes are special and mostly undefined. Success depends upon the individual effort.	
2.	Repeatable	Basic project management processes to track cost, schedule and functionality. Tools are in place to repeat success achieved on analogous programs.	Requirements management, Software project planning, Software project tracking and oversight, Software subcontract management, Software quality assurance, Software configuration management
3.	Defined	The software process is organization wide and is employed by both management and engineering. The process is documented, standardized and integrated.	Organization process focus, Organization process definition, Training program, Integrated-software management, Software product engineering, Intergroup coordination, Peer reviews
4.	Managed	The detailed measures of the software process are collected, managed, quantified, understood, and controlled.	Quantitative process management, Software quality management
5.	Optimizing	The software process continuously improves by quantified feedback from the process and testing new and creative ideas and technologies.	Defect prevention, Technology-change management, process-change management

Figure 12. CMMI Maturity Levels

We will use case study data from the Data and Analysis Center for Software (DACS) database.[1] Based on nine case studies from this database, progressing up the CMMI model produces a 5.5 gain factor for Return on Investment (ROI) ratio for at least 75% of the cases. We will use this gain factor for the potential improvement in productivity and development based on an organization's respective CMMI level.

[1] The DACS is a Department of Defense (DoD) Information Analysis Center (IAC) and designated as the DoD Software Information Clearing house for state-of-the-art technical resources and information for the software community. DACS is administratively managed by the Defense Technical Information Center (DTIC) under the DoD IAC Program. DACS is technically managed by the Air Force Research Laboratory - Information Directorate (AFRL/IF). ITT Corporation manages and operates the DACS, serving as a centralized source for current, available data and information regarding Software Engineering and Software Technology. Site address: https://www.thedacs.com/ - accessed June 25, 2007.

As mentioned above, the theoretical maximum or final carrying capacity for knowledge for our model will be a value of 10 based on Technology Readiness Levels (TRL). How fast an organization reaches this maximum capacity depends on the organization's respective knowledge building rate. As the number of program iterations is increased and CMMI levels are maximized, the knowledge building rate approaches one (1) TRL knowledge unit per year. One TRL unit per year is our maximum rate for knowledge building (Maximum Fractional KM Rate). As unit production and CMMI level are minimized due to a low number of program iterations or a low CMMI level, the rate of knowledge building is minimized and the knowledge building rate approaches zero. A zero rate represents no new knowledge units over time and no advancement in Technology Readiness Level (TRL). Knowledge Loss is how much resident organizational knowledge leaves the system primarily due to the turnover of personnel. This rate is based upon the organization's retirement rate.

The model is initialized to the current knowledge level of the system (technical maturity) and represents how knowledge builds over time.

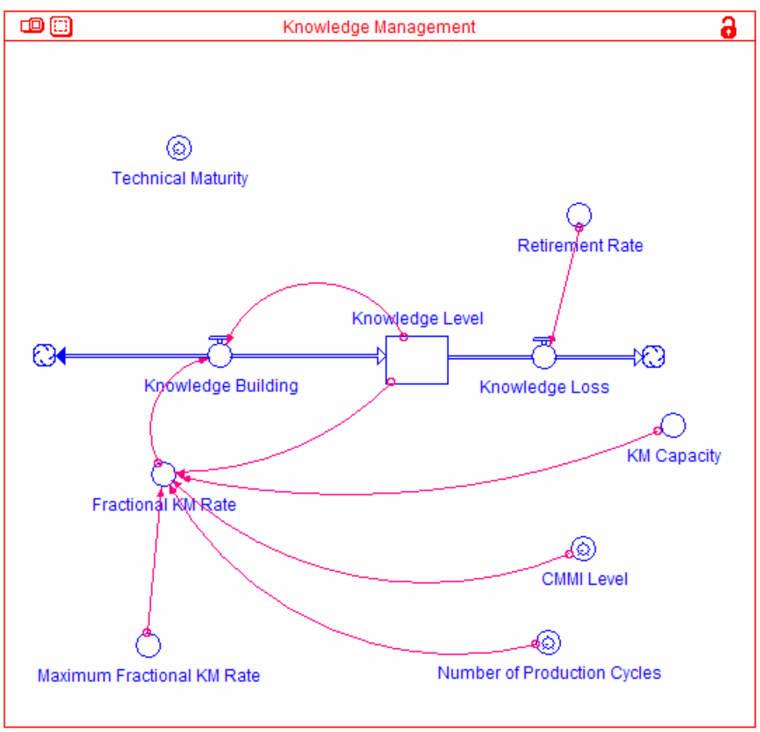

Figure 13. Knowledge Management Sub-Model

We now apply the Knowledge Management sub-model to the DoD Software Acquisition Process.

C. KNOWLEDGE MANAGEMENT APPLIED TO SOFTWARE ACQUISITION MODEL

In our model knowledge and experience directly translate to added capability. According to GAO's recent report and assessment of selected major weapons programs, this is indeed the case. GAO uses a knowledge metric for each milestone (GAO, 2007):

- Knowledge point 1: Resources and needs match. A sound business case is made for the product where customer's requirements and the developer's available resources in terms of knowledge, time, money, and capacity match. The technologies needed to meet essential product requirements have been demonstrated to work in their intended environment.
- Knowledge point 2: Product design is stable. Completion of at least 90 percent of engineering drawings at the system design review provides tangible evidence that the design is stable.
- Knowledge point 3: Production processes are mature and the design is reliable. This point is achieved when it has been demonstrated that the company can manufacture the product within cost, schedule, and quality targets. A best practice is to ensure that all key manufacturing processes are in statistical control—that is, they are repeatable.

We will take the DoD Acquisition process and apply our Knowledge Management submodels to each of the knowledge points discussed above. (See Figure 14)

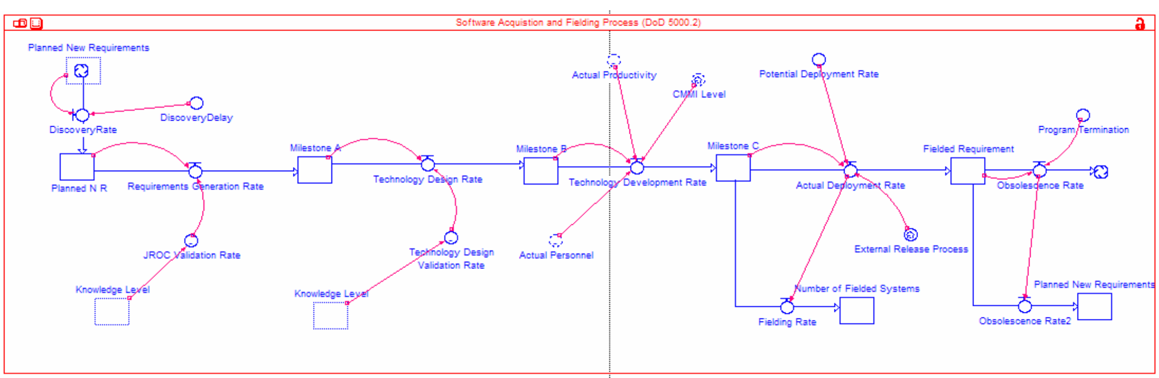

Figure 14. System Dynamics Model Representation of DoD Acquisition System

We will baseline model with the following parameters:

MODEL PARAMETER	BASELINE VALUE
Units	10 software systems *
Productivity	4 systems / 20 years
Capability Maturity Model Integration (CMMI) Level	4
Initial Technology Readiness Level (TRL)	3
Number of Production Cycles	1
Knowledge Retirement Rate	See Graphical Function (Figure 4.6)
JROC Validation Rate	See Graphical Function (Figure 4.7)

* For example, all avionics software suites for USAF and USN - 10 platforms: F- 14, F-15, F-16, F-18A, F/A-18E/F, F/A-22, EA-6B, EA-18G, F-35, and V-22

Table 2. Model Parameters

The Knowledge Retirement rate is a graphical function predicting a drop in knowledge level due to aging workforce. As the Baby Boom generation enters the retirement window, we can expect an increase from 3.5 percent to 20 percent.[2] The Baby Boom generation represents 76% of DoD's resident knowledge that must be replaced or built upon. If the Baby Boom generation left at one instant in time, it would equate to a 7.6 TRL drop in knowledge based on our model scale. The current average retirement rate is 3.5 percent. It is estimated that 20 percent will take retirement after they initially enter retirement window. Multiplying 7.6 times the current average retirement rate and future estimated retirement rate gives us our retirement rate in terms of knowledge units. (7.6 * .035 = .266 and 7.6 * .2 = 1.52) The jump from .266 to 1.52 occurs at approximately 2012 where the majority of Baby Boomers will enter the retirement window assuming a retirement age of 66.

[2] The Baby Boom generation comprises 76% of the acquisition workforce and will reach retirement window approximately 2012. (Defense Acquisition Structures and Capabilities Review, 2007).

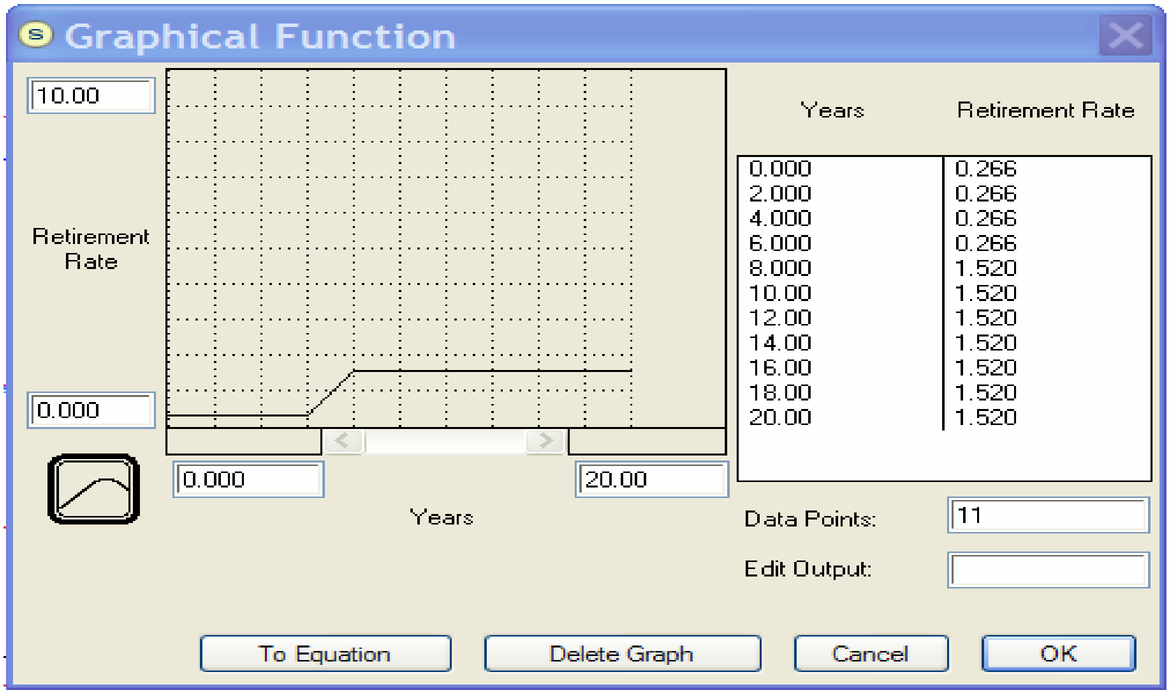

Figure 15. Graphical Representation of Retirement Rate

The JROC Validation Rate is a graphical function relating the level of requirement knowledge to process speed. A high level of knowledge and urgency of need equates to a faster requirement process.[3]

[3] An example of a high knowledge, urgent needs requirement is the infrared (IR) sight for the M1A1 tank. The M1A1 tanks had no IR sights for the .50 caliber gun system. Using the Quick Reaction Fund (QRF), an integrated IR sight for the tank system was designed, packaged, and tested. This process was done in a matter of months, and led to the retrofit by U.S. Marine Corps Systems Command of nearly 500 M1A1's in the Marine Corps inventory—all within a year. (Young, 2007).

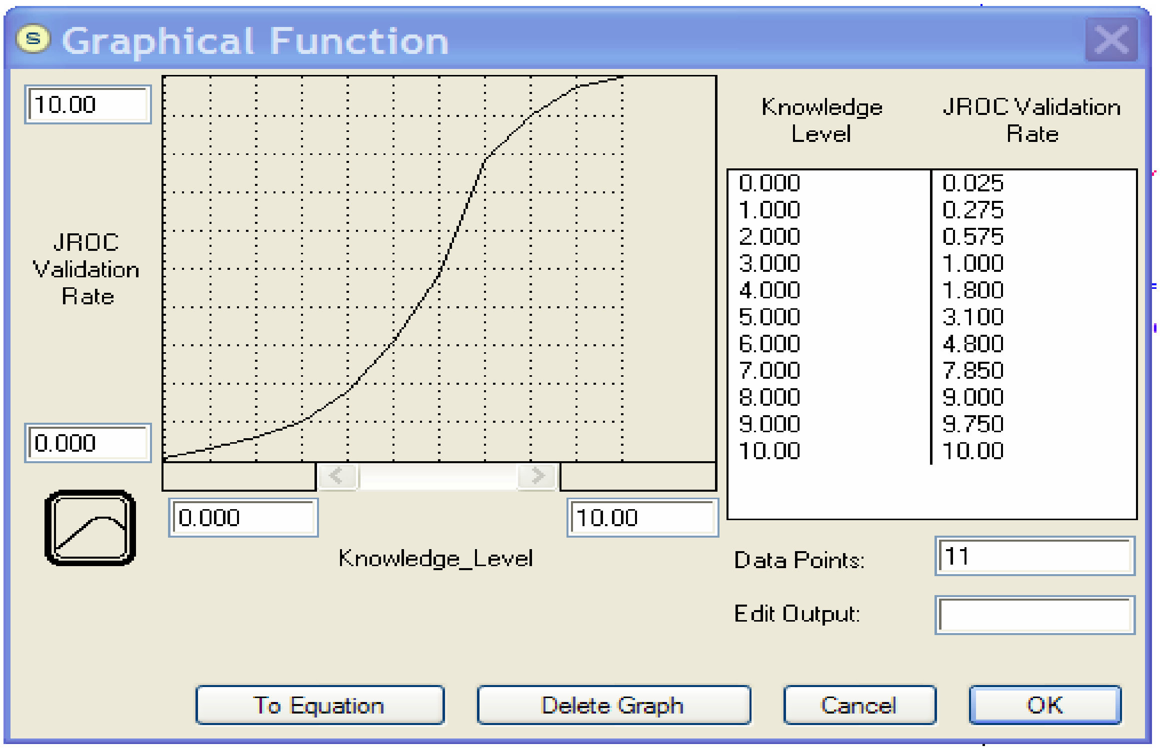

Figure 16. Graphical Representation of JROC Validation Rate

These are the key assumptions and formulations. The full list of model equations is listed in Appendix B.

D. BUDGET PERFORMANCE SUBSYSTEM

To measure budget performance we create the following sub-model. This model captures the flow of work through the system as the level of knowledge improves. As the flow of actual work is captured, a labor rate is used to keep track of resources consumed. This labor rate is applied to the actual personnel producing work to calculate the amount of resources consumed to produce actual work. Expended resources are then compared to program budget for a cost performance ratio.

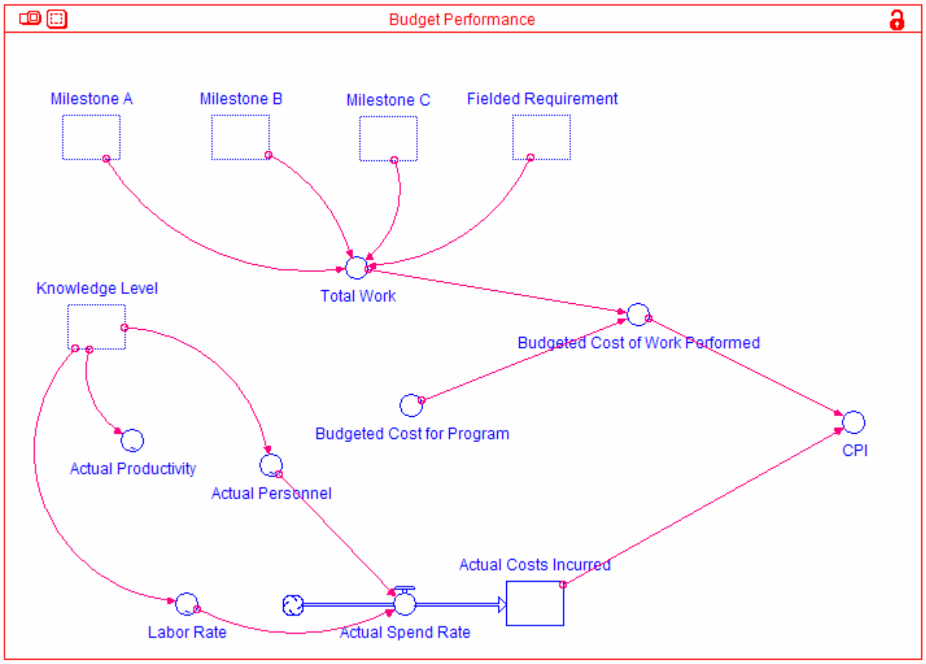

Figure 17. Sub-Model to Capture Budget Performance

We have developed the above models to relate the level of knowledge to the flow of work through the system. We will now simulate a baseline case as well as possible interventions to optimize and improve the system.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SIMULATION RESULTS

We will now take the models developed in Chapter IV and run simulations to explore and gather insights into the system. Our experiments will look at platform-centric versus domain-centric policies and how knowledge affects system output.

A. SIMULATIONS SCENARIOS

We will simulate four scenarios:

1) The Current "As-Is" System
2) The Current As-Is System with Spiral Development
3) A Proposed Domain-Centric Model
4) A Proposed Domain-Centric Model with Enterprise Integration

1. The Current System: Project-Centric Approach with Knowledge Attenuation Due to Aging Workforce

DoD's current platform-centric model often exceeds development costs by 30-40 percent with missed deadlines and performance shortfalls. (GAO, 2006). A large number of the systems in development are new and immature. Programs continue to proceed through milestone review points with low levels of knowledge. The parameters below are assumed for an immature system under development by a DoD agency.

- Capability Maturity Model Integration Level = 3.
- Initial Technology Readiness Level = 4.
- Number of Iterations: 1.
- External Release Process Delay = 2 years.

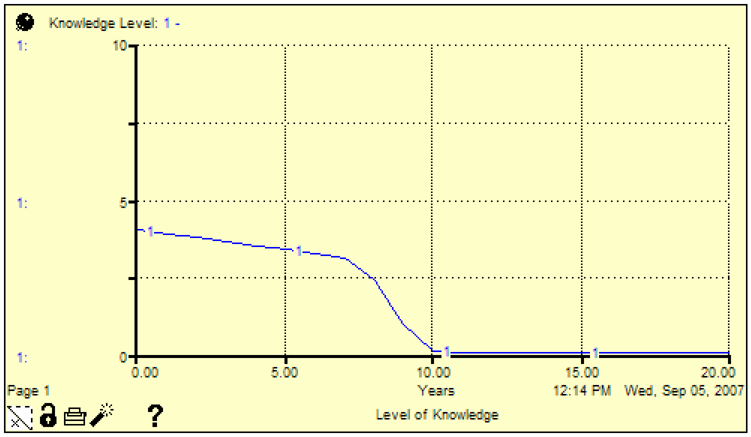

Figure 18. Simulated System Level of Knowledge

As a large complex system is developed over a number of years some knowledge is gained due to learning, but also some knowledge is lost due to personnel turnover. In Figure 18, the knowledge loss rate due to retirement of the Baby Boom generation is greater than the knowledge building rate.

Due to the low knowledge level, technical capacity and the number of fielded systems is at a low level.

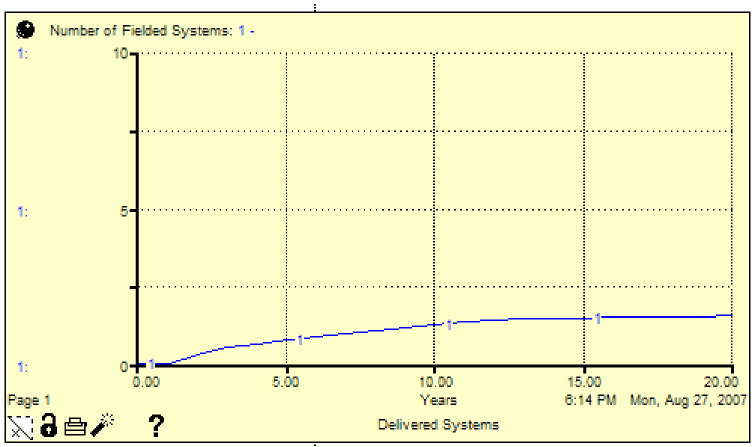

Figure 19. Delivered Systems

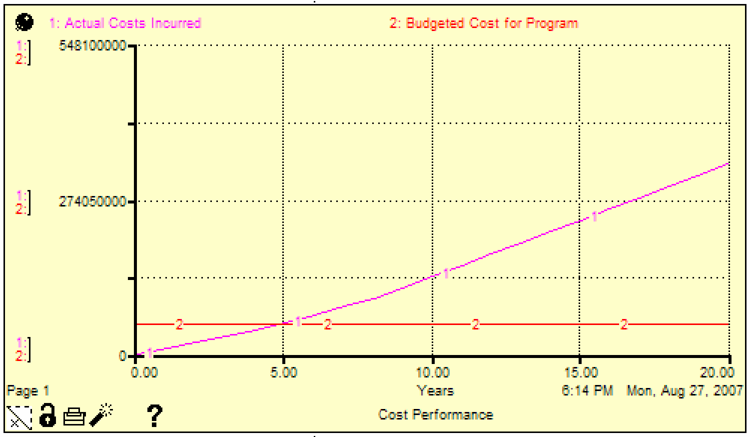

Figure 20. Budget Performance

With DoD's current platform-centric model, cost overruns typically average 30-40% with the majority of cost growth after the critical design review inherent in Milestone B. This cost growth is largely attributed to low technology readiness levels and immature technologies. (GAO, 2006) Future cost overruns are expected to be considerably higher due to the Baby Boom generation starting to leave the system at an increased rate.

The current "as-is" model suggests that the upcoming shift in workforce composition due to retirement of the Baby Boom generation will further accelerate cost overruns due to low levels of resident knowledge and reduced technical capacity.

2. Intervention #1: Low Technology with High Iteration Runs (Project-Centric with Spiral Development)

On October 30, 2002, Deputy Secretary of Defense Paul D. Wolfowitz cancelled the existing defense acquisition guidance documents DoDD 5000.1, DoDI 5000.2, and DoD 5000.2-R. In his memorandum, Wolfowitz stated that his objectives were to foster efficiency, creativity, and innovation, and to streamline mandatory acquisition procedures and processes to meet warfighter needs. The interim guidance directed that "continuous examination and adoption of innovative practices" be encouraged and that spiral development be the preferred process in any evolutionary acquisition strategy. (PM Magazine, 2003) This second simulation run represents a spiral development acquisition strategy with multiple iterations. We will capture the effects of the spiral development policy by increasing the number of iterations similar to the number of builds in a spiral development production cycle. Each iteration will represent one software build or one added increment of capability. The parameters below are assumed for this scenario: Capability Maturity Model Integration Level = 4.

- Initial Technology Readiness Level = 1.
- Number of Iterations: 10.
- External Release Process Delay = 2 years.

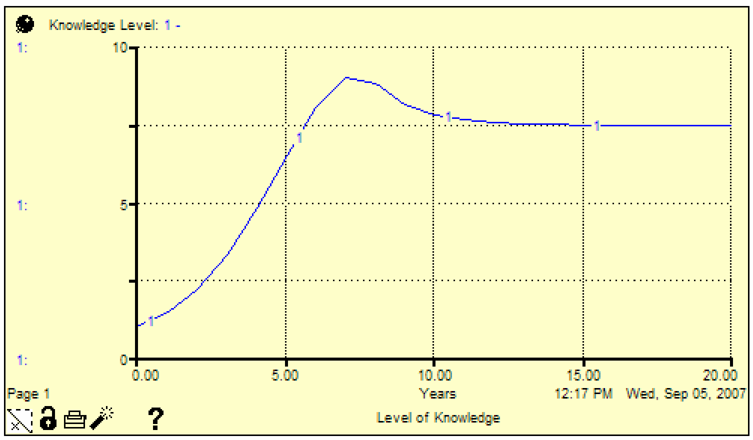

Figure 21. Intervention #1 Level of Knowledge

The high number of iterations rapidly builds the knowledge level of the technology until retirement rate starts to degrade knowledge level.

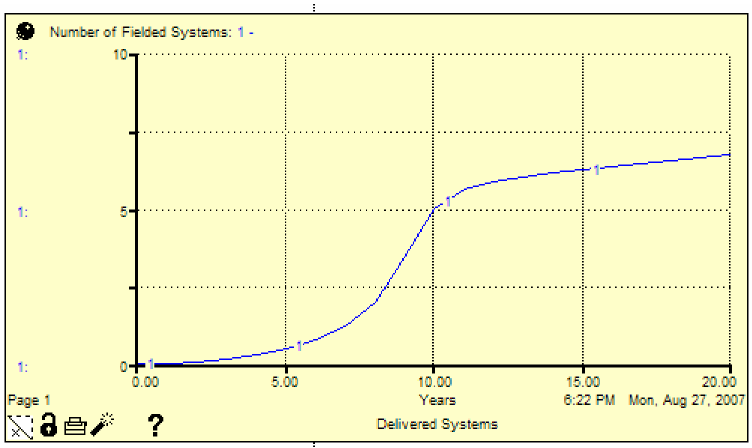

Figure 22. Intervention #1 Number of Fielded Systems

In Figure 22, technical capacity builds with each iteration until stabilizing around ten year point.

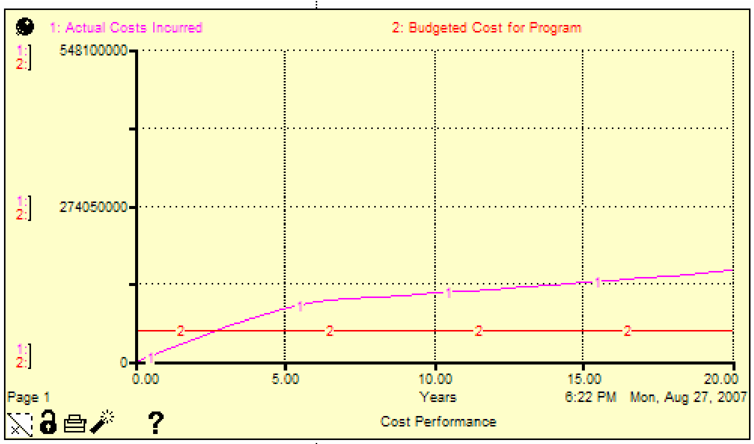

Figure 23. Intervention #1 Budget Performance

For budget performance, a cost overrun occurs due to the initial low level of technology but the rate of cost growth lessens after the five year point. The rate of cost growth minimizes due to the increased knowledge level, but the retirement rate continues to degrade knowledge which continues to result in a positive cost growth rate.

The simulation of DoD's spiral development strategy shows that from a knowledge standpoint, spiral development is an effective strategy that can be used to increase the knowledge building rate and mitigate the effects of a large retiring population. Spiral development can also be used to control costs but over a longer period.

3. Intervention #2: Stable Technology with High Iteration Runs (Domain-Centric with Product-Line Architecture)

Key success factors for software development are short time-to-market, high product quality, and lower costs. Systematic reuse throughout the development life cycle is required to achieve these goals. Systematic reuse builds technical maturity and stability by repeat usage and development. To achieve systematic reuse, the products must share a common structure or architecture. For a common structure or architecture to be effective, there must be a domain-centric view of the organization with the ability to enforce standards throughout the domain. We will capture the effects of a product-line architecture with systematic reuse by increasing the initial Technology Readiness Level (TRL) and by increasing the number of iterations. The below parameters are assumed for a domain-centric organization with a product-line architecture.

- Capability Maturity Model Integration Level = 4.
- Initial Technology Readiness Level = 7.
- Number of Iterations: 8.
- External Release Process Delay = 2 years.

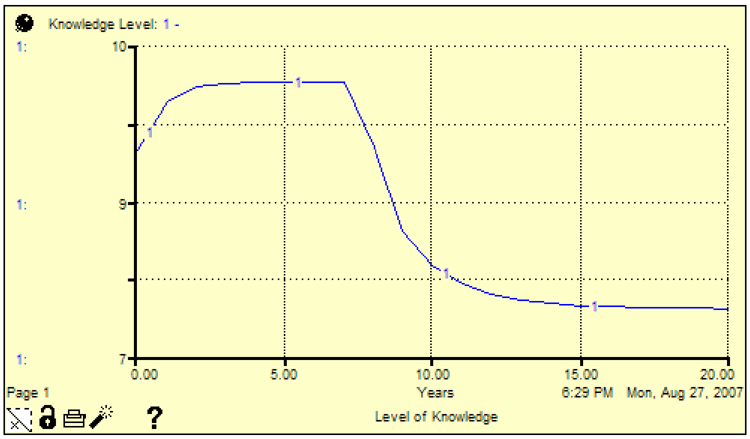

Figure 24. Intervention #2 Level of Knowledge

The knowledge building rate increases and then stabilizes at a high level until the retirement rate starts to exert its effect. The knowledge loss due to higher retirement rate reduces the knowledge level from approximately 9.5 to 7.5.

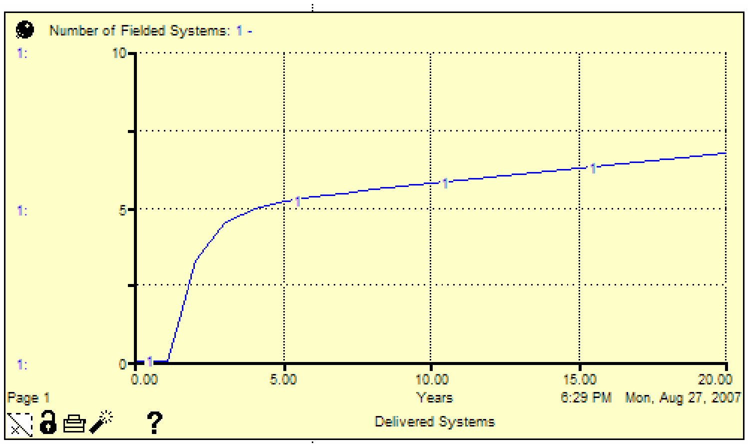

Figure 25. Intervention #2 Delivered Systems

For a domain-centric organization with a product-line architecture, the simulation shows that technical maturity and capacity rapidly grow and then stabilizes at the five year point. This is consistent with the CelsiusTech case study where after five years, CelsiusTech was able to produce the benefits of a product line architecture with shorter time to market, better cost estimation, and improved product quality.

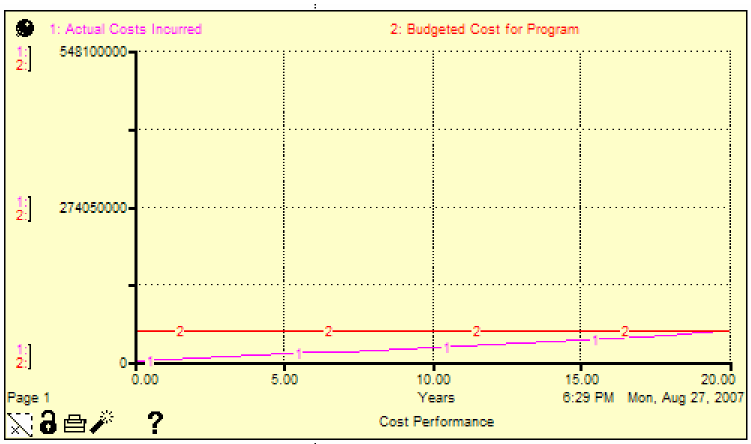

Figure 26. Intervention #2 Budget Performance

The domain-centric model reliably predicts costs and meets budget predictions even with a large drop in resident knowledge due to the retirement rate.

The simulation of a domain-centric model with a product-line architecture builds knowledge and technical capacity. The domain-centric model also mitigates the effect of the Baby Boom generation entering the retirement window. Both the Domain-centric model with a product-line architecture and the Platform-centric model with spiral development show a delta of two increments for knowledge loss. The domain-centric model, however, reaches technical maturity faster and with better cost estimation.

4. Intervention #3 – Domain-Centric Organization with Regional Software Acquisition Centers (Radical Option)

This simulation shows the effect of a highly integrated software enterprise with a product-line architecture. An end-end value stream for software is created with intra-agency services provided onsite by collocated personnel. We will capture the effects of a highly integrated and optimized software enterprise by increasing the CMMI level, maximizing the initial Technology Readiness Level, increasing the number of iterations, and by reducing process delays. The below parameters are assumed for a regionally located software enterprise with a product-line architecture in operation.

- Capability Maturity Model Integration Level = 5.
- Initial Technology Readiness Level = 10.
- Number of Iterations = 10.
- External Release Process Delay = 1.

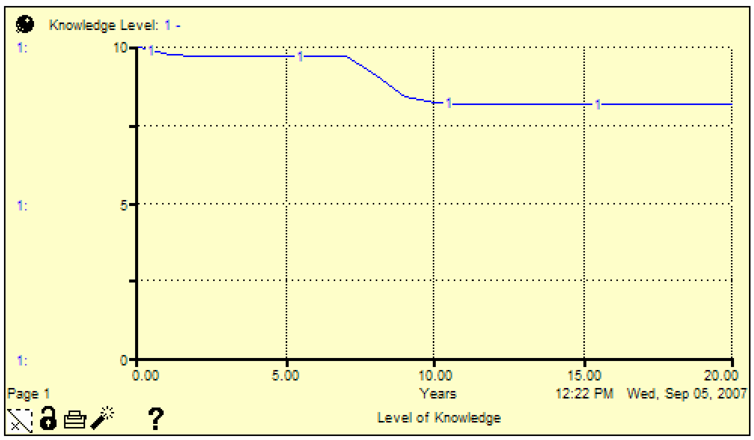

Figure 27. Intervention #3 – Level of Knowledge

The knowledge building rate is at a high level until retirement rate starts to exert its effect. The knowledge loss due to a higher retirement rate reduces the knowledge level from approximately 9.5 to 8.2.

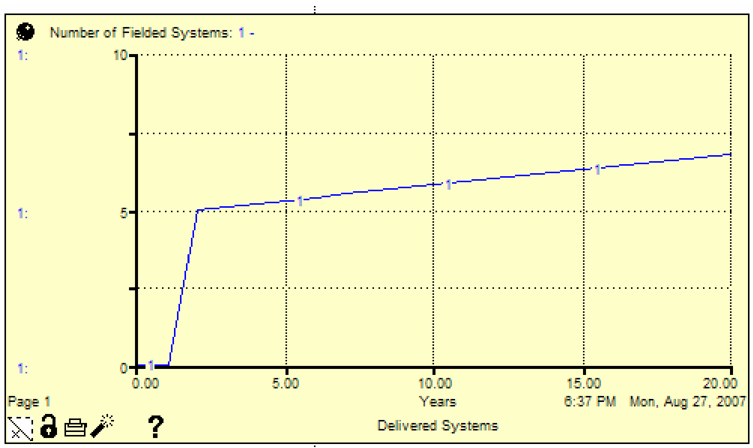

Figure 28. Intervention #3 – Delivered Systems

Due to the tighter integration of processes, delays are reduced, and technical maturity occurs in 2.5 years instead of approximately 5.0 years for a loosely coupled domain-centric organization, and 10 years for a platform-centric organization.

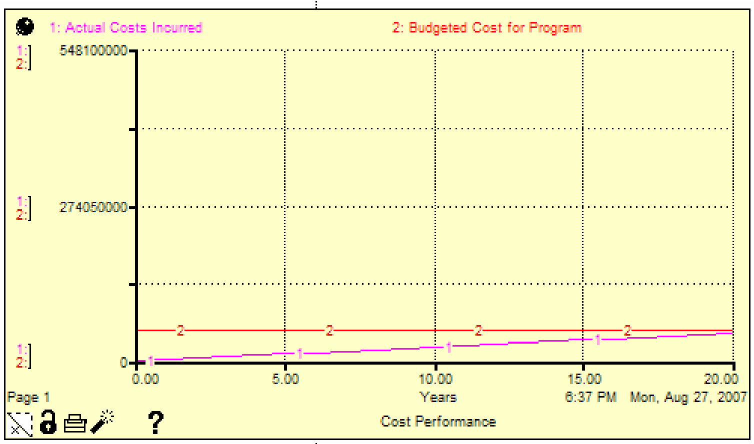

Figure 29. Intervention #3 - Budget Performance

The domain-centric, software enterprise model reliably predicts costs and meets budget predictions despite knowledge loss due to retirement rate.

B. SUMMARY OF RESULTS

Intervention	Final Knowledge Level	Tech Capacity (Fielded Capabilities)	Cycle Time for Tech Maturity	Cost Performance Index (CPI)
Baseline Case: As-Is State	0.00	1.50	10.00	0.20
Intervention #1: Project-Centric with Spiral Development	7.50	6.70	10.00	3.00
Intervention #2: Domain-Centric with Product-Line Architecture	7.50	6.70	5.00	7.10
Intervention #3: Domain-Centric with Regional Software Acquisition Centers (Radical Option)	7.50	6.80	2.50	7.60

Table 3. Summary of Results

The current baseline system will experience a significant knowledge loss with the Baby Boom generation entering the retirement window. This model does not take into account current formal training methods and their influence. It is unlikely that formal training methods will bring knowledge level up to speed as quickly as informal training methods based on actual experience. This is debatable and an area for further development and study.

Based on our simulation results, the DoD policy for Spiral Development is effective from a knowledge standpoint. Spiral Development builds technical maturity and experience of the workforce. From a project-centric standpoint, however, cycle time is not reduced by this approach and the experience gained is project or vendor specific. Also, the longer a program’s cycle time, the more influence and instability is placed on the system by turnover of personnel and financial instability. (DAPA, 2006)

A Domain-centric approach builds technical maturity, reinforces workforce experience, and improves cost performance while also reducing cycle time by a significant factor.

VI. CONCLUSIONS AND SUGGESTED AREAS FOR FUTURE RESEARCH

A. CONCLUSIONS

System Dynamics is an effective way to gather as well as explain tacit knowledge. This model is a high level, proof of concept model that has not been validated. This model can and should be expanded further to the actual program level for finer detail and discovery. The Defense Capabilities and Structure Assessment (DoD, 2005) gives the impression that DoD's past organizational changes or initiatives have no observable effect. This impression seems largely based on DoD's inability to show positive effects with any reliable metrics. If we examine commercial industry, we see that organizational structure and the ability to sense and respond to the environment is pivotal for survival and growth. The environment is surely changing for DoD and we must respond to these changes or become marginalized in technical capability.

B. FUTURE RESEARCH

There are several initiatives that may act a "tipping point" for the acquisition system. The strongest enabler is the Open Source Technology initiative. (DoD, 2006) A limiting factor for product-line development and information sharing is proprietary, vendor specific artifacts. Open source software with complementary contracting practices may break down this barrier and allow better information sharing between vendors and various communities. An area of further research would be to look at the commercial industry's incentives for open source software and determine what contracting and procurement mechanisms DoD can use to reinforce adoption of Open Source software with and between different prime contractors as well as the use of these artifacts with and between different government agencies.

Another area of further study would be to look at the level of knowledge required and the costs to maintain that level of knowledge in regards to commercially developed software versus government developed software. It is well known that Industry can develop and produce software cheaper and faster than DoD. Maintenance and lifecycle support costs are where most of the cost resides, not development. Would DoD be better served to invest in this capability and the required level of knowledge or continue to rely on Industry? These are all challenging questions that can be answered from a knowledge-costs tradeoff perspective.

APPENDIX A. SHIPMAIN PROCESS FLOW DIAGRAMS

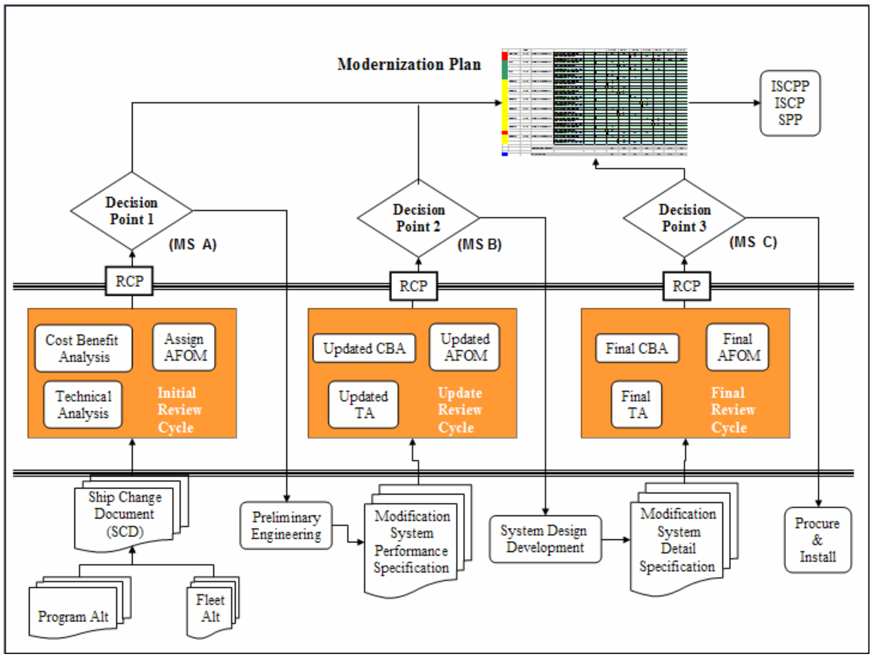

Figure 30. SHIPMAIN Process Overview

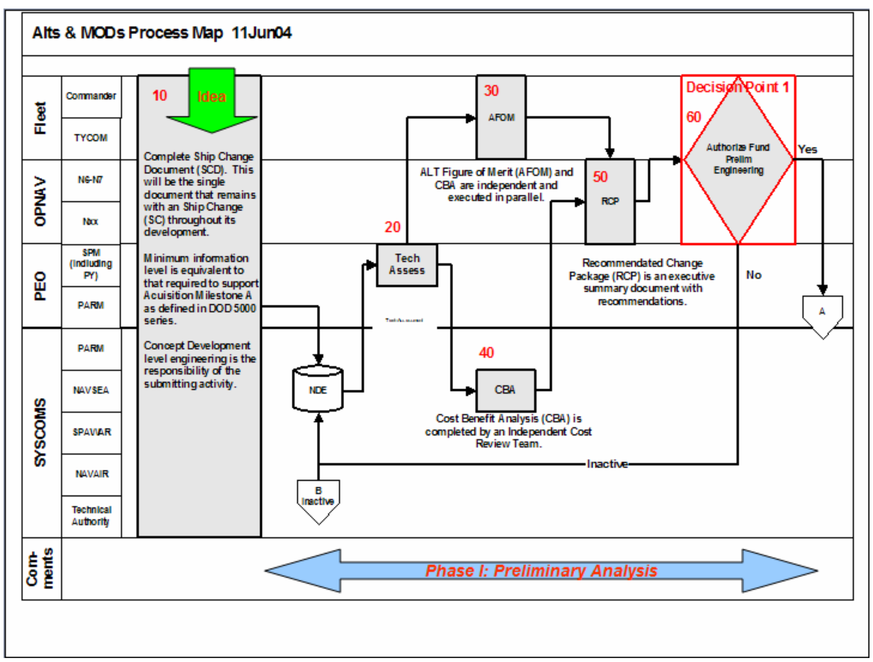

Figure 31. SHIPMAIN Phase I

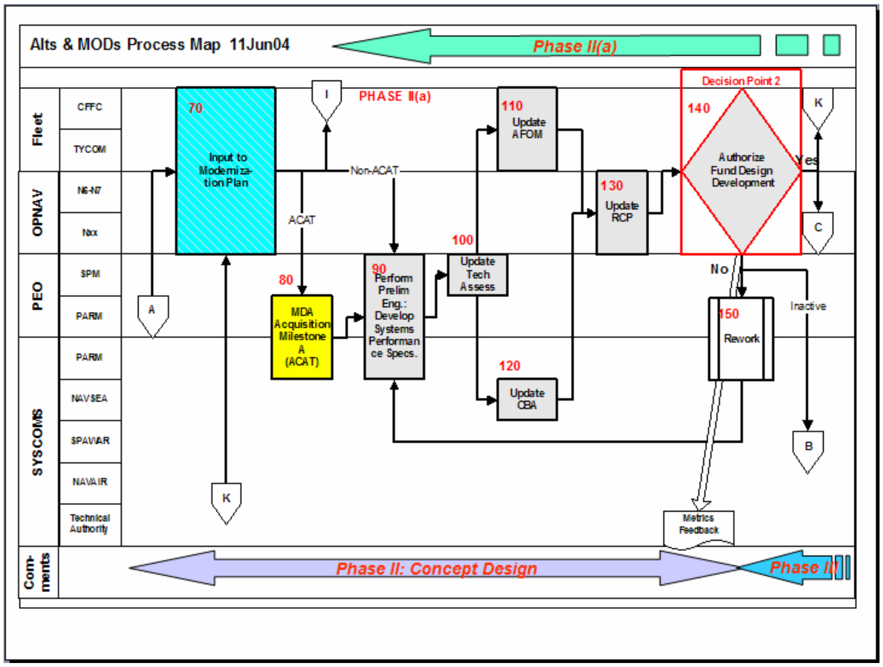

Figure 32. SHIPMAIN Phase II

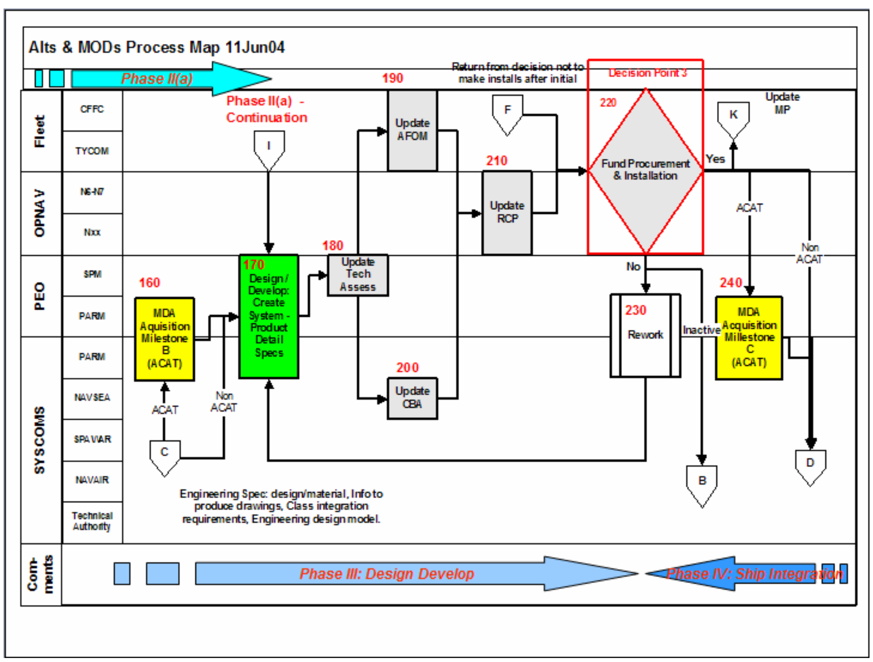

Figure 33. SHIPMAIN Phase III

APPENDIX B. MODEL EQUATIONS

Budget Performance

Actual_Costs_Incurred(t) = Actual_Costs_Incurred(t dt)
+ (Actual_Spend_Rate) * dt
INIT Actual_Costs_Incurred = 0
INFLOWS:
Actual_Spend_Rate = Actual_Personnel * Labor_Rate
Budgeted_Cost_for_Program = 140 * 365000
DOCUMENT: 3500 SLOC / 25 = 140 System Function Points
Each milestone (system function point) costs $365,000
140 System Function Points * 365,000 = $ 5,100,000 Total
Budgeted Cost for Milestone C
Budgeted_Cost_of_Work_Performed =
Total_Work*(Budgeted_Cost_for_Program/350000)
CPI = if Actual_Costs_Incurred >0 then Budgeted_Cost_of_Work_Performed /
Actual_Costs_Incurred else 1
DOCUMENT: Cost Performance Index
Total_Work = Milestone_A+Milestone_B+Milestone_C+Fielded_Requirement
Actual_Personnel = GRAPH(Knowledge_Level)
(0.00, 99.1), (1.00, 74.3), (2.00, 37.5), (3.00, 25.5), (4.00, 21.0), (5.00, 19.5),
(6.00, 15.8), (7.00, 14.3), (8.00,
12.8), (9.00, 11.3), (10.0, 10.0)
Actual_Productivity = GRAPH(Knowledge_Level)
(0.00, 366), (1.00, 429), (2.00, 508), (3.00, 634), (4.00, 760), (5.00, 949), (6.00,
1358), (7.00, 2900), (8.00,
3200), (9.00, 3365), (10.0, 3455)
Labor_Rate = GRAPH(Knowledge_Level)
(0.00, 363650), (1.00, 363200), (2.00, 362300), (3.00, 358700), (4.00, 332479),
(5.00, 301100), (6.00,
283550), (7.00, 278600), (8.00, 275900), (9.00, 275900), (10.0, 275900)
DOCUMENT: 1000 Dollars / personday
* 365 days/year = 365,000 / personyear

Knowledge Management

Knowledge_Level(t) = Knowledge_Level(t dt)
+ (Knowledge_Building Knowledge_
Leaving) * dt
INIT Knowledge_Level = Technical_Maturity
INFLOWS:
Knowledge_Building = Fractional_KM_Rate*Knowledge_Level
OUTFLOWS:
Knowledge_Leaving = Retirement_Rate

CMMI_Level = 5
Fractional_KM_Rate = Maximum_Fractional_KM_Rate * (1 (
Knowledge_Level/KM_Capacity)^(M 1))/(
M 1)
DOCUMENT: The fractional KM building rate is a growth function of knowledge relative to knowledge
carrying capacity. The parameter M from the Richards growth model determines the strengh of the
nonlinearity.
1/Period
KM_Capacity = 10
1
M
= (((10/Number_of_Production_Cycles) + (5/CMMI_Level)) / 2) + .1
DOCUMENT: Strength of Learning Curve determined by Unit Theory and Process Value Assessment (CMMI
Model)
Maximum_Fractional_KM_Rate = 1
DOCUMENT: The maximum fractional growth rate is set to 1, thus scaling time so that 1 time unit = 1/g*
(yielding the standard logistic curve).
1/Period
Number_of_Production_Cycles = 10
DOCUMENT: # Projects / Time Period Application
of Unit Theory Principles and Product Line Approach
Technical_Maturity = 1
Retirement_Rate = GRAPH(TIME)
(0.00, 0.15), (1.00, 0.6), (2.00, 0.85), (3.00, 1.05), (4.00, 1.45), (5.00, 2.40), (6.00, 4.75), (7.00, 4.75), (8.00,
4.75), (9.00, 1.95), (10.0, 1.70)

Software Acquistion and Fielding Process (DoD 5000.2)

Added_Capability(t) = Added_Capability(t dt)
+ (Fielding_Rate) * dt
INIT Added_Capability = Min(Milestone_C/dt,Actual_Deployment_Rate)
INFLOWS:
Fielding_Rate = Actual_Deployment_Rate
Build1(t) = Build1(t dt)
+ (Spiral_Development) * dt
INIT Build1 = 0
INFLOWS:
Spiral_Development = Technology_Development_Rate
Fielded_Requirement(t) = Fielded_Requirement(t dt)
+ (Actual_Deployment_Rate Obsolescence_
Rate Obsolescence_

Rate2) * dt
INIT Fielded_Requirement = 0
INFLOWS:
Actual_Deployment_Rate = if (Percent_Complete >= .80) Then
Min(Milestone_C/dt,(Potential_Deployment_Rate/(External_Release_Process +
1))) else 0
OUTFLOWS:
Obsolescence_Rate = Fielded_Requirement /Program_Termination
Obsolescence_Rate2 = Obsolescence_Rate
Milestone_A(t) = Milestone_A(t dt)
+ (Requirements_Generation_Rate Technology_
Design_Rate) * dt
INIT Milestone_A = 0
DOCUMENT: 1 System Function can have anywhere from 5K 25K
lines of ADA code: 25KSLCO per
System Function
140 maximum System Functions for CelsiusTech Case Study
Max Total Effort for (1) System = 140 x 25KSLOC = 3500 KSLOC
INFLOWS:
2
Requirements_
Generation_Rate = CONVEYOR OUTFLOW
TRANSIT TIME = Planned_N_R/JROC_Validation_Delay
OUTFLOWS:
Technology_Design_Rate =
MIN(Milestone_A/dt,Technology_Design_Validation_Rate)
Milestone_B(t) = Milestone_B(t dt)
+ (Technology_Design_Rate Technology_
Development_Rate) * dt
INIT Milestone_B = 0
INFLOWS:
Technology_Design_Rate =
MIN(Milestone_A/dt,Technology_Design_Validation_Rate)
OUTFLOWS:
Technology_Development_Rate =
MIN(Milestone_B/dt,Actual_Productivity*Actual_Personnel*CMMI_Level/2.5)
Milestone_C(t) = Milestone_C(t dt)
+ (Technology_Development_Rate Actual_
Deployment_Rate Fielding_
Rate) * dt
INIT Milestone_C = 0
DOCUMENT: 140 maximum System Functions for CelsiusTech Case Study
1 System Function can have anywhere from 5K 25K
lines of ADA code: 25KSLCO / System Function
Max Total Effort = 140 x 25KSLOC = 3500 KSLOC
INFLOWS:

Technology_Development_Rate =
MIN(Milestone_B/dt,Actual_Productivity*Actual_Personnel*CMMI_Level/2.5)
OUTFLOWS:
Actual_Deployment_Rate = if (Percent_Complete >= .80) Then
Min(Milestone_C/dt,(Potential_Deployment_Rate/(External_Release_Process +
1))) else 0
Fielding_Rate = Actual_Deployment_Rate
Planned_N_R(t) = Planned_N_R(t dt)
+ (Gen_of_New_Req Requirements_
Generation_Rate) * dt
INIT Planned_N_R = Planned_New_Requirements
TRANSIT TIME = varies
INFLOW LIMIT = INF
CAPACITY = INF
INFLOWS:
Gen_of_New_Req = Unplanned_New_Requirements/Discovery_Delay
OUTFLOWS:
Requirements_Generation_Rate = CONVEYOR OUTFLOW
TRANSIT TIME = Planned_N_R/JROC_Validation_Delay
Planned_New_Requirements(t) = Planned_New_Requirements(t dt)
+ (Obsolescence_Rate2) * dt
INIT Planned_New_Requirements = 1500
INFLOWS:
Obsolescence_Rate2 = Obsolescence_Rate
Unplanned_New_Requirements(t) = Unplanned_New_Requirements(t dt)
+ (Gen_
of_New_Req) * dt
INIT Unplanned_New_Requirements = 2000
OUTFLOWS:
Gen_of_New_Req = Unplanned_New_Requirements/Discovery_Delay
Average_Project_Size = 3500
3
Discovery_
Delay = 1
DOCUMENT: Delay Time to Identify New Unplanned Requirements
(Years)
External_Release_Process = 2.5
Percent_Complete = Build1/Average_Project_Size
Potential_Deployment_Rate = 3500/1
Program_Termination = 10
JROC_Validation_Delay = GRAPH(Knowledge_Level)
(0.00, 70.0), (1.00, 158), (2.00, 280), (3.00, 420), (4.00, 770), (5.00, 2205), (6.00,
2660), (7.00, 3028), (8.00,
3255), (9.00, 3465), (10.0, 3483)
Technology_Design_Validation_Rate = GRAPH(Knowledge_Level)

(0.00, 3535), (1.00, 3780), (2.00, 4060), (3.00, 4375), (4.00, 4830), (5.00, 5775), (6.00, 8785), (7.00, 9590),
(8.00, 10045), (9.00, 10185), (10.0, 10500)

Not in a sector

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Beaugureau, D., Hashimoto, C., & Herrin, R. (2003). Interoperability Testing and the New Acquisition Guidance. PM Magazine, 15.

Bergey, J., & Goethert, W. (2001) Developing a Product Line Acquisition Strategy for a DoD Organization: A Case Study. (CMU/SEI-2001-TN-021). Pittsburgh, PA: Carnegie Mellon University Software Engineering Institute.

Brownsword, L., & Clements, P. (1996). A Case Study in Successful Product Line Development. (CMU/SEI-96-TR-016). Pittsburgh, PA: Carnegie Mellon University Software Engineering Institute.

Dibert, J., & Velez, J. (2006). An analysis of earned value management implementation within the F-22 system program office's software development (NPS-AM-06-052).Monterey, CA: Naval Postgraduate School.

Dillard, J. (2003).Centralized Control of Defense Acquisition Programs: A Comparative Review of the Framework from 1987 - 2003 (NPS-AM-03-003). Monterey, CA: Naval Postgraduate School.

DoD. Contract Pricing Reference Guides. Update retrieved July 2007, from http://www.acq.osd.mil/dpap/contractpricing/vol2chap7.htm.

DoD. (2006). Defense Acquisition Performance Assessment Report.

DoD.(2007). Defense Acquisition Structures and Capabilities Review Report.

DoD. (2007). DoD and DNI Chief Information Officers Establish Shared Vision for a Joint Services Based Environment to Enable Information Sharing. Retrieved August 2007 from http://www.defenselink.mil/releases/release.aspx?releaseid=11146.

DoD. (2006). Implementing the Net-Centric Data Strategy Progress and Compliance Report. DoD CIO.

DoD. (2004). Joint Defense Capabilities Study - Improving DoD Strategic Planning, Resourcing and Execution to Satisfy Joint Capabilities.

DoD. (2006). Open Technology Development Roadmap Plan.

DoD. (2003). Systemic Analysis of Software Intensive System Acquisition Issues. Software Technology Conference.

DoD. (2005, May). Technology Readiness Assessment (TRA) Deskbook.

GAO. (2007). Defense Acquisitions: Assessment of selected weapons programs (GAO-06-391). Washington, DC: General Accountability Office.

GAO. (2007). Defense Acquisitions: Role of Lead System Integrator on Future Combat Systems Program Poses Oversight Challenges (GAO-07-380). Washington, DC: General Accountability Office.

GAO. (2007). Tactical Aircraft: DoD Needs a Joint Integrated Strategy (GAO-07-415). Washington, DC: General Accountability Office.

Karoly, L., & Panis, C. (2004). The 21st Century at Work: Forces Shaping the Workforce. Santa Monica, CA: RAND.

Kroecker, S. (2007). Developing Future Program Leaders: Part 1. Defense AT&L: January-February 2007, p. 12.

NASA. (2007). Learning Curve Calculator. Retrieved July 2007 from http://cost.jsc.nasa.gov/learn.html.

PCAST. (2004). President's Council of Advisors on Science and Technology: Sustaining the Nation's Innovation Ecosystem.

Young, J. (2005). Statement of the Honorable John J. Young, Jr. Assistant Secretary of the Navy (Research Development and Acquisition) Before the House Armed Services Committee on Acquisition in the Department of Defense. Washington, DC: HASC.

Young, J. (2007). Statement of the Honorable John J. Young, Jr. Assistant Secretary of the Navy (Research Development and Acquisition) Before the House Armed Services Committee on Armed Services Committee on Terrorism, Unconventional Threats and Capabilities. Washington, DC: HASC.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Tarek Abdel-Hamid
   Naval Postgraduate School
   Monterey, California

4. Daniel Dolk
   Naval Postgraduate School
   Monterey, California

5. Richard J. Dixon
   Naval Postgraduate School
   Monterey, California