NPS-AM-09-043

EXCERPT FROM THE
PROCEEDINGS

OF THE

SIXTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

MODELING OPEN ARCHITECTURE AND EVOLUTIONARY ACQUISITION: IMPLEMENTATION LESSONS FROM THE ARCI PROGRAM FOR THE RAPID CAPABILITY INSERTION PROCESS

Published: 22 April 2009

by

David N. Ford and John T. Dillard

6th Annual Acquisition Research Symposium of the Naval Postgraduate School:

Volume II: Defense Acquisition in Transition

May 13-14, 2009

Prepared for: Naval Postgraduate School, Monterey, California 93943

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Conference Website:
www.researchsymposium.org

Proceedings of the Annual Acquisition Research Program

The following article is taken as an excerpt from the proceedings of the annual Acquisition Research Program. This annual event showcases the research projects funded through the Acquisition Research Program at the Graduate School of Business and Public Policy at the Naval Postgraduate School. Featuring keynote speakers, plenary panels, multiple panel sessions, a student research poster show and social events, the Annual Acquisition Research Symposium offers a candid environment where high-ranking Department of Defense (DoD) officials, industry officials, accomplished faculty and military students are encouraged to collaborate on finding applicable solutions to the challenges facing acquisition policies and processes within the DoD today. By jointly and publicly questioning the norms of industry and academia, the resulting research benefits from myriad perspectives and collaborations which can identify better solutions and practices in acquisition, contract, financial, logistics and program management.

For further information regarding the Acquisition Research Program, electronic copies of additional research, or to learn more about becoming a sponsor, please visit our program website at:

www.acquistionresearch.org

For further information on or to register for the next Acquisition Research Symposium during the third week of May, please visit our conference website at:

www.researchsymposium.org

THIS PAGE INTENTIONALLY LEFT BLANK

Modeling Open Architecture and Evolutionary Acquisition: Implementation Lessons from the ARCI Program for the Rapid Capability Insertion Process

Presenter: David N. Ford received his BS and MS degrees from Tulane University, New Orleans, LA, and his PhD degree from the Massachusetts Institute of Technology, Cambridge. He is a Professor in the Construction Engineering and Management Program, Department of Civil Engineering, Texas A&M University, College Station, and serves as a Research Associate Professor of Acquisition with the Graduate School of Business and Public Policy at the US Naval Postgraduate School in Monterey, CA. Prior to joining Texas A&M, he was on the Faculty of the Department of Information Science, University of Bergen, Bergen, Norway, where he researched and taught in the System Dynamics Program. For over 14 years, he designed and managed the development of constructed facilities in industry and government. His current research interests include the dynamics of development supply chains, strategic managerial flexibility, and resource allocation policies. Dr. Ford is a member of INFORMS, ASCE, and other professional organizations.

David N. Ford, PhD, PE
Zachry Department of Civil Engineering
Texas A&M University
College Station, TX 77843-3136
Voice: 979-845-3759
Fax: 979-845-6554
Email: davidford@tamu.edu

Author: COL John T. Dillard, US Army (Ret.), managed major weapons and communications programs for most of his 26-year career in the military, including development efforts for the Javelin and Army Tactical Missile System (ATACMS) missile systems. His last assignment was head of all Defense Department contract administration in the New York metropolitan area. Dillard now serves as a senior lecturer with the Graduate School of Business and Public Policy at the US Naval Postgraduate School in Monterey, CA. He is also a 1988 graduate of the Defense Systems Management College, Program Managers' course. His current research interests include defense acquisition policy, organizational design and control, and product development. He is a member of PMI and other professional organizations.

John T. Dillard, Senior Lecturer
Graduate School of Business and Public Policy
Naval Postgraduate School
1 University Circle
Monterey, CA 93943
Voice: 831-656-2650
Mobile: 831-277-3457
Email: jdillard@nps.edu

Abstract

Providing system interoperability and evolving technologies in major DoD systems are two important acquisition challenges in preparing the military to meet current and future demands. The Acoustic Rapid COTS Insertion (ARCI) program successfully addressed many of the associated challenges. That program was studied as the basis for modeling the planned Rapid Capability Insertion Process (RCIP) approach for continuous, reduced-cost upgrading of assets. ARCI used atypical methods in the face of atypical program requirements and conditions. A previously developed acquisition program model was adapted to reflect ARCI and used for model validation. This model was then changed to reflect the basic conditions expected to be faced by RCIP programs. The model demonstrated the potential of RCIP to significantly improve program performance. However, implementation risks are identified that may degrade potential performance, including increased oversight, the use of more new development, and the resulting integration scope and risk. When incorporated into the model, these risks were shown to significantly decrease RCIP performance. Means for successfully managing the RCIP design based on the ACRI program and RCIP operations are suggested for use in addressing the identified implementation risks.

Introduction

Providing system interoperability and evolving technologies in major DoD systems are two important acquisition challenges in preparing the military to meet current and future demands. The use of legacy and other weapons platforms, joint Service solutions, the information and communication needs of Network Centric Systems (NCS), and coordination with allies in joint operations require the development of weapons systems that can operate across system, platform, and systems-of-systems boundaries. Traditional DoD acquisition approaches do not fully provide the interoperability and development speed needed to meet these demands. The continued, and in some cases accelerating, evolution of technologies continuously creates new challenges that are difficult to forecast and require fast acquisition response. Threat matrices also evolve, changing the capabilities required to meet them. Short capability improvement cycle-times are needed to respond to these moving targets for acquisition efforts. The development of an Integrated Weapons System (IWS) for surface ships is an example of a major acquisition effort to provide system (and platform) interoperability and exploit technology evolution to meet changing threats. The current work focuses on acquisition approaches to meet these challenges.

Naval Open Architecture (OA) (DAU, 2009) is a breakthrough acquisition approach that develops and facilitates the use of acquisition processes, which integrate interoperable systems that evolve with technologies, threats, and program environments (e.g., funding). OA does this through five principles: 1) modular design and design disclosure, 2) reusable application software, 3) interoperable joint warfighting applications and secure information exchange, 4) lifecycle affordability, and 5) encouraging competition and collaboration through the development of alternative solutions and sources. Evolutionary Acquisition (EA) (DAU, 2009) is a somewhat recently developed acquisition approach that uses the repeated integration of only mature-enough technologies into products to speed capability improvement for warfighters. OA and EA can act synergistically to meet their objectives. However, effective implementation is critical for success. Particularly in large, complex systems that span platforms, the successful implementation of OA and EA is not obvious or easy.

Despite their potential, OA and EA have not yet been fully developed or implemented in DoD acquisition. Previous research (Ford & Dillard, 2008; Dillard & Ford, 2007) suggests that the DoD can successfully integrate open systems and Evolutionary Acquisition. This supports the Navy’s current development of the Rapid Capability Insertion Process (RCIP) to implement Open Architecture and Evolutionary Acquisition (described later). The Navy’s Acoustic Rapid COTS (commercial off-the-shelf) Insertion program (ARCI) experience (described later) demonstrates that these approaches can be integrated and applied successfully. An improved understanding of how OA and EA have been used successfully and can be used in RCIP is needed to better apply them across systems and platforms and, thereby, improve acquisition programs.

The Research Approach

Evolutionary Acquisition and open systems approaches combine to create a complex set of development processes that evolve over time. An improved understanding of these processes and their management is available through formal modeling of the most important components and relationships that drive system performance and risk. Due to the number and complexity of the components and their relationships, the formal model structure and rigor of calculations can simulate and forecast performance and risk better than informal, tacit predictions by humans. Therefore, we applied a computational experimentation approach to investigating Evolutionary Acquisition and open systems projects, integrating theory and practice in a computational tool that allows controlled experimentation through simulation.

Previous research and modeling of Open Architecture and Evolutionary Acquisition is being used as the foundation of the current work. That model was first revised and improved to reflect the ARCI program to develop a basis for understanding success factors in OA and EA implementation. This required the development of a deep understanding of the relevant aspects of the ARCI acquisition program (summarized next). The ACRI model was then revised to reflect the Rapid Capability Insertion Process. Model analysis was used to better understand the requirements for success in RCIP.

The System Dynamics Modeling Methodology

The system dynamics methodology was applied to model the ARCI program. System dynamics is one of several established and successful approaches to systems analysis and design (Flood & Jackson, 1991; Lane & Jackson, 1995; Jackson, 2003). The methodology has been extensively used for this purpose, including to study several aspects of development projects. System dynamics shares many fundamental systems concepts with other systems approaches, including emergence, control, and layered structures. Therefore, system dynamics can address issues such as risk in large complex systems such as the DoD acquisition projects (Lane, Größler & Milling, 2004). The methodology’s ability to model many diverse system components (e.g., work, people, money, information), processes (e.g., design, technology development, quality assurance, rework), and managerial decision-making and actions (e.g., forecasting, resource allocation) makes it useful for investigating acquisition programs. Forrester (1961) develops the methodology's philosophy, and Sterman (2000) specifies the modeling process with examples and describes numerous applications.

The system dynamics methodology applies a control theory perspective to the design and management of complex human systems. The perspective focuses on how the internal structure of a system impacts managerial behavior and performance over time. The system dynamics approach is unique in its integrated use of stocks and flows, causal feedback, and time delays to model structures and policies. Stocks represent accumulations or backlogs of work, people, information, or other portions of the system that change over time. Flows represent the movement of those commodities into, between, and out of stocks. For example, Figure 1 shows a simple stock and flow diagram of one possible arrangement of the backlogs and movements of work within a single activity (e.g., Advanced Development) of an acquisition program. Stocks are represented by boxes. Flows are represented by arrows between the boxes with the valve symbols. Arrowheads indicate the direction of movement of the work.

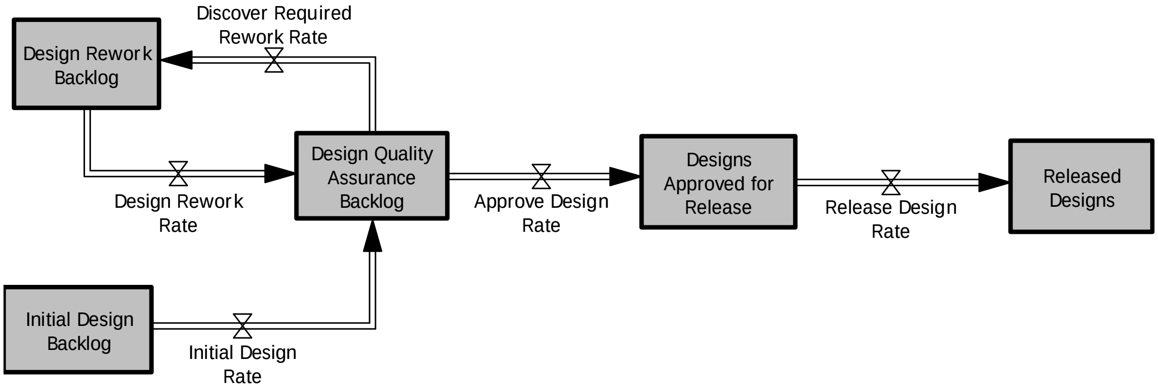
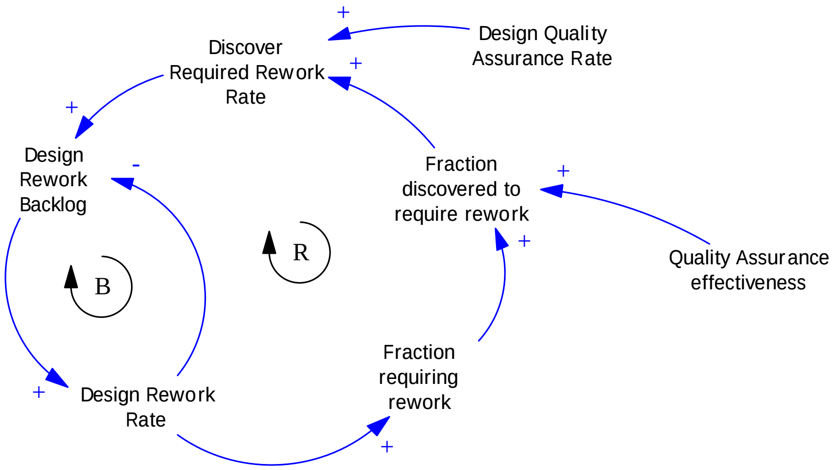


Figure 1. A Stock and Flow Diagram of Work Backlogs and Development Activities

Feedback is modeled conceptually in system dynamics with causal loop diagrams. Figure 2 shows a portion of a causal loop diagram for a single activity of an acquisition program. In causal loop diagrams, arrows indicate the direction of causal influence. The variable at the tail of an arrowhead influences the variable at the head of the arrow. A plus sign at an arrowhead indicates that the impacted variable and driving variable move in the same direction (i.e., an increase in the driving variable increases the impacted variable, and a decrease in the driving variable decreases the impacted variable). A negative sign at an arrowhead indicates that the impacted variable and driving variable move in opposite directions (i.e., an increase in the driving variable decreases the impacted variable, and a decrease in the driving variable increases the impacted variable). The two types of feedback loops are also illustrated in Figure 2. A balancing loop (“B” in Figure 2) tends to control or limit the movement of the variables in the loop. In contrast, a reinforcing loop (“R” in Figure 2) tends to move systems farther and farther from their initial conditions at faster and faster speeds. The behavior pattern generated by a specific feedback loop (e.g., exponential growth or movement toward a target) can be identified by sequentially tracing these impacts on variables through the series of causal links that describe the loop. See Sterman (2000) for a detailed description of the building and use of causal loop diagrams.



Discover Required Rework Rate
Design Quality Assurance Rate
Design Rework Backlog
Fraction discovered to require rework
Quality Assurance effectiveness
B
R
Design Rework Rate
Fraction requiring rework
+
+
+
-
+
+
+
+
+

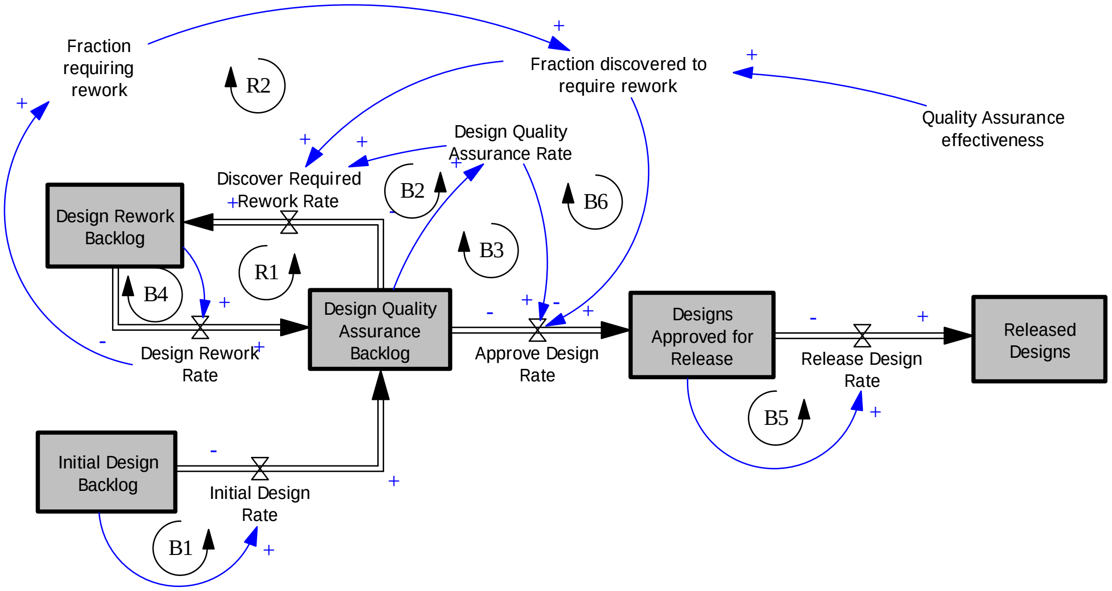
Feedback Loop Legend

B – Rework backlog increases rework rate, controlling the size of the backlog

R – Poor quality rework increases the work fraction requiring rework and rework backlog, further increasing the amount of work requiring rework

Figure 2. A Causal Loop Diagram of a Portion of an Advanced Development Phase

Stock and flow diagrams and causal loop diagrams can be integrated into system structure diagrams that simultaneously describe the feedback and accumulation/flow nature of the system being modeled. Figure 3 shows a system structure diagram of a model of an acquisition program phase. The diagram integrates the stock and flow diagram in Figure 1, the causal loop diagram in Figure 2, and some of the other important portions of the system. The feedback loop legend briefly describes how each feedback loop structure creates system behavior.


Fraction requiring rework
R2
Fraction discovered to require rework
Quality Assurance effectiveness
Design Quality Assurance Rate
Discover Required Rework Rate
B2
B6
Design Rework Backlog
R1
B3
B4
Design Quality Assurance Backlog
Designs Approved for Release
Released Designs
Design Rework Rate
Approve Design Rate
Release Design Rate
B5
Initial Design Backlog
Initial Design Rate
B1

<u>Feedback Loop Legend (partial)</u>

B1 – An increase in the Initial Design Backlog increases the initial design rate, thereby controlling the backlog

B2 – An increase in the Quality Assurance (QA) Backlog increases the QA rate and discovery of rework, thereby controlling the backlog

B1 – An increase in the Quality Assurance (QA) Backlog increases the QA rate and design approval rate, thereby controlling the backlog

B4 – An increase in the Rework Backlog increases the rework rate, thereby controlling the backlog

B5 – An increase in the accumulation of approved designs increases the size of the design release, thereby controlling the Approved Design accumulation

B6 – An increase in the Quality Assurance (QA) Backlog increases the QA rate, discovery of rework, fraction discovered, and approval rate, thereby controlling the backlog

R1 – An increase in the Quality Assurance (QA) Backlog increases the QA rate, discovery of rework, Rework Backlog, and rework rate, thereby increasing the QA Backlog further

R2 – An increase in the rework rate increases the fraction requiring rework, fraction discovered, discovery rate, and Rework Backlog, thereby increasing the rework rate further.

Figure 3. A System Structure Diagram of a Portion of Advanced Development

The full power of system dynamics can be realized only through formal simulation of the system's evolution. Formal simulation models developed from conceptual models are sets of nonlinear differential equations simulated with difference equations. Because no closed-form solutions are known, system behaviors over time are simulated. The simulator uses initial or current conditions, calibration values of constant parameters, and the difference equations to calculate conditions in the nest time period. Although the methodology initially assumes that small changes over time can be used to describe systems (e.g., the continuous adjustment of resources toward demands for those resources); however, discrete changes (e.g., the release of a complete design) at specific dates (e.g., a scheduled upgrade date) can also be modeled.

When applied to development projects, system dynamics focuses on how performance evolves in response to interactions among development strategy (e.g., Evolutionary Acquisition versus traditional acquisition), managerial decision-making (e.g., the allocation of resources), and development processes (e.g., concurrence). System dynamics is considered appropriate for modeling acquisition programs because of its ability to explicitly model critical aspects of development projects (Ford & Sterman, 1998; Cooper, 1993a, September; 1993b, September; 1993c; Cooper & Mullen, 1993; Cooper, 1994). System dynamics has been successfully applied to a variety of project management issues, including prediction/discovery of failures in project fast-track implementation (Ford & Sterman, 2003b, September), poor schedule performance (Abdel-Hamid, 1988; Taylor & Ford, 2006; 2008), the impacts of changes (Rodriguez & Williams, 1998; Cooper, 1980), the planning of fast-track construction projects (Pena-Mora & Li, 2001; Pena-Mora & Park, 2001), construction innovation (Park, Napa & Dulaimi, 2004), change management (Lee, Pena-Mora & Park, 2005; 2006; Park & Pena-Mora, 2003), resource allocation (Lee et al., 2007), and concealing rework requirements on project performance (Ford & Sterman, 2003a, September). See Lyneis and Ford (2007) for a review and analysis of the application of system dynamics to projects.

The ARCI Program

Information on the ARCI program was collected as the basis for modeling the OA and EA aspects of its acquisition process. In particular, differences between ARCI and traditional acquisition with an evolutionary approach were investigated. Data was collected primarily through a review of Navy documents (Johnson, 2007; Chief of Naval Operations, 2009), contractor program documents (Lockheed Martin, 2003; 2009), defense analyst documents (Global Security, n.d.), previous research concerning the program (e.g., Beaudreau, 2006; Johnson, 2004), and an extended interview with Bill Johnson, who developed and managed the ARCI program (Johnson, 2009). The data collection focused on the acquisition (development) aspects of ARCI. A summary of the results of that data collection follow.

Although it occurred within the established DoD acquisition processes of its time, the ARCI program was atypical in several important ways. The description here focuses on the program’s atypical nature, as it relates to the current work. See Beaudreau (2006) and Johnson (2004) for additional program descriptions. Three atypical aspects of the ARCI program in particular generated the need for and prompted the use of a new and different acquisition approach: 1) the urgent operational need, 2) tight constraints on funding, and 3) an environment of acquisition reform.

An Urgent Operational Need

In September of 1995, the Submarine Sonar Technology Panel reported a serious reduction in acoustic superiority. The reduced superiority resulted in reductions in the “stand off” distance between US submarines and other vessels (particularly other submarines), the distance at which US submarines recognize other vessels. The standoff distance is determined by the noise radiated from vessels and the capabilities of the recognizing ship through its sonar systems. Although the radiated noise of other vessels had progressively reduced (Figures 4 and 5), US sonar capabilities had not progressed in-step. Improved sonar systems could recapture the lost acoustic superiority. Importantly, the acoustic superiority loss had already occurred by 1995, and the need to regain it was considered urgent by the operating submarine fleet. *ARCI needed to develop solutions fast.* Figures 4 and 5 are examples of data used to support these findings and recommendations.

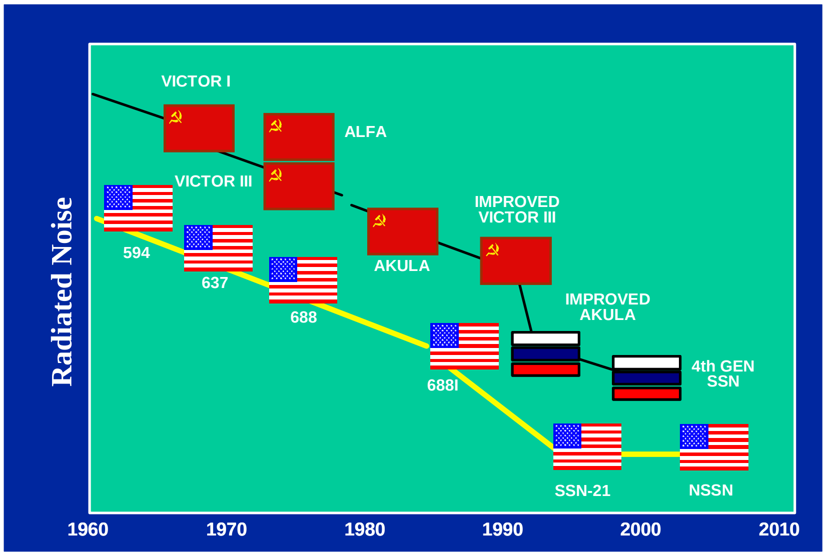


Figure 4. FSU (Former Soviet Union)/US Nuclear Stealth
(Johnson, 2007)

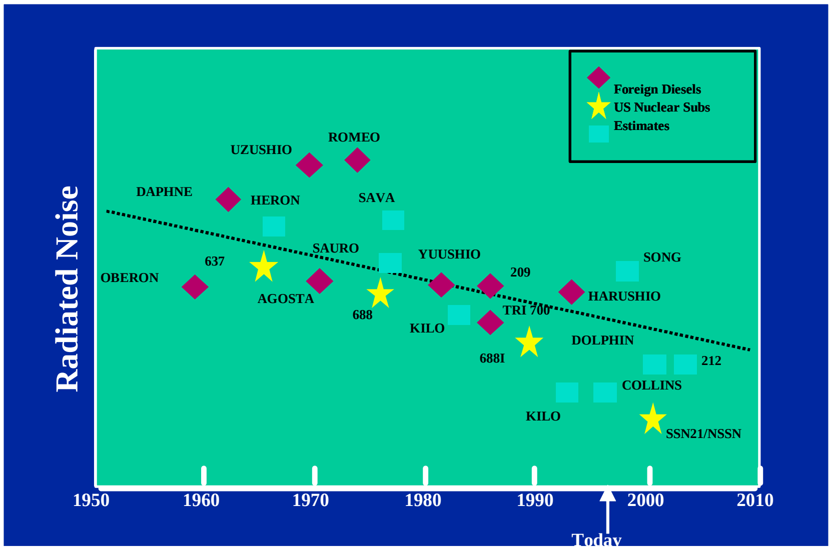


Figure 5. Diesel Rated Noise Trend
(Johnson, 2007)

Based on these findings, the Submarine Sonar Technology Panel recommended a radical transformation of the approach to designing and fielding sonar systems.

Tight Constraints on Funding

By 1995, the Cold War was over and funding for the DoD acquisition had reduced sharply, including Sonar Development and Combat Control Development funding (Figures 6 and 7).

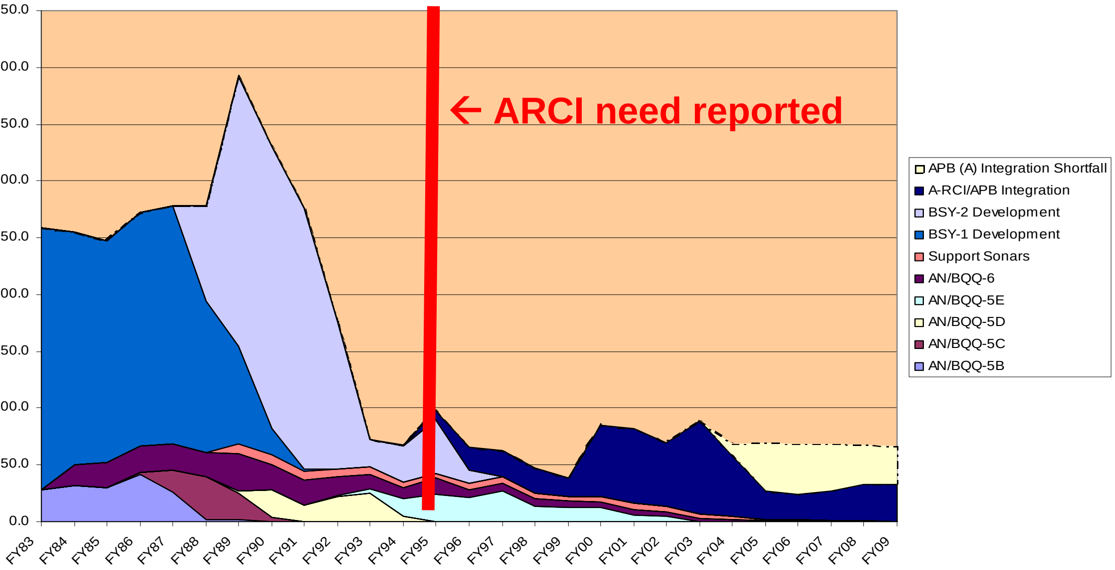


Figure 6. Sonar Development Funding
(Johnson, 2007)

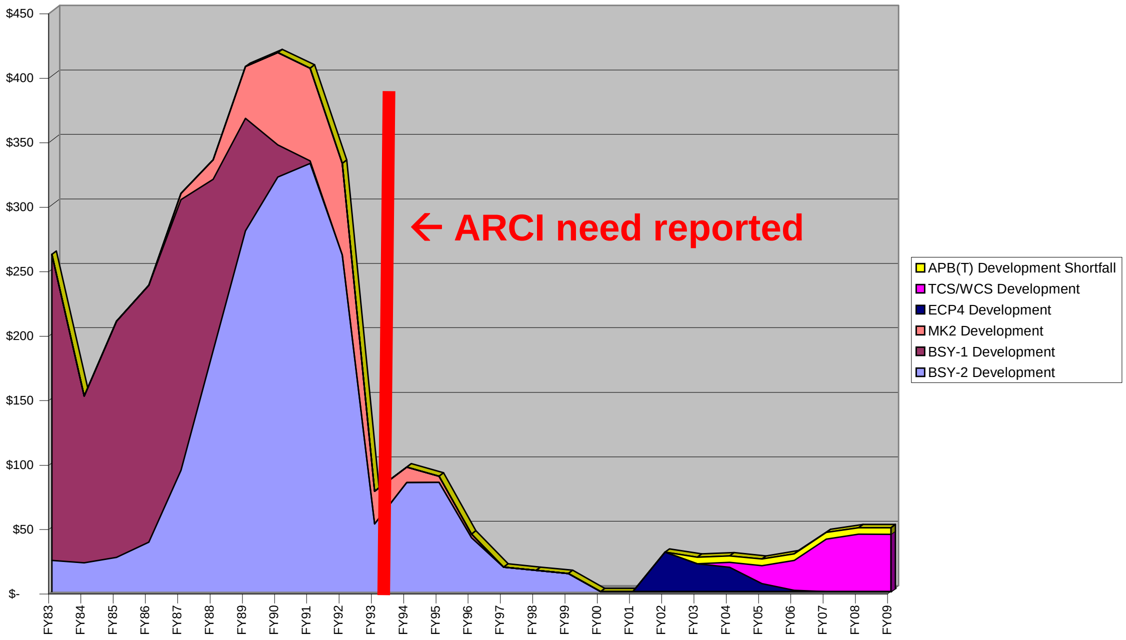


Figure 7. Combat Control Development Funding
(Johnson, 2007)

Traditional acquisition approaches, such as the development of unique systems for one or more sonar systems, were not available due to the large funding requirements of these approaches. ARCI had to develop solutions relatively inexpensively, at much less cost than required by the traditional DoD acquisition approaches.

An Environment of Acquisition Reform

Although not a characteristic of the ARCI program itself, the DoD acquisition processes were evolving faster than usual during the period in which ARCI began. This had potentially significant impacts on the program in terms of allowing it more than the usual amount of freedom to pursue and develop innovative acquisition perspectives, methods, and tools. These potential impacts are investigated later in the current work.

The ARCI Program Results

The ARCI program succeeded in significantly improving US submarine sonar systems quickly and at great savings. Figures 8 and 9 illustrate the demonstrated performance improvements.

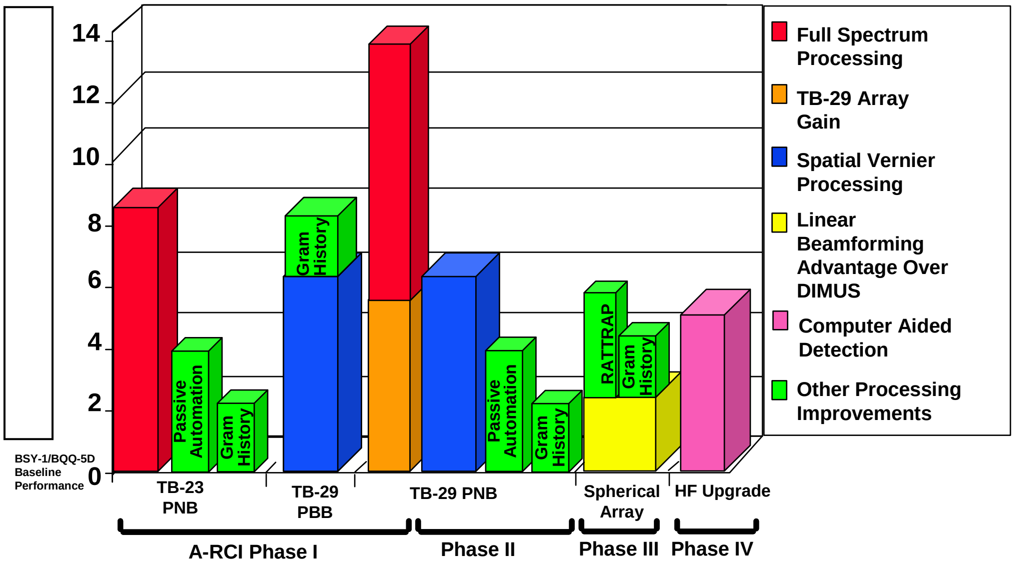
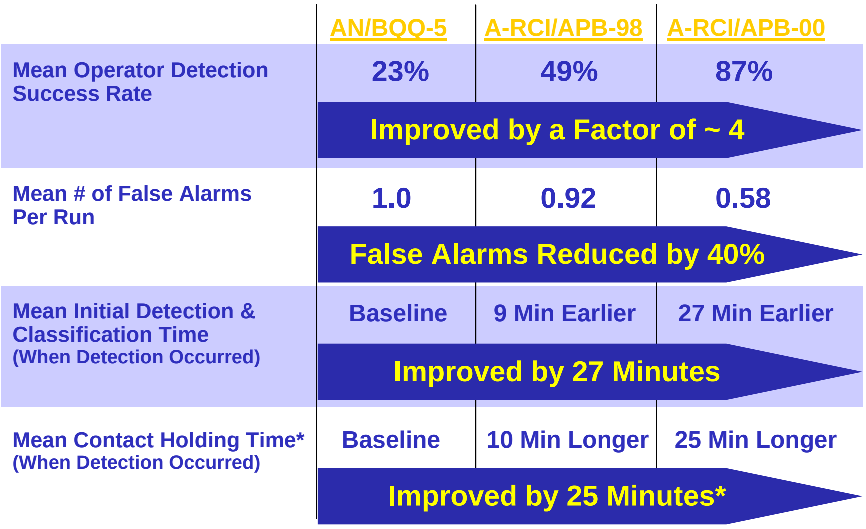


Figure 8. Demonstrated Performance Gains
(Johnson, 2007)



	AN/BQQ-5	A-RCI/APB-98	A-RCI/APB-00
Mean Operator Detection Success Rate	23%	49%	87%
	Improved by a Factor of ~ 4		
Mean # of False Alarms Per Run	1.0	0.92	0.58
	False Alarms Reduced by 40%		
Mean Initial Detection & Classification Time (When Detection Occurred)	Baseline	9 Min Earlier	27 Min Earlier
	Improved by 27 Minutes		
Mean Contact Holding Time* (When Detection Occurred)	Baseline	10 Min Longer	25 Min Longer
	Improved by 25 Minutes*		

* Measured holding time limited by the length of recorded tape.

Figure 9. Towed Array Processing Performance-improvement Trend
(Johnson, 2007)

ARCI performed quickly. Phase I improvements were installed on Agusta in December of 1997, and performance improvements delivered 18 months after the MDA decision. By the eighth anniversary of the ARCI MDA decision in June of 2004, ARCI had installed on over 50 submarines with at least four generations of hardware and software upgrades. These durations are much shorter than those in most comparable acquisition programs.

In addition to improving sonar system performance, ARCI generated large cost savings (Johnson, 2007) by reducing budget allocations across SCN, OPN, O&MN, RDT&E, and MilCon by over 50% ($7.6 billion to $3.6 billion) when the 1983-1993 budget allocations are compared to the 1996-2006 allocations. These savings reflect a reduction in Development and Production by a factor of six and a reduction in Operating and Support costs by a factor of eight. ARCI also realized over $25 million in cost avoidance for logistics support, including:

- Over $1 million in technical manuals,
- Over $2 million in direct vendor delivery,
- Over $19 million in interactive, multimedia instruction, and
- $3 million in outfitting spares reduction.

In summary, ARCI was an extremely successful acquisition program. A fundamental question for learning how to improve other acquisition programs is “Why was ARCI so successful?” Several factors, internal to the program and from its environment, help explain this success. Beaudreau (2006) focused on the role of the Modular Open Systems Approach (MOSA), now incorporated into the Navy’s Open Architecture approach, changing culture, and systems engineering (including spiral development, now termed Evolutionary Acquisition). The current work focuses on the dynamic nature of the ARCI program and what that nature suggests about the successful implementation of acquisition programs.

Open Architecture and Evolutionary Acquisition in the ARCI Program

ARCI was created in the early 1990s in an unusual acquisition environment that was dominated by an urgent need for significant improvement in active fleet capabilities, very constrained funding, and ongoing acquisition reforms. More specifically, submarine sonar hardware and software needed large improvements in performance. Complete solutions were not available and ready for operational testing when ARCI began. The need to develop solutions and make improvements quickly required an evolutionary approach. In addition, existing capabilities used legacy systems, which made repeated and fast changes difficult and expensive. Moving away from the legacy systems to an Open Architecture system potentially provided the flexibility needed for frequent upgrades as technologies developed. Program managers initially planned to replace legacy hardware with COTS (a central tenant of OA) to take advantage of the increased computing capability of hardware developed since the original system development and to facilitate future upgrades. Reduced hardware size provided space for the redesign of cabinets, etc. so that COTS products would meet military reliability requirements not met by those products "out of the box." ARCI managers originally planned to write middleware to link the new hardware and legacy software. However, analysis revealed that rewriting the operating software in a modern software platform (C++) was less expensive than developing middleware and also provided opportunities for an Open Architecture for software upgrades. Therefore, the Open Architecture approach was expanded to include software. Four acquisition iterations were initially designed (a central tenant of EA), each to address a different portion of the sonar system: 1) the towed array, 2) the hull array,[1] 3) the spherical array, and 4) the high frequency arrays. Each iteration used the standard DoD acquisition phases at the time of the program that identified and specified requirements, acquired technologies, designed and developed products, and integrated those solutions into ships.

As described so far, ARCI was a straightforward (albeit challenging) integration of Open Architecture and Evolutionary Acquisition. However, the ARCI program included some important features that distinguish it from known descriptions of the implementation of open systems and Evolutionary Acquisition. First, consider the dynamic nature of the need (evolving threats) and solutions (technology evolution). As hardware and software technologies improved and threats evolved, additional ARCI iterations would be needed. Improvements would be needed on an almost continuous basis to adequately improve fleet performance. Therefore, ARCI needed to be able to generate many repeatable capability upgrade iterations. This required ARCI to develop a process that integrated continuous processes with phased development, Open Architecture, and Evolutionary Acquisition. This was done partially by setting frequent upgrade release dates and not letting those dates slip. The first iteration was released 18 months after the identification of initial requirements, with subsequent upgrades every 12 months. This is much more frequent than the common DoD practice. The frequent integration of improvements was possible only by utilizing many previously developed technologies and solutions from a variety of sources (e.g., ONR, small businesses, academics). "Leverage, leverage, leverage" was a mantra in ARCI that referred to the program's emphasis on the use of existing technologies and solutions.

[1] Some towed array upgrades were included in some of the second (hull array) iterations to respond to the fleet's overwhelming support based on the results of initial towed array improvement results and the fleet's urgent need for improvement.

ARCI completed frequent upgrades through a second important difference between ARCI and other OA and EA programs that is related to the relationship of requirements, technologies, products, and implementation to specific acquisition iterations. Traditional DoD acquisition (including traditional EA) strongly link specific requirements to specific development blocks at the start of the block. Tests for specific blocks can be failed if the requirements linked to that block are not met and development is slowed to be sure that promised requirements are included.[2] Strongly linking requirements to blocks before solutions have been developed requires a flexible schedule—in case of development problems—lots of money to speed development, or both. ARCI had little flexibility of time or money, so it made requirements flexible to meet iteration deadlines (i.e., the commitments to upgrade at specific intervals) and control costs. This was done with a combination of a deviation from the traditional acquisition process and the use of a different conceptualization and utilization of several acquisition processes. ARCI delayed the selection of technologies and products to be included in each iteration until as late as possible (typically, about six months before delivery) and only included (at the program manager's discretion) those improvements for which developed technologies and solutions were available and in-hand. Requirements for which solutions were not yet available were delayed until solutions had been developed. ARCI is distinguished from many other DoD programs by its ability to locate the authority to include or delay meeting requirements with the program managers. According to the program manager, this was accepted by the fleet largely because the frequent iterations provided an opportunity for delays in meeting requirements to be relatively short, and solutions were being developed relatively rapidly.

ARCI managers also adopted a fundamentally different mental model of the acquisition process than was described in the DoD policy at the time of the program (e.g., *5000.1*) and extended concepts that are described in current policy (USD (AT&L), 2003b, May 12, sections 1 and 2, pp. 12-13). Current policy describes sequential acquisition phases (Materiel Solution Analysis, Technology Development, and Engineering and Manufacturing Development) that are repeated after requirements are developed, with continuous technology development and maturation (USD (AT&L), 2003b, May 12, Figure 2). In contrast, ARCI used continuous requirements development, technology development, and advanced development. Only the six-month implementation phases (analogous to Manufacturing Development) were viewed as specific to individual upgrades. This, and the Open Architecture approach to solutions, required ARCI to aggressively pursue and actively manage and coordinate continuous and parallel requirements revision, technology identification and development, and product development. This approach (three continuous processes and one iteration-based phase) is fundamentally different than traditional acquisition (all iteration-based phases) or current policy (one continuous process and several iteration-based phases). This approach also required a different set of government and contractor skills and relationships.

ARCI changed important relationships among program participants. The prime contractor was forced to take a role of primarily providing coordination but not generating solutions. This was to prevent solution bias in choosing technologies and products for inclusion in upgrades. Solutions were developed by multiple and diverse organizations (e.g., academia, ONR, small businesses) and chosen based on transparent assessments by an objective team

[2] This may be part of why traditional EA is difficult to plan. Program managers must successfully predict which requirements will be filled through future technology development, product design, and implementation when they commit to meet specific requirements for specific development blocks, often long before that technology and product development has occurred or can be reliably forecasted.

of experts. This successfully prevented purposeful or accidental sole-source acquisition by providing suppliers that were not awarded contracts with realistic opportunities to fairly compete and potentially win future ARCI work. These changes required several atypical program management skills.

Modeling the ARCI Program

The simulation model used here is based on a previously developed formal (i.e., computer simulation) system dynamics model of a DoD acquisition project using Evolutionary Acquisition and some aspects of open systems. The model is purposefully simple relative to actual practice to expose the relevant relationships, with a focus on the open systems and Evolutionary Acquisition aspects. Therefore, although many development processes and features of program participants interact to determine program performance, only those features that describe the critical evolutionary, Open Architecture, and ARCI-specific nature of the program are included. For example, the model assumes that resource productivities are fixed, that work backlogs are available for development, and that work packages are completed in accordance with schedule requirements (i.e., work packages on the critical path are completed first) but does not identify specific critical-path work packages. The literature cited above investigates the impacts of these and other factors influencing program performance. The model generates complex and realistic behavior patterns despite its relative simplicity when compared to the actual DoD acquisition programs. A brief description of the conceptual model that was used as the basis for the formal model provides a foundation for describing the current model of ARCI. See Ford and Dillard (2008) for a detailed description of the previous model.

A Conceptual Model of an Evolutionary Acquisition Program

The model structure reflects the structure of development work moving through the separate development blocks of an acquisition project. In the model, four types of work flow through each block of an acquisition project: requirements, technologies, product component designs, and manufactured products. Each type of work flows through a development phase that completes a critical aspect of the project: 1) develop requirements, 2) develop technologies, 3) design product components (advanced development), and 4) manufacture products. The exception is requirements, which also measures progress through the final phase, 5) conduct user product testing. Figure 10 shows development phases and information flows in a single block.

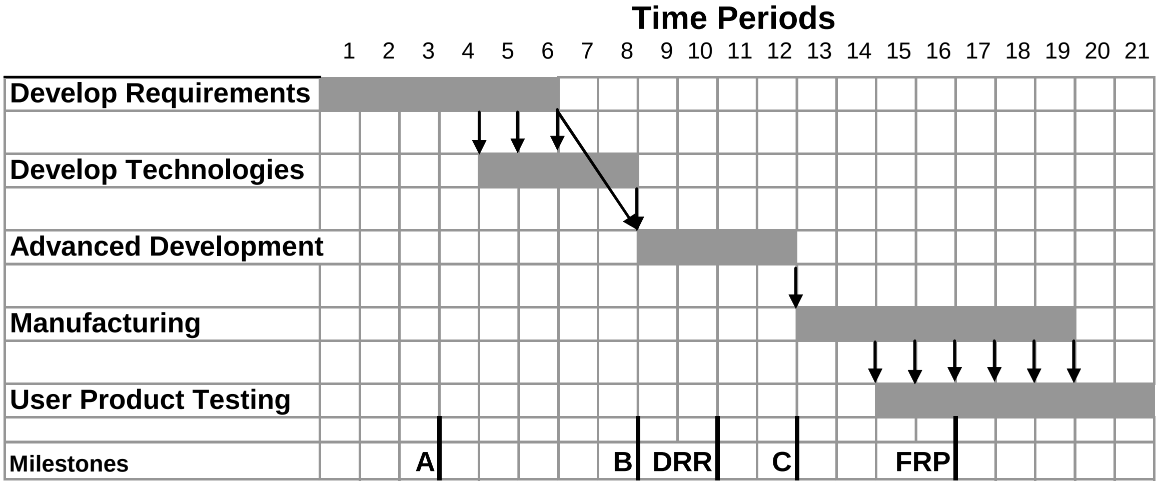


Figure 10. Information Flows in a Single-block Acquisition Project
(Ford & Dillard, 2008)

In Figure 10, arrows between phases indicate primary information flows. The start of all phases (except the development of requirements) is constrained by the completion of previous (“upstream”) phases. These constraints are relaxed in the ARCI model to reflect continuous development phases. In the previous model, the completion of some requirements allows for the start of technology development, reflecting the concurrent nature of this portion of acquisition. Both requirements development and technology development must be completed for advanced development to begin. The completion of advanced development allows manufacturing to begin. When some products have been manufactured, they are shipped to users for readiness testing. Figure 10 also identifies the five major reviews within a single acquisition block (A, B, Design Readiness Review, C, and Full-rate Production) at their approximate times during a project.

Each of the five phases in a development block (shown in Figure 10) are modeled with the workflows through the phase as a value chain of alternating backlogs and development activities with two types of rework cycle (within phases and between phases). The value chain is described with the boxes and pipes and with valves along the bottom of Figure 11. The value chain passes from the Initial Completion Backlog, through the Initial Completion Rate, into the Quality Assurance Backlog, through the Approval Rate, into the stock of Work Approved, and through the Release Rate to the accumulation of Work Finished and Released. Rework cycles are inherent in development projects and have been modeled and used extensively to explain and improve project management (Lyneis, Cooper, & Els, 2001; Ford & Sterman, 1998; Cooper & Mullen, 1993; Cooper, 1980; 1993a, February; 1993b, February; 1993c; 1994; Taylor & Ford, 2006; 2008). The scope of work is measured with the number of equal-sized work packages that must be completed in a development phase.

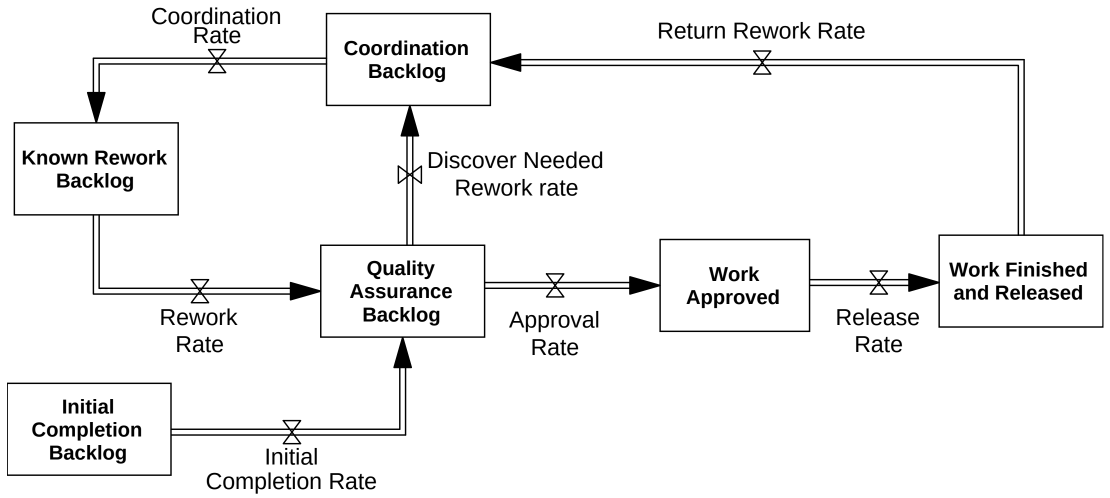


Figure 11. Work Backlogs and Flows through a Development Phase

For most phases in most blocks, all work starts in the backlog[3] of work needing to be initially completed ("Initial Completion Backlog" box at the bottom of Figure 11). The ARCI project includes an exception, which will be described later. As work is first completed, it enters the stock of work needing quality assurance (QA). Quality assurance could take many forms, including reviews of designs by senior engineers, prototype building and testing, and the inspection of work. Work needing quality assurance accumulates in a Quality Assurance Backlog (the box in the middle of Figure 11). If work passes QA (either because it is correct or the need for changes is not detected), it is approved and adds to the stock of Work Approved. When sufficient work has been approved, a package is released, adding to the stock of Work Finished and Released to other phases or users. The release package size is a management decision, often based on the characteristics of the phase. For example, in semiconductor development, the vast majority of the design code must be completed prior to release for a prototype build since almost all of the code is needed to design the masks. In other development settings, managers have broad discretion in setting release package sizes.

In rework cycles, between-phases work that is found to require changes moves into a stock of tasks that require changes that must be resolved through coordination with the phase responsible for the problem ("Coordination Backlog"). Classic examples include designers working with users to refine ambiguous or infeasible requirements or manufacturing engineers meeting with product designers to explain why parts can't be built as specified in the drawings. After coordination resolves the disputed issues, these tasks move to the stock of work known to need rework ("Known Rework Backlog") and are subsequently reworked and returned to quality assurance for re-inspection, testing, etc.

Since quality assurance is imperfect, some tasks requiring rework can be missed and erroneously approved and released. These rework requirements may be discovered later by another work phase. We assume that all defects are discovered in final product testing by users. When the phase that discovers the problem reports it, the generating phase is notified, and the affected tasks are moved from the stock of work considered finished to the coordination backlog

[3] Because the flows of development activities reflect the completion of the activity, the backlogs, as used here, include work in progress as well as work on which development has not yet been started.

and then eventually reworked. For example, a test phase may discover a short circuit across two layers in a prototype chip. If the error is traced to the design, test engineers must notify the designers and work with them to specify the location and characteristics of the short circuit. The designers must then rework, re-check and re-release the design, followed by changes in layout, tape-out, masking, and prototype fabrication.

The previous model of Open Architecture and Evolutionary Acquisition simulated the movement of work through an acquisition program. That model linked all five phases to specific iterations, which were completed at different intervals (Figure 10). Figure 12 depicts an acquisition project with multiple iterations or blocks. The first block is the same as Figure 10 above. Subsequent blocks have the same basic information flow, but can also be delayed by the completion of phases in previous blocks or constrained by the lack of progress in their own block.

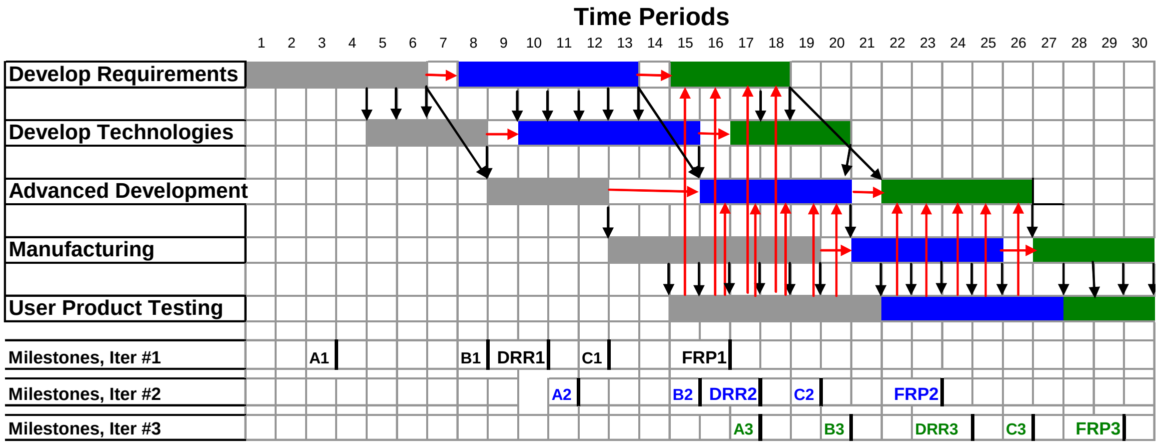


Figure 12. Information Flows in a Three-block Acquisition Project
(Ford & Dillard, 2008)

Modeling Open Systems in an Evolutionary Acquisition Program

The previous simulation model reflected some important aspects of open systems by changing model parameters to reflect impacts of open systems suggested by the literature. As an example, Table 1 describes some of the open systems impacts derived from Meyers and Oberndorf (2001) that were incorporated into the model.

Table 1. Impacts of Open Systems on Evolutionary Acquisition Due to Changes Suggested by Meyers and Oberndorf (2001)
(Ford & Dillard, 2008)

Change Required by Open Systems	**Impact on Evolutionary Acquisition Processes**
1) Build standards & COTS for program use	Increases Requirements scope in Block 1 Increases Technology Development scope in Block 1
2) Build high-level model with open systems	Increases Technology Development scope in Block 1
3) Document use of OS	Increases Technology Development scope in all blocks
4) Coordinate standards	Increases scope of all phases in all blocks
5) Implement OS	Decreases Advanced Development scope in all blocks Fewer Advanced Development design problems in all blocks
6) Integrate components	More Advanced Development integration problems in all blocks More Manufacturing integration problems in all blocks

Model Changes to Reflect the ARCI Program

The structure of the simulation model of a traditional acquisition program that adopts Open Architecture and Evolutionary Acquisition approaches was changed to better reflect the ARCI program. The primary changes are:

- Rename “Advanced Development” as “Design” to reflect the broader acquisition approach to this phase in ARCI that often adopted existing solutions instead of developing new solutions such as is often done in many traditional programs.

- Rename the “Manufacturing” phase to the “Integration” phase to reflect the nature of this activity in ARCI.

- Model the Requirements, Technology, and Design phases as a single, continuous development activity that occurs throughout the program.

- Begin the program with a set of initially developed requirements to be addressed but no inflow of new requirements. This reflects the conditions at the beginning of the program and the nature of the needs that the program was addressing (i.e., largely understood and described).

- Model the Integration activity as separate phases (as in the previous model), but start those phases at specific times (6 months before release), and end them at specific Integration release dates.

- Fix Integration release dates at 1.5 years after the program start (MDA) for the first release and then annually thereafter (i.e., at weeks 78, 130, 182, and 234).

- Disaggregate supplier-resource types into three types, reflecting those addressing technology acquisition, design, and implementation. The resources include several types of suppliers: contractors, ONR, government labs, and academic agencies.

Resources-for-requirements was not separately modeled because the requirements were largely already developed at the start of the program, and resources for checking and revising requirements was not considered by the program manager to constrain program progress.

- Disaggregate government program-management resources into three types, reflecting the same three types of resources as supplier modeling: technology, design, and integration work.

Little specific data was available for model parameter estimates. Therefore, the ARCI model was calibrated using data collected through the interview with the program manager and with modeler estimates. Figure 13 shows typical simulated behavior patterns of the ARCI program.

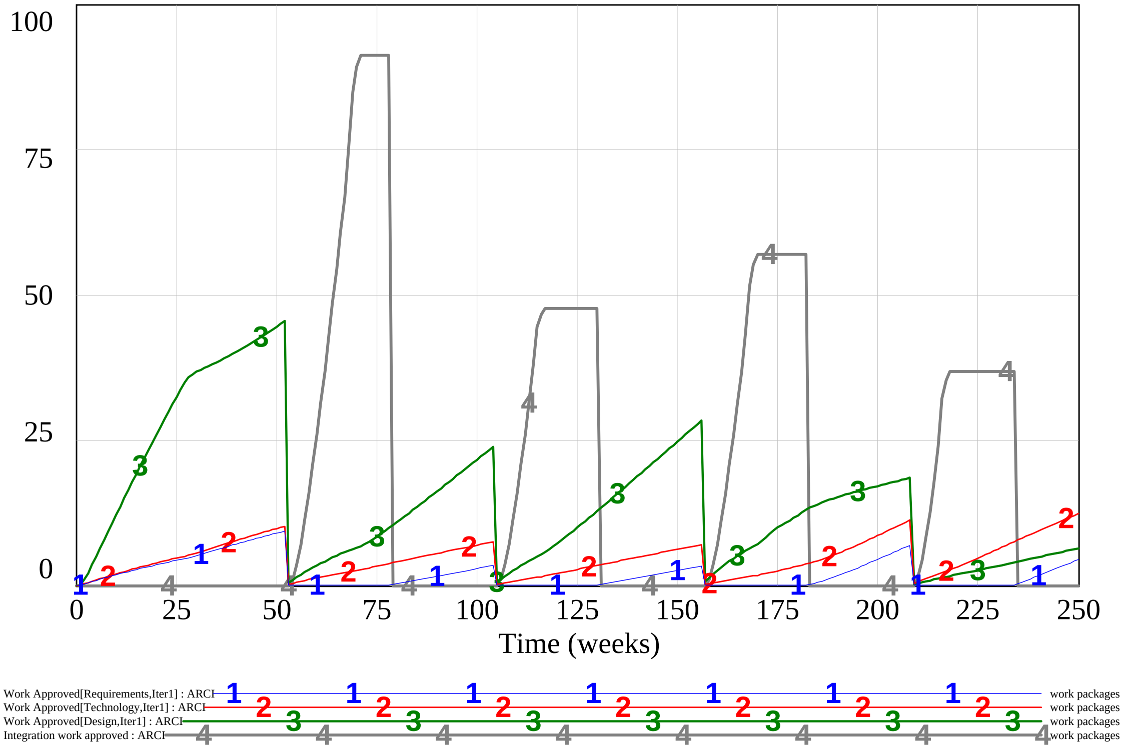


Figure 13. Approved Work in the Simulated ARCI Program

The vertical axis in Figure 13 is work packages, as described above. Figure 13 reflects the critical behavior patterns that describe the ARCI program. Work for each upgrade progresses first through the checking and revision (as required) of requirements (blue line #1 in Figure 13), subsequent acquisition of technologies to fulfill those requirements (red line #2 in Figure 13), and design of upgrade solutions using those technologies (green line #3 in Figure 13). As in the ARCI program, these continue throughout all upgrades (weeks 0–250 in Figure 13). But the accumulation of mature-enough requirements, technologies, and designs for each upgrade are collected at weeks 52, 104, and 156 (6 months before each release) to initiate the Integration phase for the upgrade. The four Integration phases (grey line #4 in Figure 13) each last six months and end at the release of each upgrade package to the fleet for operational testing and use. Consistent with ACRI, the revision of requirements, acquisition of technologies, and design of solutions does not stop during integration but continues, as show in Figure 13 by the overlapping of the progress rate lines during the Integration phases.

The ARCI model was tested for its usefulness for investigating implementation issues. Standard model tests as described by Sterman (2000) were used, including testing both the model structure and behavior. The model is based on previously developed system dynamics models of product development in several industries that have been developed and tested over several decades, as described and referenced above. Model structure was tested for similarity to the structure of the actual system through one-to-one linking of model components and specific parts of the system structure and units-consistency checks. Models were tested for their ability to generate “the right behavior patterns for the right reasons” (i.e., for the same reasons as in the actual system) using extreme-conditions testing and the comparison of simulated behavior patterns with an understanding of the behavior of the actual and similar systems. In extreme-conditions testing, one or more model parameter values are set to represent extreme conditions, which the modeler can use to predict correct model behavior. For example, the extreme condition of no resources should generate a program with no progress. The ARCI model generated reasonable behavior over a wide range of parameter values. The model behavior is similar to the described project behaviors, also supporting the model’s ability to reflect the relevant portions of the ARCI program. Based on these tests, the model was considered useful for investigating RCIP implementation issues.

Modeling RCIP

The Rapid Capability Insertion Process (RCIP) seeks to develop a process that can capture the types of performance improvements realized by ARCI in more and larger acquisition programs. The upgrading of AEGIS and its preparation for net-centric warfare is a potential application of RCIP. RCIP is based on the ARCI program and includes its core concepts and changes from most traditional acquisition projects, including those that adopt the Open Architecture and Evolutionary Acquisition approaches. However, based on an interview with one of the RCIP developers (2009), there will be differences between ARCI and RCIP, primarily:

- RCIP will be applied to larger acquisition efforts (e.g., AEGIS);
- After an initial start-up phase, RCIP will receive and develop a continuous stream of new requirements instead of having a fixed set of established requirements in place, as ACRI had;
- RCIP is initially planned to release upgrades to the fleet every two years, thereby adopting a cycle that is twice as long as that used in ARCI; and
- RCIP is planned to use 12-month integration periods, twice as long as those in ARCI.

These differences were integrated into the simulation model to provide an estimate of the potential of the RCIP approach. This represents a simple (and simplistic, as will be explained) scaling of the ACRI approach to RCIP. Figure 14 shows RCIP’s potential performance. Steady-state output exceeds the ARCI’s average output, although ARCI’s transitional (versus steady-state) nature and model calibration preclude useful direct comparisons.

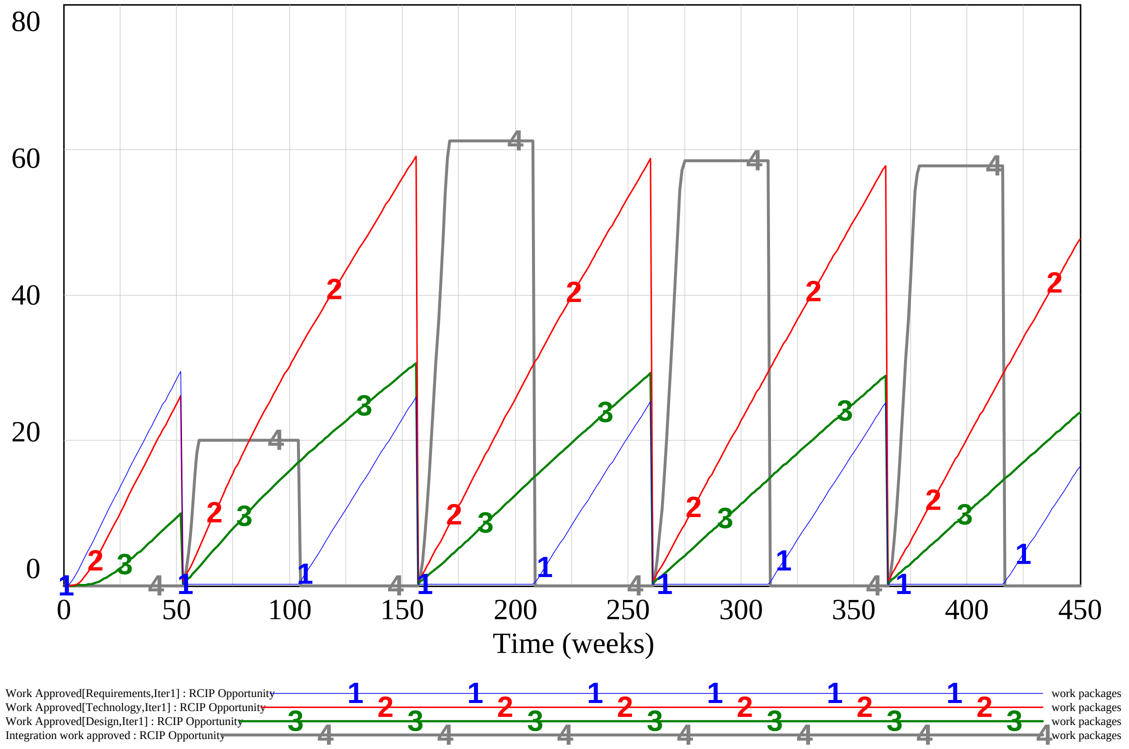


Figure 14. Simulated RCIP Program Behavior

Figure 14 includes the fundamental, desired behavior of RCIP, a basically continuous process of requirements development upgrades (after an initial start-up phase). While useful as a benchmark for the current work, important risks must be addressed to better reflect the RCIP approach.

The RCIP Implementation Risks

A simple scaling-up of the ARCI program into an RCIP program will not capture the potential performance (especially considering that the model above is simplistic) because it ignores important implementation risks that can degrade RCIP performance when compared to its potential. In addition to the changes from ARCI to RCIP listed above, several implementation challenges pose risks that may affect RCIP, including: 1), an increased pool of suppliers due to increased scale, 2) a reduced number of off-the-shelf technologies and designs available for use, resulting in a need for more new development, and 3) increased systems that solutions must be integrated across. Table 2 contrasts the three acquisition programs to highlight their differences and identify some RCIP implementation risks.

Table 2. Contrasts among Traditional Phased, ARCI, and RCIP Acquisition Programs

Acquisition Feature	**Phased Program with OA & EA**	**ARCI Program**	**RCIP Programs** (vs. ARCI)
Development processes (Req., Tech, AdvDev)	Repeated separate phases	Primarily continuous processes, known requirements	*Continuous processes with continuous inflow of requirements*
Innovation sources	Primarily through Prime Contractor	Primarily Off-the-shelf solutions	*Mix of new development & off-the-shelf. More new development*
Product System Modularity	Often integrated across phases & development blocks	Primarily separate systems (towed, hull, spherical, high frequency)	*More systems and system interactions. More inter-system integration required*
Govt./Supplier Relationships	Prime contractor	“Prime” coordinator & multiple solution suppliers	*Larger solution supplier pool*
Primary Locus of Performance Flexibility	Cost, Schedule	Scope	*Scope with possible flexibility in cost*

The Primary Locus of Program Flexibility (the last row in Table 2) describes a generic model of program management that can partially explain the ARCI success and facilitate the design and management of RCIP programs. The model describes how program management handles, in practice, the ubiquitous circumstances of having inadequate resources (broadly defined) to meet all performance targets (e.g., cost <= budget, completion date <= deadline, capabilities >= warfighter needs). In these circumstances, program management is forced to select one or more performance dimensions that will not meet targets and project by how much they will underperform. The dimension or dimensions that are chosen is the Primary Locus of Program Performance Flexibility. A common saying among commercial contractors (although perhaps not said to their clients) that captures the essence of this model is “Fast, cheap, good. Pick two.” Table 2 identifies the Primary Locus of Performance Flexibility as a significant difference between traditional programs with Open Architecture and Evolutionary Acquisition and programs adopting the ARCI/RCIP approach. In the former, performance flexibility is primarily located in the cost and schedule dimensions. In contrast, in the ARCI program, it was in the scope included in the current upgrade. In the RCIP approach, it is expected to remain in the scope dimension, with the possibility that cost may also provide some flexibility.

The RCIP’s expected implementation risks were integrated into the simulation model. Specifically:

- Increased scope is expected to attract increased oversight and, therefore, reduce productivity due to the use of resources (primarily labor) in the preparations for reviews, etc. (20% reduction estimated).

- Existing inventories of requirements, off-the-shelf technologies, and off-the-shelf designs were reduced by 50% to reflect the need for more new development. This will require their initial development in addition to the testing and revisions included in the ARCI model.

- Increased new development will also require more integration effort than off-the-shelf solutions, which have already been partially developed for integration upon adoption. Therefore, the amount of integration work was increased by 25%.
- Increased new development will also make integration more difficult than off-the-shelf solutions, which have been partially tested for integration upon adoption. Therefore, the amount of iteration required in the integration phases was increased by 25%.

Figure 15 shows the simulated RCIP program with implementation risks. The program retains its fundamental behavior pattern of a primarily continuous upgrade process.

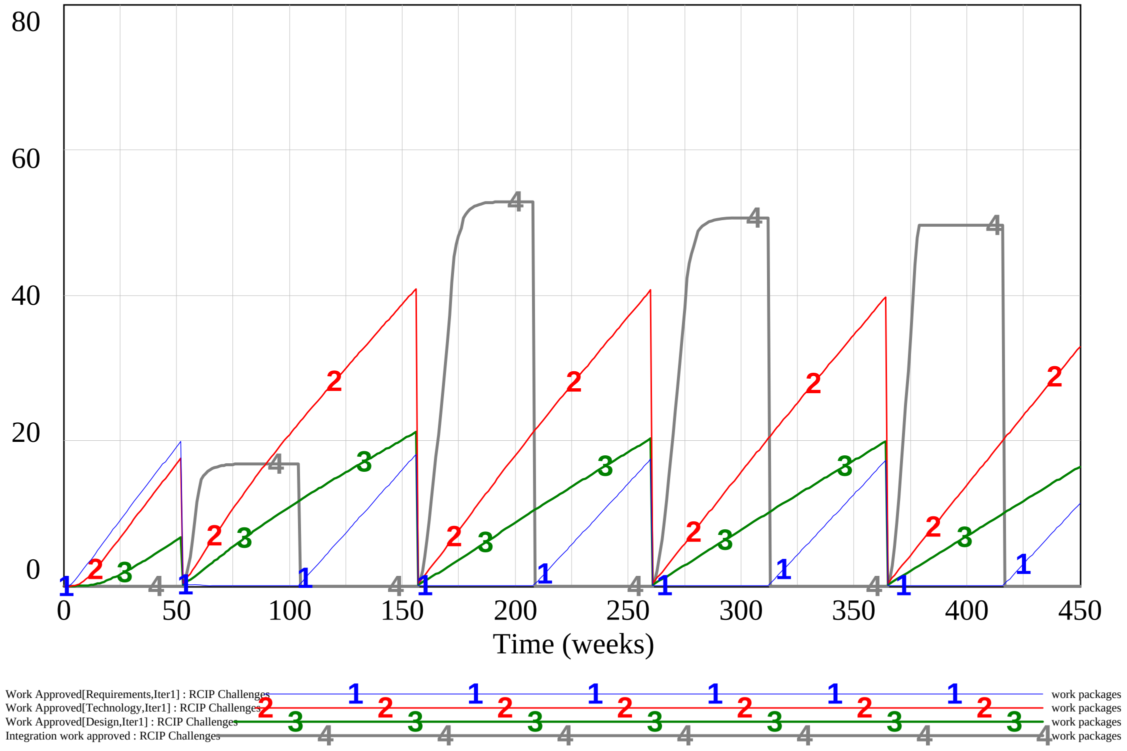


Figure 15. Simulated RCIP with Implementation Challenges

Implications of Implementation Risks for RCIP Success

The work completed and released to the fleet when RCIP implementation risks are considered (Figure 15) is significantly less than the potential (Figure 14). Figure 16 illustrates this difference (about 14% in the simulated program) by accumulating the Integration phase work released across four upgrades, without (blue line #1) and with (red line #2) implementation risks included. RCIP implementation risks must be addressed to capture the full potential of the ARCI approach in RCIP.

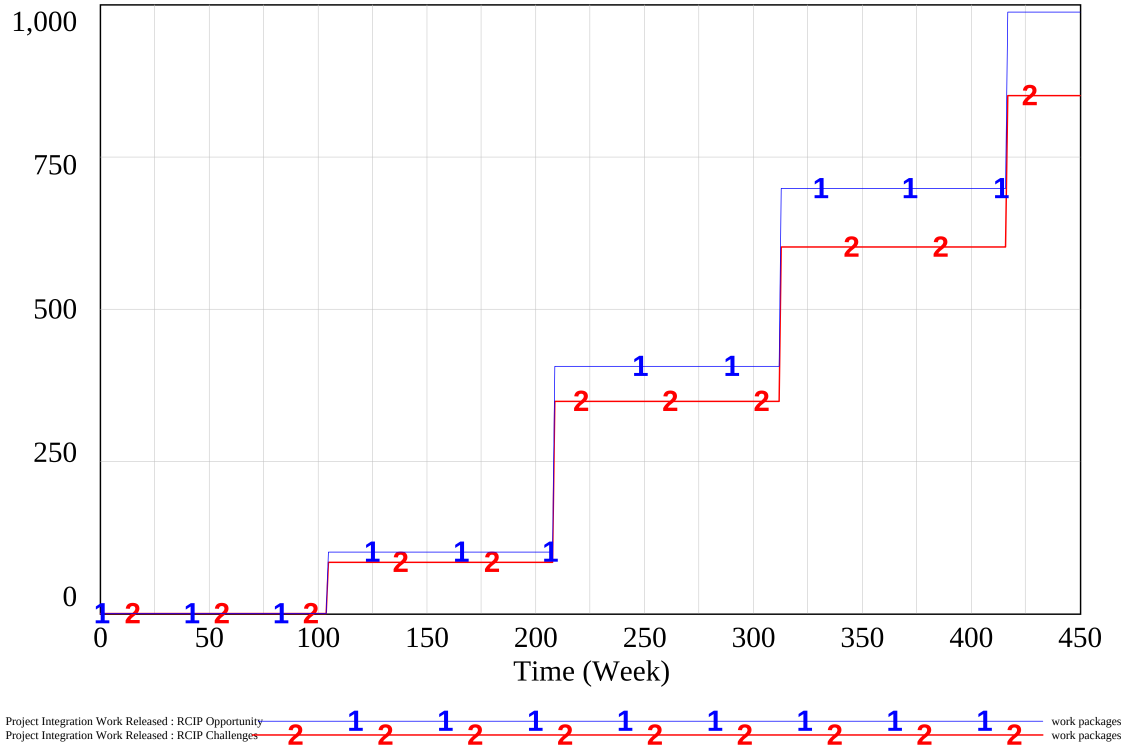


Figure 16. Cumulative RCIP Performance without and with Implementation Issues

Another RCIP implementation risk is program management burnout. The ACRI program manager specifically identified the potential of burnout in his program management team due to the repeated, intense Integration phases. To investigate the possibility and severity of this risk to RCIP implementation, the total required (but not necessarily provided) government program-management workforce size was simulated for the ARCI program, RCIP without implementation risks, and RCIP with implementation risks (Figure 17). Figure 17 clearly shows the spikes in demand for program management during the Integration phases for all three simulations. Notice that the peaks are significantly higher for both RCIP simulations than for the ACRI simulation. This suggests that the burnout risk will be larger for RCIP than it was for ACRI. *Successfully implementing a sustainable RCIP program will require a method to address potential burnout of the government program-management workforce.*

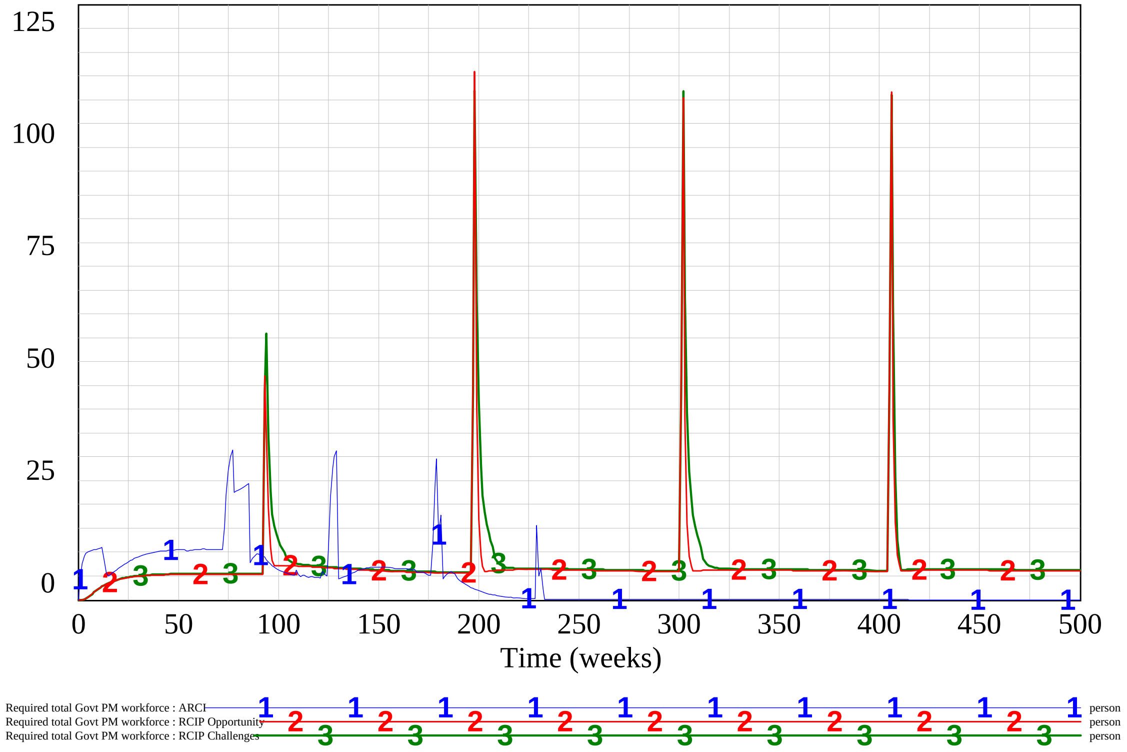


Figure 17. Simulated Total Required Government Program-management Workforce: ARCI, RCIP without and with Implementation Risks

Managing RCIP Implementation Risks

The RCIP's implementation risks can be managed through the careful design of its processes, organizations, and their interactions. Specific recommendations based on the ARCI program and the modeling and analysis discussed above are described in Table 3.

Table 3. Managing RCIP Implementation Risks

Acquisition Feature	**ARCI Program**	**RCIP Programs** (vs. ARCI)	**RCIP Implementation Risk Management**
Development processes	Primarily continuous processes, known requirements	Continuous processes with continuous inflow of requirements	*Standardize continuous processes and add rigor for sustainability*
Innovation sources	Primarily Off-the-shelf solutions	Mix of new development & off-the-shelf. More new development	1) *Adapt continuous processes into a mixture of off-the-shelf & new development solutions* 2) *Implement EA "only-mature-enough" strategy*
Product System Modularity	Primarily separate systems (towed, hull, spherical, high frequency)	More systems and system interactions. More inter-system integration required	*Operationalize modular configuration management for large-scale acquisition with focus on integration*
Government / Supplier Relationships	"Prime" coordinator & multiple-solution suppliers	Larger-solution supplier pool	*Formalize open, transparent, objective, & repetitive competition processes and organizations*
Primary Locus of Performance Flexibility	Scope	Scope with possible flexibility in cost	1) *Improve user-acquisition coordination to facilitate scope flexibility* 2) *Operationalize ARCI management of solution acquisition to make RCIP responsive to warfighter priorities*

Conclusions

The Acoustic Rapid COTS Insertion (ARCI) program was studied as the basis for modeling the planned Rapid Capability Insertion Process (RCIP) approach for continuous, reduced-cost upgrading of warfighting assets. ARCI used atypical methods in the face of atypical program requirements and conditions. ACRI was very successful in improving performance quickly for reduced costs. A previously developed acquisition program model was adapted to reflect ARCI and used for model validation. This model was then changed to reflect the basic conditions expected in RCIP programs. The model demonstrated the potential of RCIP to improve program performance. However, implementation risks were identified that may degrade potential performance, including increased oversight, the use of more new development, and the resulting integration scope and risk. When incorporated into the model, these risks were shown to significantly decrease RCIP performance. The means for successfully managing the RCIP design based on the ACRI program and RCIP operations are suggested for use in addressing the identified implementation risks

Based on the work described above, we conclude that RCIP has great potential to improve acquisition. But the failure to identify and successfully address implementation risks, in particular, can significantly constrain RCIP program performance. Special attention must be paid in the design of the RCIP approach to the differences between the ACRI program and the features, characteristics, and environmental conditions that RCIP programs will face. The five principles, concepts and tools and methods embodied in the Navy's Open Architecture approach to acquisition are likely to be particularly useful in developing RCIP and addressing its implementation risks. Many Open Architecture concepts were used successfully in the ACRI program, including modular design, design disclosure, interoperability, lifecycle affordability, and lots of vigorous (and vigorously managed) competition to generate a wide range of possible solutions from many sources. Applying Open Architecture required strong, assertive, government program management but provided the basis for extraordinary success. Similar extraordinary success is possible in RCIP programs but will also require a process based on Open Architecture and vigorous and assertive management by the government. By doing so, RCIP can become an example of effective and efficient acquisition for widespread adoption to many acquisition efforts.

Acknowledgements

The authors thank the NPS Acquisition Research Program, especially RADM James Greene, for financial support and leadership; the Integrated Weapons Systems group, especially Nick Gertner and Mark Wessman, for financial support and a critical look into the ARCI program and Open Architecture; and Bill Johnson, ARCI program manager, for critical information.

List of References

Abdel-Hamid, T. (1988). Understanding the "90% syndrome" in software project management: A simulation-based case study. *The Journal of Systems and Software*, *8*, 319-330.

Beaudreau, M. (2006). *Acoustic rapic COTS insertion: A case study in spiral development* (NPS-PM-06-041). Monterey, CA: Acquisition Research Program, Naval Postgraduate School.

Chief of Naval Operations, Submarine Warfare Division. (2009). *Future submarine force, acoustic rapid COTS insertion (ARCI)*. Retrieved April 12, 2009, from http://www.navy.mil/navydata/cno/n87/future/arci.html

Cooper, K.G. (1980). Naval ship production: A claim settled and a framework built. *Interfaces*, *10*(6), 20-36.

Cooper, K.G. (1993a, February). The rework cycle: Why projects are mismanaged. *PM Network*, 5-7.

Cooper, K.G. (1993b, February). The rework cycle: How it really works…and reworks. *PM Network*, 25-28.

Cooper, K.G. (1993c). The rework cycle: Benchmarks for the project managers. *Project Management Journal*, *14*(1), 17-21.

Cooper, K.G. (1994). The $2,000 hour: How managers influence project performance through the rework cycle. *Project Management Journal*, *15*(1), 11-24.

Cooper, K.G., & Mullen, T. (1993). Swords and plowshares: The rework cycles of defense and commercial software development projects. *American Programmer*, *6*(5), 41-51.

DAU. (2009). Naval open architecture. Retrieved April 1, 2009, from https://acc.dau.mil/oa; https://acc.dau.mil

Dillard, J., & Ford, D.N. (2007, May). Too little too late? Modeling the risks of spiral development. In *Proceedings of the Fourth Annual Acquisition Research Symposium*. Monterey, CA: Naval Postgraduate School.

Flood, R.L., & Jackson, M.C. (1991). Creative problem solving: Total systems intervention. Chichester, England: Wiley.

Ford, D.N., & Dillard, J.T. (2008, May). Modeling the integration of open systems and evolutionary acquisition in DoD programs. In *Proceedings of the Fifth Annual Acquisition Research Symposium*. Monterey, CA: Naval Postgraduate School. Retrieved from http://acquisitionresearch.net/index

Ford, D., & Sterman, J. (1998, Spring). Modeling dynamic development processes. *System Dynamics Review*, *14*(1), 31-68.

Ford, D., & Sterman, J. (2003a, September). The liar's club: Impacts of concealment in concurrent development projects. *Concurrent Engineering Research and Applications*, *111*(3), 211-219.

Ford, D., & Sterman, J. (2003b, September). Overcoming the 90% syndrome: Iteration management in concurrent development projects. *Concurrent Engineering Research and Applications*, *111*(3), 177-186.

Forrester, J.W. (1961). *Industrial dynamics*. Cambridge, MA: MIT Press.

Global Security. (n.d.). BQQ-10 A-RCI Acoustic Rapid COTS Insertion. Global Security.org. Retrieved April 12, 2009, from http://www.globalsecurity.org/military/systems/ship/systems/an-bqq-10.htm

Jackson, M.C. (2003). *Systems thinking: Creative holism for managers*. Chichester, England: Wiley.

Johnson, W. (2004). The A-RCI process—Leadership and management principles. *Naval Engineers Journal*, *116*(4), 99-106.

Johnson, W. (2007, July). ARCI—A historical perspective. Briefing. Presented to the Submarine Sonar Technology Panel.

Johnson, W. (2009, January 6-7). [Communication with researchers]. Washington, DC.

Lane, D.C., Größler, A., & Milling, P.M. (Eds.). (2004). Rationality in system dynamics: Selected papers from the first European system dynamics workshop, Mannheim University. *International Journal Systems Research and Behavioral Science*, *21*(4, Special Edition)..

Lane, D.C., & Jackson, M.C. (1995). Only connect! An annotated bibliography reflecting the breadth and diversity of systems thinking. *Systems Research*, *12*, 217-228.

Lee, S., Pena-Mora, F., & Park, M. (2005). Quality and change management model for large scale concurrent design and construction projects. *ASCE Journal of Construction Engineering and Management*, *131*(8), 890-902.

Lee, S., Pena-Mora, F., & Park, M. (2006). Reliability and stability buffering approach: Focusing on the issues of errors and changes in concurrent design and construction projects. *ASCE Journal of Construction Engineering and Management*, *131*(5), 452-464.

Lockheed Martin. (2009). *Acoustic rapid COTS insertion (ARCI)*. Retrieved April 12, 2009, from http://www.lockheedmartin.com/products/AcousticRapidCOTSInsertion/index.html

Lockheed Martin Maritime Systems and Sensors. (2003). *Acoustic rapid COTS insertion (ARCI)*. Retrieved April 12, 2009, from http://www.lockheedmartin.com/data/assets/881.pdf

Lyneis, J.M., & Ford, D.N. (2007). System dynamics applied to project management: A survey, assessment, and directions for future research. *System Dynamics Review, 23*(4), 157-189.

Lyneis, J.M., Cooper, K.G., & Els, S.A. (2001). Strategic management of complex projects: A case study using system dynamics. *System Dynamics Review*, *17*(3), 237-260.

Meyers, B.C., & Oberndorf, P. (2001). *Managing software acquisition: Open systems and COTS products.* Boston, MA: Addison-Wesley.

Park, M., Nepal, M., & Dulaimi, M. (2004). Dynamic modeling for construction innovation. *ASCE Journal of Management in Engineering*, *20*(4), 170-177.

Park, M., & Pena-Mora, F. (2003). Dynamic change management for construction: Introducing the change cycle into model-based project management. *System Dynamics Review*, *19*(3), 213-242.

Pena-Mora, F., & Li, M. (2001). Dynamic planning and control methodology for design/build fast-track construction projects. *ASCE Journal of Construction Engineering and Management*, *127*(1), 1-17.

Pena-Mora, F., & Park, M. (2001). Dynamic planning for fast-tracking building construction projects. *ASCE Journal of Construction Engineering and Management*, *127*(6), 445-456.

Rodriguez, A., & Williams, T.M. (1998). System dynamics in project management: Assessing the impacts of client behavior on project performance. *Journal of the Operational Research Society*, *49*, 2-15.

Stapleton, J. (2006, September 26). *APB/A-RCI performance driven acquisition reform for undersea warfare superiority*. Briefing. Slide 2.

Sterman, J.D. (2000). *Business dynamics, system thinking and modeling for a complex world.* New York: Irwin McGraw-Hill.

Taylor, T.R., & Ford, D.N. (2008, June). Managing tipping point dynamics in development projects. *ASCE Journal of Construction Engineering and Management*, *134*(6), 421-431.

Taylor, T., & Ford, D. (2006, Spring), Tipping point dynamics in development projects. *System Dynamics Review*, *22*(1), 51-71.

USD (AT&L). (2003a, May 12). The defense acquisition system (DoD Directive 5000.1). Washington, DC: Author.

USD (AT&L). (2003b, May 12). Operation of the defense acquisition system (DoD Directive 5000.2). Washington, DC: Author.

THIS PAGE INTENTIONALLY LEFT BLANK

2003 - 2009 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities-based Planning
- Capital Budgeting for DoD

- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-tem Attrition
- Retention
- The Navy’s Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness
- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)
- Risk Analysis for Performance-based Logistics
- R-TOC Aegis Microwave Power Tubes

- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Earned Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

THIS PAGE INTENTIONALLY LEFT BLANK

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org