Modeling the Integration of Open Systems and Evolutionary Acquisition in DoD Programs

Presenter: David N. Ford received BS and MS degrees from Tulane University and his PhD degree from the Massachusetts Institute of Technology, Cambridge. He is a Professor in the Construction Engineering and Management Program, Zachry Department of Civil Engineering, Texas A&M University and a Research Associate Professor of Acquisition with the Graduate School of Business and Public Policy at the US Naval Postgraduate School in Monterey, CA. 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.

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 Army Tactical Missile System (ATACMS) and Javelin 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
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

Open systems and evolutionary acquisition are two recent innovations designed to improve program performance with flexibility. The full potential of these approaches has not been captured, partially because of integration challenges during implementation. The current work investigates the impacts of open systems and evolutionary acquisition on DoD development programs. Development changes required to simultaneously use open systems and evolutionary acquisition are used to identify and describe impacts of implementation on program process and management. A dynamic simulation model of a program using both evolutionary acquisition and open systems is described and used to map these impacts. Simulation results generally support the suggested impacts identified in the literature and provide a possible explanation for changes in program performance. Implications for practice

relate to changes in the types and timing of risk and a potential trading of design obsolescence risk for standards obsolescence risk.

Introduction

In order for the military to prepare to meet current and future capability demands, major DoD systems must improve with evolving technologies and be interoperable. The continued and, in some cases, accelerating evolution of technologies creates new challenges that are difficult to forecast and require rapid acquisition response. Integrated human-computer decision-making tools, advanced materials, NCS tools, and nano-level structures are examples of evolving technologies that present challenges and potential solutions that must be integrated by defense acquisition programs. The use of legacy and other weapon platforms, joint service solutions, the information and communication needs of Network Centric Systems (NCS), and coordination with allies in joint operations each require weapon systems that can operate across system, platform, and systems-of-systems boundaries. Providing those systems is an important acquisition challenge. Past DoD acquisition approaches have not fully provided the interoperability needed to meet these demands.

Open systems (OSJTF, 2004, September) and evolutionary acquisition (DoD, 2004, November, section 4.4.1) are two relatively recent DoD acquisition initiatives that seek to address system interoperability and technology evolution challenges and that help the DoD meet current and future capability needs. An open systems (OS) approach and evolutionary acquisition (EA) share several high-level objectives. Both approaches seek to improve performance over the system's lifetime and reduce acquisition cycle-time. Both approaches also attempt to improve system performance via flexibility for the integration of new technologies and information into systems as they evolve. The open systems approach facilitates upgrades through modularity. EA does this by multiple product releases and deliberate deferral of some functionality—allowing technologies and requirements to evolve and mature. Both OS and EA seek to reduce acquisition cycle-time to provide currently available functionality. OS provide a means of incorporating current and future functionality, and evolutionary acquisition limits the scope of development blocks to only the technologies and capabilities that are attainable in the near future.

Open systems and evolutionary acquisition share at least two important implementation approaches. First, both OS and EA incorporate flexibility into acquisition to manage uncertainty in technology. Open systems build flexibility into development products with modular design and standardized key interfaces. Evolutionary acquisition builds flexibility into development processes through the design of incremental capability blocks. These flexibilities create options that potentially increase system performance, reduce cost, or both, by allowing technological uncertainties to partially resolve before important development decisions are made. Second, both OS and EA place emphasis upon interfaces to address interoperability. Within an evolutionary approach, interface management is critical to successfully integrating designs across development blocks. This need increases for systems with interfaces across platforms or systems-of-systems. In contrast to these challenges, an OS approach focuses on explicitly identifying and managing key interfaces that can benefit from modular design and open systems as a means of improving interoperability.

The evolutionary acquisition challenge and the open systems method suggest that the two acquisition approaches must be integrated and may be synergistic. But the complexity of the processes and the requirements of the two approaches make their integration, synergy, and successful implementation anything but easy or certain. The requirements of the approaches have been largely identified, and some of the required changes in programs for the use of EA and OS together have been identified. But a focused study of the impacts of integrating open systems and evolutionary acquisition is needed both to identify the impacts on development processes and to point to potential program design and management actions in order to exploit their potential. It is not obvious how to investigate and better understand the integration and implementation issues presented by evolutionary acquisition and open systems as concurrent approaches. ***How does the use of evolutionary acquisition and open systems together impact a system's development processes and management? How do those impacts affect acquisition program performance?***

The current work partially addresses these issues as follows. The researchers review the evolutionary acquisition and open systems approaches through the lens of their influence on program processes and management. Required changes in programs identified in the existing literature are then used to describe challenges to integrating the approaches and to describe specific influences on program management. After describing the modeling approach and simulation model of an acquisition program, the researchers map the specific influences into changes in model variables. They then use the results of simulations of the evolutionary acquisition program without and with open systems as a basis for a discussion of both the needs for successful programs that use both approaches, as well as the use of simulation modeling as a tool for investigating these acquisition-implementation issues. The paper closes with recommendations for future work.

Evolutionary Acquisition

In the year 2000, the Defense Department promulgated the term "evolutionary acquisition" (EA) in its policy documents governing the strategy for acquisition of materiel and mandated such strategies be used as the preferred approach to procurement (USD(AT&L), 2000, October 23). Later elaborated as spiral and incremental strategies, these approaches contrast to others that are based on more serial, sequential or singular efforts to arrive at a product solution. The latter are often termed as: single-step-to-full-capability, grand design, big bang, technological leap, waterfall, rational-comprehensive, and the unified development method (Forsberg, Mooz & Cotterman, 2005, p. 354). The overarching goals and principles of the DoD's evolutionary acquisition are to ensure that the Defense Acquisition System provides useful military capability to the operational user as rapidly as possible, and such strategies shall be the preferred approach to satisfying operational needs. Evolutionary acquisition strategies define, develop, and produce/deploy an initial, militarily useful capability ("Block 1") based upon proven technology, time-phased requirements, projected threat assessments, and demonstrated manufacturing capabilities. They also plan for subsequent development and production/deployment of increments beyond the initial capability over time (Blocks II, III, and beyond) (USD(AT&L), 2000, October 23). Figure 1 shows the conceptual difference between a traditional single-step-to-capacity acquisition process and an evolutionary acquisition process with two development blocks, as described in the 1996 and 2003 versions of the *DoD 5000* series.

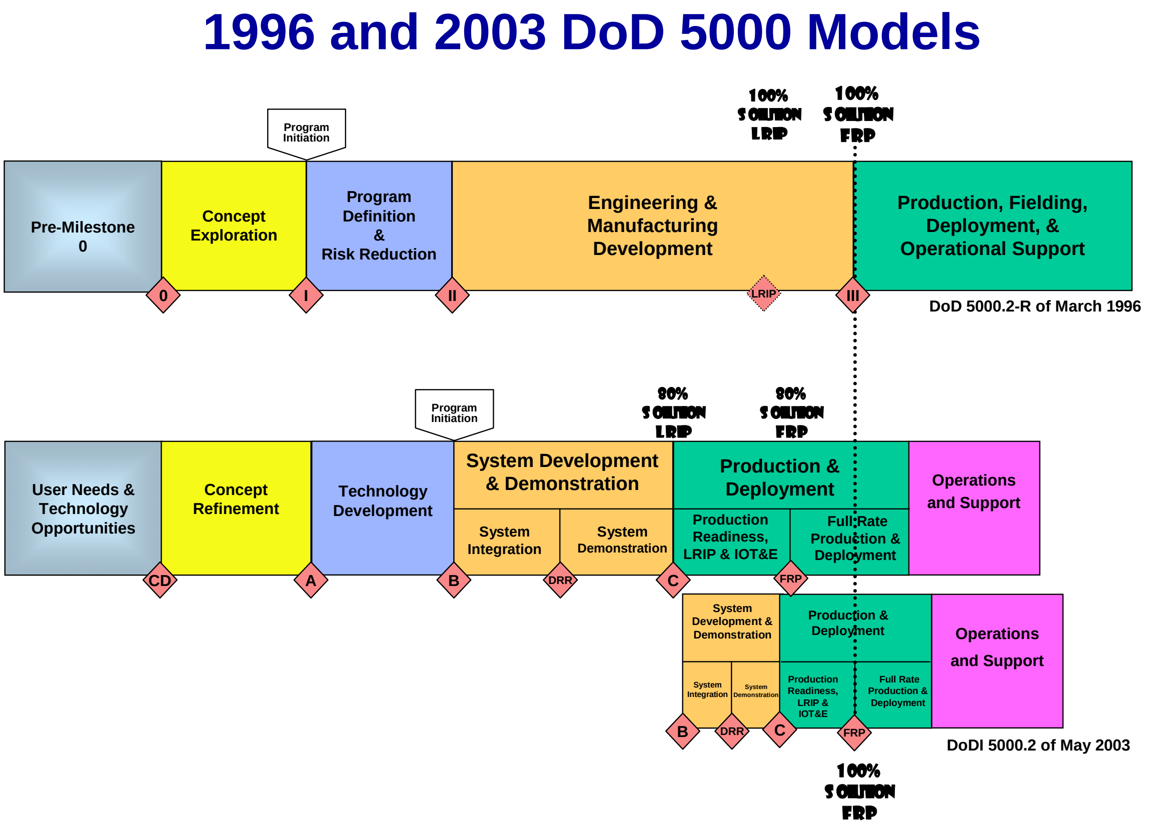

1996 and 2003 DoD 5000 Models
100% SOLUTION LRIP
100% SOLUTION FRP
Program Initiation
Pre-Milestone 0
Concept Exploration
Program Definition & Risk Reduction
Engineering & Manufacturing Development
Production, Fielding, Deployment, & Operational Support
0
I
II
LRIP
III
DoD 5000.2-R of March 1996
80% SOLUTION LRIP
80% SOLUTION FRP
Program Initiation
User Needs & Technology Opportunities
Concept Refinement
Technology Development
System Development & Demonstration
System Integration
System Demonstration
Production & Deployment
Production Readiness, LRIP & IOT&E
Full Rate Production & Deployment
Operations and Support
CD
A
B
DRR
C
FRP
System Development & Demonstration
System Integration
System Demonstration
Production & Deployment
Production Readiness, LRIP & IOT&E
Full Rate Production & Deployment
Operations and Support
B
DRR
C
FRP
DoDI 5000.2 of May 2003
100% SOLUTION FRP

Figure 1. Comparison of Traditional Single Step to Capacity and Evolutionary Acquisition Approaches
(Dillard, 2005)

The policy for evolutionary acquisition was aimed at improving all parameters of program success, but clearly and explicitly, its single most important objective was to reduce long product cycle-times to deliver operationally useful equipment. Figure 1 illustrates the hypothetical earlier start of production and the overlapping development blocks that are characteristic of evolutionary acquisition.

The authors, in their previous work (Dillard & Ford, 2007) investigated implementation challenges of evolutionary acquisition using the same approach we are using in the current work. We found, in part, that an evolutionary development approach significantly increases the number of development phases and activities that must be managed and coordinated at any given time over that required for single-block development. This, consequently, increases the organizational project management resource needs for successful acquisition over those necessary for single-block projects. Using open systems with an evolutionary approach may or may not accentuate these challenges.

Open Systems in DoD Acquisition

Open Systems were made a part of DoD acquisition in *DoD 5000.1* (USD(AT&L), 2003, May 12) which says, “a modular open systems approach shall be employed where feasible” (p. 7). A subsequent memorandum (USD(AT&L), 2004, July 7) clarified the central role of OS in acquisition by saying the approach is “an integral part of the toolset that will help DoD achieve its goal of providing the joint combat capabilities required in the 21st century, including supporting and evolving these capabilities over their total life-cycle” (p. 8). The Open Systems Joint Task Force (OSJTF) leads the DoD OS effort (OSTJF, 2004, September). Several terms defined in that guide are relevant to and used in the current work, including:

- **Open architecture:** An architecture that employs open standards for key interfaces within a system.
- **Open standards**: Standards that are widely used, readily available, consensus-based, published and maintained by recognized industry standards organizations (versus “closed,” which are not).
- **Open system:** A system that employs modular design, uses widely supported and consensus-based standards for its key interfaces, and has been subjected to successful validation and verification tests to ensure the openness of its key interfaces.
- **Open systems environment (OSE):** A comprehensive set of interfaces, services, and supporting formats, plus aspects of interoperability of application, as specified by Information Technology (IT) standards and profiles. An OSE enables information systems to be developed, operated, and maintained independent of application-specific technical solutions or vendor products.

An open systems approach uses the concepts of key versus non-key interfaces and open versus closed interfaces, as defined above, to build flexibility into programs. Figure 2 illustrates potential locations of these interfaces in a conceptual system with modular subsystems/components. The centrality of these concepts to the open systems approach greatly increases the importance of the intended and unintended impacts of a shift away from the traditional focus on customized designs to integration through open interfaces.

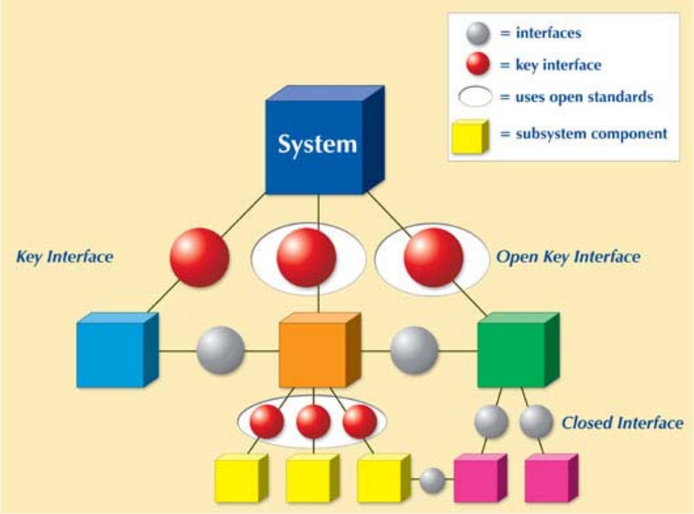


Figure 2. Types of Systems Interfaces
(OSJFT, 2004)

Challenges of Integrating Evolutionary Acquisition and Open Systems

Program managers using open systems and evolutionary acquisition in an integrated fashion may be able to achieve interoperability and insert evolving technologies better than project managers using either approach alone. But, despite their potential, the combination of OS and EA has not yet been fully developed or implemented in DoD acquisition. This is perceived to be largely because the issues related to their implementation have not been completely identified or resolved. This incomplete resolution of implementation for open systems and evolutionary acquisition individually makes understanding their interactions and the impacts of those interactions on acquisition programs difficult. Therefore, the challenges and solutions for implementing either approach are not yet fully understood. For example, the application of OS to hardware/software systems may present particular challenges.

The adoption and use of open systems in DoD acquisition requires several different activities that impact the acquisition process in different ways. Meyers and Oberndorf (2001) identify some of these activities. We describe the most important activities identified by Meyers and Oberndorf with our assessment of their impacts on the evolutionary acquisition process:

1. **Build a baseline of standards and commercial-off-the-shelf (COTS) products.** This change increases the scope of the Block 1 requirements phase and early design (pre-system acquisition) to describe the requirements in terms of standards.
2. **Build a high-level model of the system for use in applying the open systems approach.** This change increases the scope of early design in Block 1.
3. **Document the open architecture in a way that shows the evaluation of alternative architectures, identifies components, technologies, etc.** This change increases the scope of the early design activities and advanced development phases in all Blocks.
4. **Coordinate standards and establish liaisons with standards bodies and users.** This change increases scope of all phases in all blocks because it is an on-going process.
5. **Implement the use of the selected standards in the development process.** This change decreases scope of advanced development phase in all blocks due to component design activities being replaced with component selection.
6. **Integrate components into the product and test the integrated system.** This change increases problems/rework in advanced development and manufacturing phases of all blocks.

Hanratty, Lightsey, and Larson (1999) also investigated the use of open systems in acquisition. They describe the impacts of OS on acquisition as a shift away from design (which, in OS, is completed by the broader commercial market) to integration of elements into products (which, in OS, is increasingly completed with elements that were not developed specifically for the DoD). Hanratty, Lightsey, and Larson identified several areas of open systems design that pose risks, which we describe with our assessment of the primary impacts of OS on evolutionary acquisition processes.

1. **Slower integration and testing of standards-based elements into products.** This change delays the discovery of integration problems until later in projects.

2. **Reduced DoD control over standards.** This change increases the number and size of design problems due to faster evolution of the standard used in the product.
3. **Increased standards-selection risk due to evolution of standards and the possibility that standards will not endure.** This change both increases the number and size of design problems due to the possibility that the selected standard will not endure, and increases testing and integration (regardless of whether problems are discovered or not) due to more frequent changes in standards.
4. **Increased standard change risk—knowing when to shift from one standard to another.** This change increases testing and integration (regardless of whether problems are discovered or not) due to more frequent changes in standards. It also increases the number and size of integration problems that need to be discovered and resolved due to both the need to change to the new standard more often and the possibility of changing too early, too late, or to the wrong standard if more than one are available (e.g., competing for market dominance).
5. **Increased and continuous testing requirements due to the need to integrate evolving commercial and non-developmental items into systems.** This change increases testing and integration (regardless of whether problems are discovered or not) due to more frequent component redesigns.
6. **Development of support concepts early in the acquisition cycle—causing increased standards-selection risk due to large amounts of information needed about currently available standards.** This change increases standards research and planning early in acquisition, which would include increased interface design and management.
7. **Reduced control over detailed component design due to design by industry based on industry-controlled standards.** This change increases the number and size of integration problems due to component designs that do not exactly match product needs.

These specific influences pose significant individual challenges. However, they might also interact in ways that are difficult to predict or immediately recognize and address. In the Model Use section, we describe how we mapped these influences onto specific parts of an acquisition process to better understand how they impact program performance.

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. The current work reflects project, product development, and management theories.

The system dynamics methodology was applied to model a DoD acquisition project with evolutionary processes and open systems. System dynamics uses a computational experimentation approach to understanding and improving dynamically complex systems. The system dynamics perspective focuses on the roles of accumulations and flows, feedback, and nonlinear relationships in managerial control. The methodology’s ability to model many diverse system components (e.g., work, people, money), processes (e.g., design, technology development, quality assurance), and managerial decision-making and actions (e.g., forecasting, resource allocation) makes it useful for investigating acquisition projects. Forrester (1961) develops the methodology's philosophy, and Sterman (2000) specifies the modeling process with examples and describes numerous applications. When applied to development projects, system dynamics focuses on how performance evolves in response to interactions among development strategy (e.g., evolutionary development vs. traditional), managerial decision-making (e.g., scope developed in specific blocks), and development processes (e.g., concurrence). System dynamics is considered appropriate for modeling acquisition projects because of its ability to explicitly model critical aspects of development projects (Ford & Sterman, 1998; Cooper, 1993a,b,c; 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), poor schedule performance (Abdel-Hamid, 1988), and the impacts of changes (Rodriguez & Williams, 1997; Cooper, 1980) and concealing rework requirements (Ford & Sterman, 2003a) on project performance. See Lyneis and Ford (2007) for a review of the application of system dynamics to projects.

The simulation model used here 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 below. Therefore, the model is founded on well-established and tested components. Previous models have developed structures for many components and aspects of acquisition. However, previous models have not been used to investigate the integration of EA and OS in acquisition projects. The current model was originally developed to investigate EA and is described in detail by Dillard and Ford (2007).

A Conceptual Model of an Evolutionary Acquisition Program

The model 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: the development of requirements, the development of technologies, the design of product components, and the manufacture of products. Within a development block, 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. Development phases and information flows in a single block, as depicted in the model is shown in Figure 3. 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. The completion of some requirements allows 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 3 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. These reviews are necessary, but are “off-core” activities that add work beyond that needed to complete the basic products of each phase (requirements, technologies, designs, products, and readiness for use confirmation).

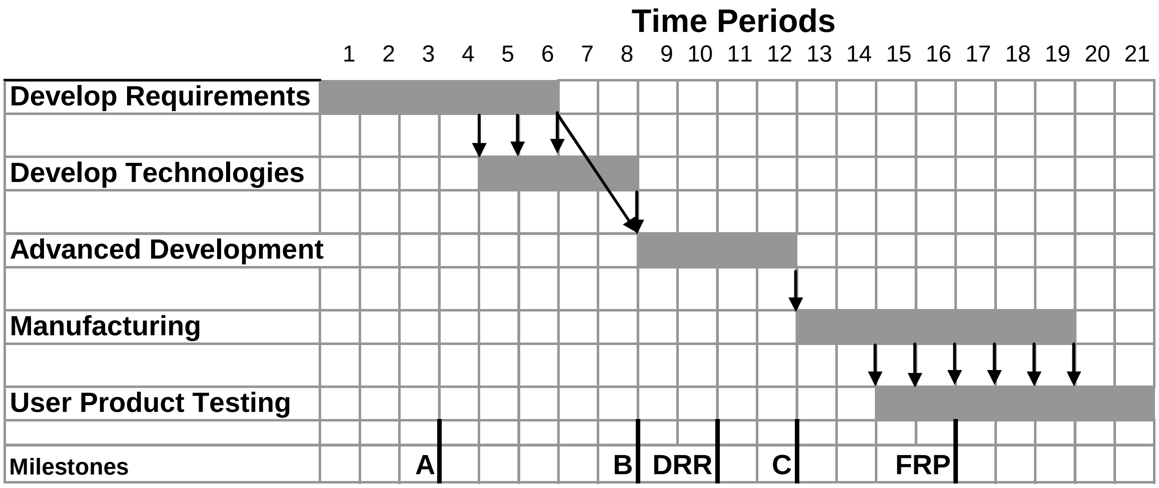


Figure 3. Information Flows in a Single Block Acquisition Project

Figure 4 depicts an acquisition project with multiple iterations or blocks. The first block is the same as Figure 3 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. Importantly, in addition to the flow of information downstream through phases (black arrows in Figure 4), multiple iteration acquisition also provides opportunities for information to flow upstream, such as from User Product Testing in an earlier iteration to Develop Requirements or Advanced Development in a subsequent iteration (red vertical arrows in Figure 4).

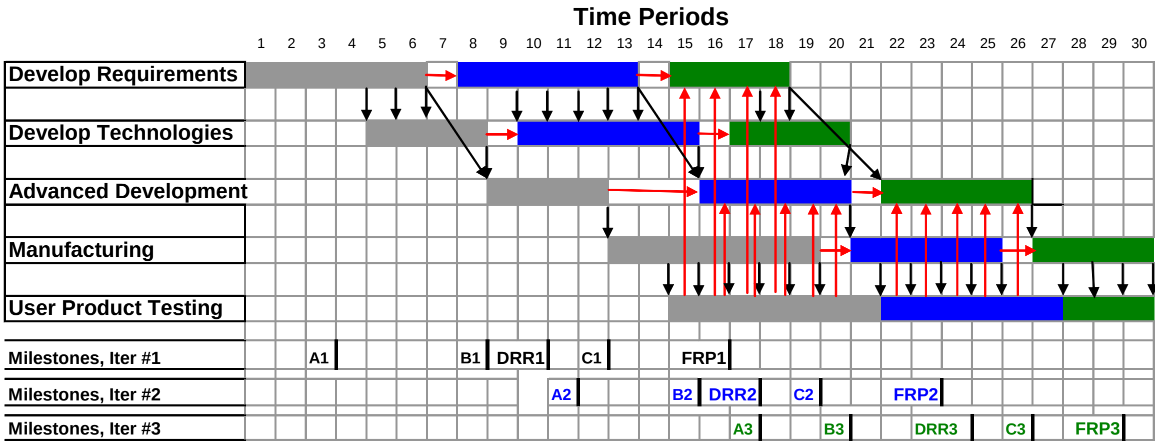


Figure 4. Information Flows in a Three-block Acquisition Project

A Formal Simulation Model of an Evolutionary Acquisition Program

The conceptual model described above was used to build a formal computer simulation model of an acquisition program that can reflect evolutionary acquisition and the use of open systems. See Dillard and Ford (2007) for details. The simulation model is a system of nonlinear differential equations. Each phase is represented by a generic structure, which is parameterized to reflect a specific phase of development.

Project performance is measured in three dimensions: schedule, cost, and product-performance risk. Schedule performance is measured by the time required to test and approve a given number or fraction of requirements by users. Cost is measured in dollars based on the size of direct and indirect work forces and the duration of phases and blocks. Product-performance risk is measured by the average percent of the requirements provided (approved by users) at any given time. This average reflects the combination of multiple requirements. All the requirements can be considered met completely when the average percent of the requirements provided is 100% for a development block.

The formal model was calibrated to the Javelin project described by Dillard and Ford (2007) based on data collected from a manager on the project (the second author) and performance data (e.g., schedule and costs) on the project. The model was tested with the three types of tests of system dynamics models suggested by Forrester and Senge (1980): structural similarity to the actual system, reasonable behavior over a wide range of input values, and behavior similarity to actual systems. The model was found to be useful for investigating the impacts of OS and EA on acquisition projects.

Model Use

To investigate the impacts of opens systems on evolutionary acquisition, we simulated a project similar to the Javelin project twice: first as if the project did not use open systems and then as if the project used an open systems approach. We then compared the behavior and project performance. The program base-case model and simulation described in Dillard and Ford (2007) reflects an evolutionary acquisition program that does not include open systems impacts. To add the impacts of open systems to the model, we first mapped the identified impacts based on Meyers and Oberndorf (2001) onto model variables as follows (Table 1):

Table 1. Impacts of Open Systems on Evolutionary Acquisition due to Changes Suggested by Meyers and Oberndorf (2001)

Change Required by Open Systems	Impact on Evolutionary Acquisition Processes
1) Build standards & COTS for program use	Increases Requirements scope in Block1
2) Build high-level model with open system	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

We also mapped the impacts of required changes to acquisition projects identified by Hanratty, Lightsey, and Larson (1999) onto model variables as follows (Table 2):

Table 2. Impacts of Open Systems on Evolutionary Acquisition due to Changes Suggested by Hanratty, Lightsey, and Larson (1999)

Change Required by Open Systems	**Impact on Evolutionary Acquisition Processes**
7) Slower integration and testing	a1) Reduces problem discovery in Technology Development and Advanced Development phases in all blocks a2) Increases problem discovery in Manufacturing phases in all blocks b1) Decreases problem discovery in earlier blocks (all phases except Requirements) b2) Increases problem discovery in later blocks (all phases except Requirements)
8) Track and change with evolving standards	More problems in Advanced Development and Manufacturing phases in later blocks Increases scope in Technology Development and Advanced Development phases in all blocks
9) Increase testing to discover increased integration problems	Increases scope in Technology Development, Advanced Development, and Manufacturing phases in all blocks
10) Build support system (OSE)	Increased scope in Requirements phase in Block 1

Several of the changes above impact the same portions of an evolutionary process, sometimes in the same directions and sometimes in opposite directions. Therefore, we regrouped the impacts (Table 3) according to model variable that described a specific program block and development phase (e.g., scope of work in Block 1, Requirements Phase). The three variables found to best describe the impacts of open systems on evolutionary acquisition programs are the scope of work, rework fraction, and quality assurance (QA) effectiveness. In the table below and within the model, the scope represents the work that must be completed in a development phase. The Rework Fraction reflects the number of problems that are created in a development phase. The QA effectiveness reflects the difficulty of discovering problems to be resolved. The unit of measure of change was chosen as the percent change from the base case that the use of open systems would cause. This normalizes impacts for different phases (e.g., a change of 10 to a phase with a scope of 50 is very large compared to the same change to a phase with a scope of 5,000) and facilitates assessment of the changes. No known data is available to complete Table 3 based on an actual acquisition program. However, order of magnitude estimates that are in a reasonable rank order of size are adequate because of the preliminary nature of the study. The net changes of all the specific influences are summarized in Table 3. See Appendix A for a more detailed description of the estimates.

Table 3. Estimated Changes in Evolutionary Acquisition Processes to Reflect Open Systems

Program Block and Phase	Scope of Work	Rework Fraction	QA Effectiveness
DEVELOPMENT BLOCK 1			
Requirements	+7	0	0
Develop Techn.	-15	0	-10
Advanced Dev.	-17	-5	-10
Manufacturing	+2	+5	+5
Testing by User	+1	0	-5
Net Change from Base Case	**-0.22**	**0%**	**-20%**
DEVELOPMENT BLOCK 2			
Requirements	+1	0	0
Develop Techn.	-16	0	-5
Advanced Dev.	-17	0	-5
Manufacturing	+2	+10	+10
Testing by User	+1	0	0
Net Change from Base Case	**+29%**	**+10%**	**0%**
DEVELOPMENT BLOCK 3			
Requirements	+1	0	0
Develop Techn.	-16	0	0
Advanced Dev.	-17	+5	0
Manufacturing	+2	+15	+15
Testing by User	1	0	+5
Net Change from Base Case	**+29**	**+20**	**+20**

Simulation Results

Figure 5 shows a plot of the simulated percent of project requirements provided to users by the acquisition program without open systems (Line 1) and with open systems (Line 2). The simulated program has three development blocks, and the simulation clearly shows the evolutionary acquisition nature of the program—with three increases in requirements provided as each development block is completed. The simulation also shows the program with open systems provides as many or more requirements at any point in time than the program without open systems. This supports the open systems approach claim that it can facilitate providing more requirements faster.

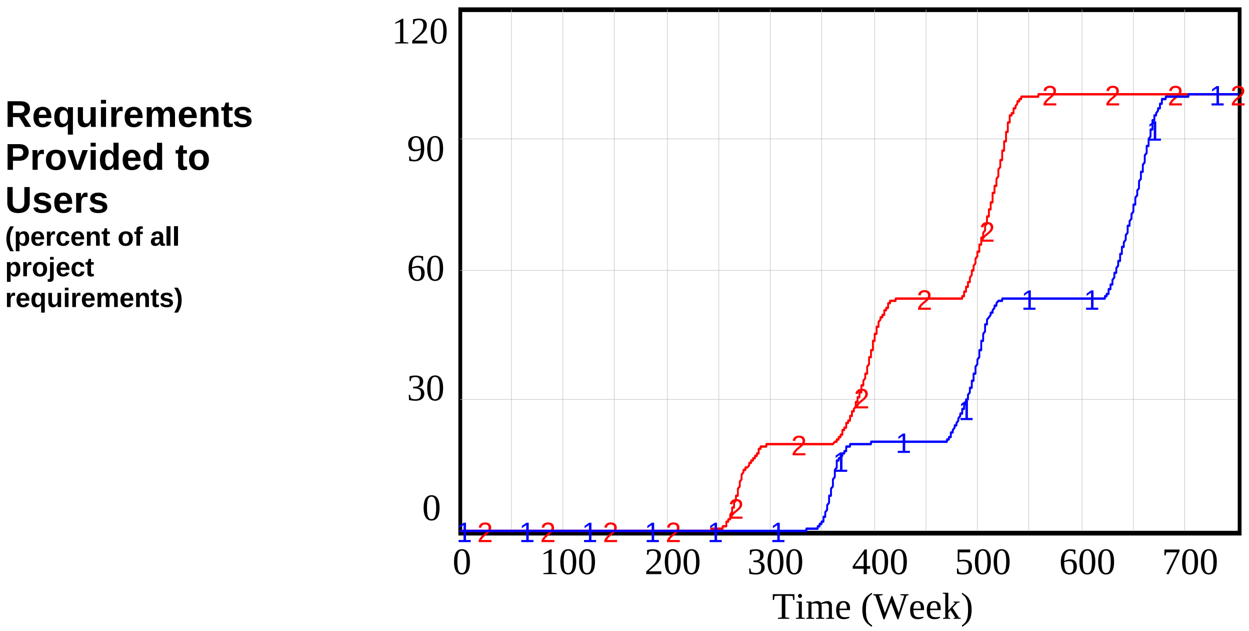


Figure 5. Requirement Fulfillment with Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

In addition to supporting the potential gains available through evolutionary acquisition and open systems, the simulation describes the interaction of evolutionary acquisition and open systems in more detail, providing the opportunity for improved understanding. The simulation shows that the improvement in time-to-requirement increases with each block, indicating that open systems can improve this dimension of program performance during multiple development blocks. An ***open systems approach may leverage its benefits when used with evolutionary acquisition through repeated capture of benefits generated in early development blocks in subsequent development blocks.*** If an OS approach is implemented with EA, programs may be able to reap the benefits first achieved in earlier blocks in subsequent downstream blocks, effectively benefitting more than once for the open systems work done early.

However, time to delivery of requirements is only one measure of program performance. Cost is another important performance measure. The simulated program without open systems costs $5.39 million through complete release to users, and the program with open systems costs $3.84 million through complete release to users.[1] Reduced costs are an established potential benefit of using open systems, largely through reduced design scope. This is the case in the model, in which a significant reduction in design scope is assumed to be a fundamental impact of using open systems. However, the simulation points out an additional potential cost benefit of using open systems. Shorter programs tend to cost less (all other things held equal). Therefore, ***open systems can improve cost performance by interacting with evolutionary***

[1] Actual cost savings may be significantly different due to smaller reductions in design scope.

acquisition to enhance the schedule performance available through evolutionary acquisition alone.

A third important performance measure is the quality of the developed product. Less-than-desired quality can be caused by many things, including not or partially fulfilling requirements, design errors that reduce product performance or increase operations or maintenance costs, and integration errors that make future upgrades difficult, slow, or expensive. Design and integration errors are particularly important in the current work because of their central role in open systems. Acquisition program changes required by open systems clearly alter the nature, number, and timing of both design and integration errors. Generally, early design errors are expected to be reduced, but later integration errors may increase due to evolving standards. Errors that are discovered and addressed during an acquisition program are not as problematic as those that remain after the product has been put into service. Undiscovered and released errors are problematic because they can severely increase operations, maintenance, and upgrade costs.

The model was used to simulate the number of undiscovered errors in released work without and with open systems. Figure 6 shows the evolution of the number of undiscovered and released errors as a percent of the program scope. In general, the number of released errors increases as work is completed until the next development phase begins receiving development work, finding errors, and returning them to upstream phases for resolution.

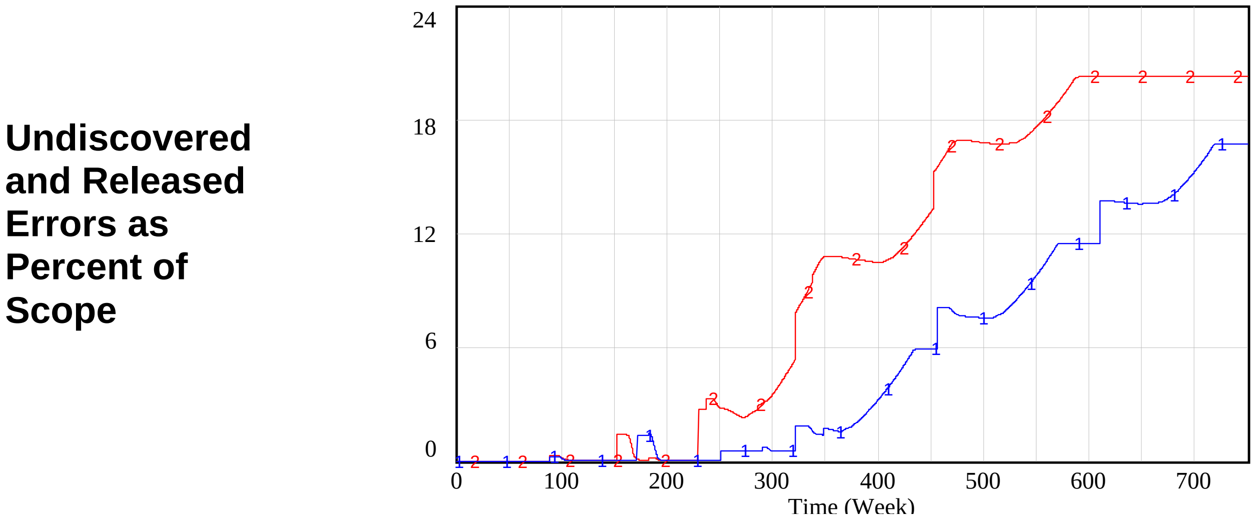


Figure 6. Undiscovered Problems in Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

Figure 6 shows that the simulated project with open systems generates and fails to find and resolve more errors before release. To further investigate this, the errors were disaggregated into design errors and integration errors based on the assumption that errors in the early development phases of each block (requirements and technology development and advanced development) are primarily design errors, and errors in manufacturing and testing are primarily integration errors. Figure 7 shows the undiscovered and released design errors as a percent of scope with and without open systems, and Figure 8 shows the undiscovered and released integration errors as a percent of scope with and without open systems. Note that the vertical scale in Figure 8 (0-20%) is four times larger than the vertical scale in Figure 7 (0-5%) for clarity.

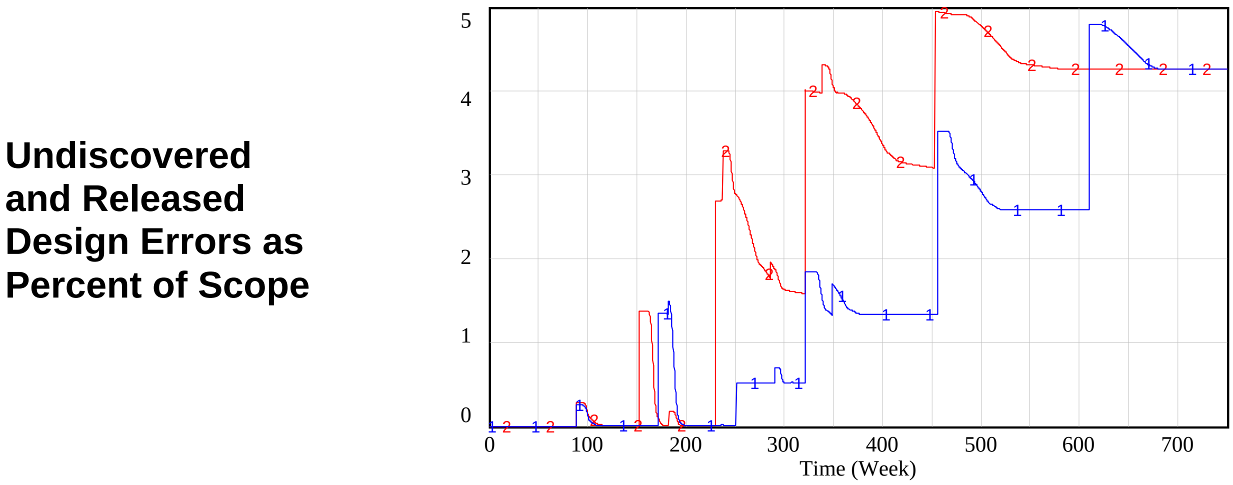


Figure 7. Undiscovered and Released Design Errors in Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

The differences in the timing of when design errors are generated, discovered and resolved, or missed and released is primarily due to the faster development with open systems. More importantly, the total percent of design errors at the completion of the program is nearly the same for the two programs. This suggests that the important impacts of open systems on evolutionary acquisition may be found in design errors.

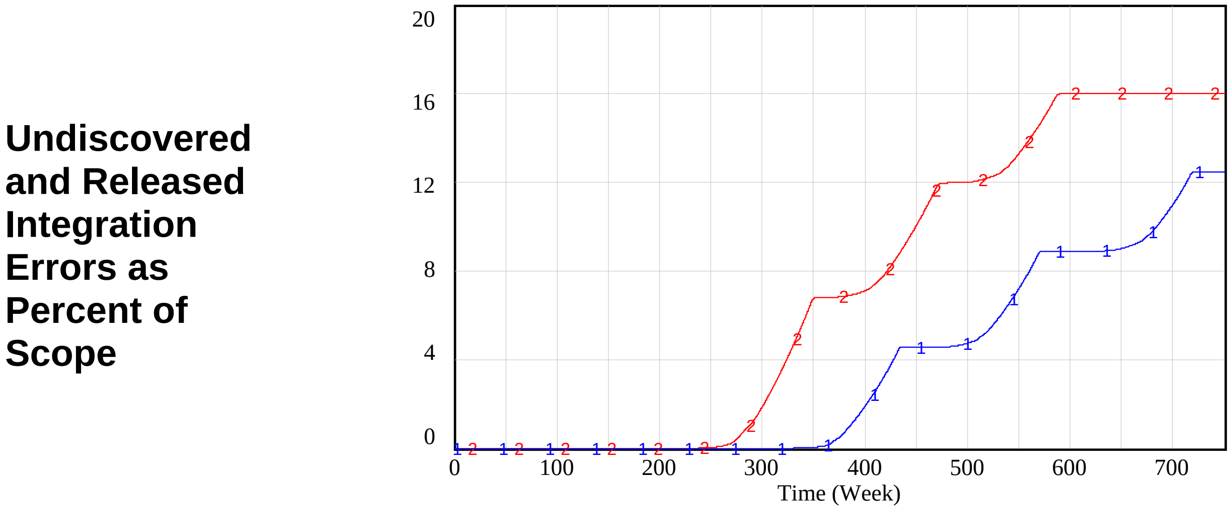


Figure 8. Undiscovered Integration Errors in Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

There are at least two important differences between the number of undiscovered and released design errors (Figure 7) and the number of undiscovered and released integration errors (Figure 8). First, the programs generated and failed to resolve three to four times as many integration errors than design errors. This suggests that PMs using open systems must address integration issues if they wish to succeed. This finding also supports the importance of the shift from design to integration identified by other investigators. Second, the program with open systems generated at least 25% more integration errors than the program without open systems (3+% more than 13%). This difference in integration errors explains essentially the entire difference in total undiscovered and released errors (Figure 6).

In summary, the simulation results show that open systems can interact with evolutionary acquisition to improve the timing of products (Figure 5), reduce development costs, and increase the number of undiscovered and released integration errors (Figures 6-8). This suggests that ***open systems and evolutionary acquisition can interact to improve schedule and cost performance, but that these benefits may come at the cost of increased risk of high operations, maintenance, and upgrade costs when the integration errors are eventually discovered and must be resolved.***

Implications for Evolutionary Acquisition Practice with Open Systems

The identification of impacts of open systems on evolutionary acquisition programs and the simulation results carry potentially valuable implications for acquisition program managers.

Shifting the Types and Amounts of Risk

Adding open systems to evolutionary acquisition shifts the program management focus from design to standards and integration. This impacts when the program accepts and must manage different types and amounts of risk. Open systems reduce design risks by designing components, subsystems, and systems to be consistent with established standards. Design risk is also reduced, as an OS approach uses pre-designed and pre-tested components that have been designed and tested to established standards. Open systems may increase other risks, however. Standards selection and change risks are increased because programs using open systems are more dependent on standards than programs using customized designs; OS also have little influence over the evolution of those standards. Integration risks may increase significantly as standards change over the product lifecycle, and new standards may not be compatible with the current design of products. Different types of skills are needed to manage different types of risk. For example, detailed component design risk management requires technical expertise for design review and component testing, but integration risk management requires a broader, systems understanding of the product and how subsystems work together to fulfill requirements. Acquisition programs using open systems need a different set of risk-management skills than programs not using open systems. Less-detailed technical expertise will likely be needed, and more integration and systems expertise will be needed. ***If open systems are integrated into evolutionary acquisition (which repeats the development process over multiple blocks), then acquisition programs will require significant and extended integration and systems expertise.*** This will also change the skill sets needed by the DoD acquisition workforce.

A Temporal Shift in Program Risks

Design risks occur relatively early in programs and product lifecycles, whereas integration risks occur relatively late. Therefore, the use of open systems will shift program risk to emerge later in projects. The simulations support this result with the increase in the number of undiscovered and released integration errors with open systems. If costs follow risk, this may result in lower development costs due to lower design risk, but higher operating, maintenance, and upgrade costs due to higher integration risk. Figure 9 describes the relative costs in a product lifecycle. Integration of OS into EA may reduce Research and Development costs when programs can capture design benefits, but may increase Operating and Support costs when integration and evolving standards risks may increase costs. The sizes of these cost changes are uncertain, but the potential for early reductions in cost and later increases in cost are real.

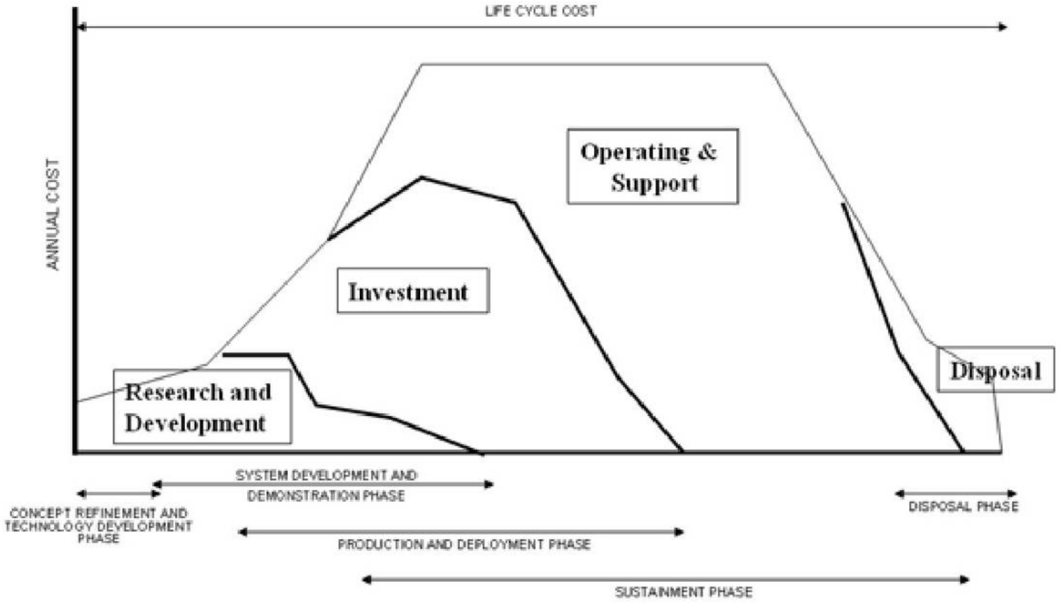


Figure 9. Relative Costs during a Product Lifecycle
(Defense Acquisition Guidebook, 2004, November, p. 43)

By stretching acquisition across multiple blocks, evolutionary acquisition may accentuate the impacts of a temporal shift in program risk. Therefore, if using open systems causes this temporal shift in risks, then programs integrating open systems and evolutionary acquisition may experience an increase in the relative size of product costs during use.

Trading Design Obsolescence for Integration Obsolescence

Traditional acquisition processes commit programs to customized designs and, therefore, bear significant design obsolescence risk when threats and technologies evolve away from the design. An open systems approach can reduce that risk by allowing the use of more plug-and-play components that can be replaced with improved components that meet the chosen standard. However, by using open systems, a program must also commit to one or more standards early in development and, therefore, bear significant standards obsolescence risk if and as standards evolve away from the needs of the program and as integration problems increase. ***Evolutionary acquisition's need for integration across multiple development blocks can increase the impact of open systems on obsolescence risk. Adding open systems to evolutionary acquisition may cause programs to trade away design risk for increased integration risk.***

Conclusions

The current work has extended and expanded the descriptions of the impacts of using open systems and evolutionary acquisition together on development processes and management. We then mapped those impacts into a computer simulation model and used that model to investigate how open systems and evolutionary acquisition interact. Results include that the changes required to implement open systems in evolutionary acquisition significantly impact development processes and management, particularly scopes of design, standards, and integration work, the generation of different types of problems, and the timing of the discovery of problems. The shift from a focus on design to a focus on integration was found to be particularly important. Simulation reinforced the potential for open systems to accelerate acquisition and revealed a potentially important distinction between design and integration errors in explaining the impacts of required changes. Implications for practice included shifts in the type and timing of risks due to open systems use and the possibility of trading design obsolescence for integration obsolescence (e.g., compatibility).

This research has contributed to the understanding of open systems and evolutionary acquisition is several ways. The work improved the description and specification of impacts of acquisition policy on acquisition practice. The work also used dynamic computer simulation to model and investigate open systems and to model evolutionary acquisition and open systems together, both for the first time to our knowledge. The results of the simulation reinforced several suggested impacts of open systems and provided additional causal rationale behind why suggested impacts may occur. These rationales were the basis of potential implications for the evolutionary acquisition practice with open systems. The reasoning provided based on the computer simulation can be used to extend and deepen decision-makers' understanding of open systems and evolutionary acquisition and design program processes and management.

Future researchers can improve and extend the work described here by gathering additional data about the use of open systems with evolutionary acquisition in practice and, in so doing, testing the existence and importance of suggested impacts. The similarity of the model and, thereby, confidence in results can be improved by using additional acquisition projects that use both evolutionary acquisition and open systems.[2] Finally, additional recommendations for practice can be developed based upon the model developed here and elsewhere. These investigations can further develop the understanding of how to effectively integrate open systems and evolutionary acquisition and, consequently, improve the systems and products provided to warfighters.

[2] The authors are currently working with a large navy acquisition project to do this.

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.

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.

Dillard, J.T. (2005, August-November). Toward centralized control of defense acquisition programs: A comparative review of the framework from 1987 – 2003*. Acquisition Review Journal, 12*(3), 331-344.

Dillard, J.T., & Ford, D.N. (2007, April 30). *From amorphous to defined, balancing the risks of evolutionary acquisition* (NPS-AM-07-002). Monterey, CA: Naval Postgraduate School.

DoD. (2004, November). *Defense acquisition guidebook*. Retrieved April 4, 2008, from https://akss.dau.mil/DAG/Guidebook/IG_c4.4.1.asp)

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.

Forrester, J., & Senge, P. (1980). Tests for building confidence in system dynamics models. *TIMS Studies in the Management Sciences*, *14*, 209-228.

Forsberg, K., Mooz, H., & Cotterman, H. (2005). *Visualizing project management* (3rd ed.). Hoboken, NJ: John Wiley & Sons.

Hanratty, M., Lightsey, R.H., & Larson, A.G. (1999, Winter). Open systems and the systems engineering process. *Acquisition Review Quarterly, 6*(4), 47-59.

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.

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

Open Systems Joint Task Force (OSJTF). (2004, September). *Open systems joint task force (OSJTF) program manager’s guide: A modular open systems approach (MOSA) to acquisition*. Retrieved April 9, 2008, from http://www.acq.osd.mil/osjtf/mosapart.html.

Rendon, R.G. (2006, January). *Using a modular open systems approach in defense acquisitions: Implications for the contracting process* (NPS-GSBPP-06-007). Monterey, CA: Naval Postgraduate School.

Rodriguez, A., & Williams, T.M. (1997). System dynamics in project management: Assessing the impacts of client behavior on project performance. *Journal of the Operational Research Society*, *49*, 2-15

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

USD(AT&L). (2000, October 23). *Operation of the defense acquisition system* (DoD Directive 5000.2). Washington, DC: Author.

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

USD(AT&L). (2004, April 5). *Amplifying DoDD 5000.1. Guidance regarding modular open systems approach (MOSA) implementation.* Memorandum. Washington, DC: Author.

USD(AT&L). (2004, July 7). *Instructions for Modular Open Systems Approach (MOSA) implementation*. Memorandum. Washington, DC: Author.

Appendix A. Mapping Specific Influences of Open Systems onto Evolutionary Acquisition Programs Processes

The researchers estimated the impact of each specific, identified and described influence on the scope of work, rework fraction, and quality assurance (QA) effectiveness. They measured the scope of work by the number of equal-sized work packages that must be completed in a development phase. They measured the rework fraction by the percent of those work packages that require changes; this measurement reflects the number of problems that are created in a development phase. They measured the QA effectiveness with the fraction of the work packages discovered to need rework. Although no known data is available as a basis for the estimated changes, order of magnitude estimates that are in a reasonable rank order of size are adequate because of the preliminary nature of the study. To facilitate mapping of the specific influences above to model changes, the researchers listed the specific influences after the individual impacts on each model parameter.

Table 4. Detailed Estimated of Changes in Evolutionary Acquisition Processes to Reflect Open Systems

Program Block and Phase	Scope of Work	Rework Fraction	QA Effectiveness
DEVELOPMENT BLOCK 1			
Requirements	+1+1+5 (1,4,10)	0	0
Develop Techn.	+1+1+1-20 +1+1(1,2,3,5,8,9)	0	-5 -5 (7a,7b)
Advanced Dev.	+1-20 +1+1 (4,5,8,9)	-10 +5(5,6)	-5 -5 (7a,7b)
Manufacturing	+1 +1(4,9)	+5 (6)	+10 -5 (7a,7b)
Testing by User	+1 (4)	0	-5 (7b)
Net Change in Base Case	**-22**	**0**	**-20**
DEVELOPMENT BLOCK 2			
Requirements	+1 (4)	0	0
Develop Techn.	+1+1 -20+1+1 (3,4,5,8,9)	0	-5 (7a)
Advanced Dev.	+1-20 +1+1 (4,5,8,9)	-10 +5 +5(5,6,8)	-5 (7a)
Manufacturing	+1 +1(4,9)	+5 +5 (6,8)	+10 (7a)
Testing by User	+1 (4)	0	0
Net Change in Base Case	**29**	**10**	**0**
DEVELOPMENT BLOCK 3			
Requirements	+1 (4)	0	0
Develop Techn.	+1+1-20+1+1 (3,4,5,8,9)	0	-5 +5 (7a,7b)
Advanced Dev.	+1-20+1 +1(4,5,8,9)	-10 +5+10 (5,6,8)	-5 +5 (7a,7b)
Manufacturing	+1+1 (4,9)	+5 +10 (6,8)	+10 +5 (7a,7b)
Testing by User	+1 (4)	0	+5 (7b)
Net Change in Base Case	**29**	**20**	**20**

THIS PAGE INTENTIONALLY LEFT BLANK

2003 - 2008 Sponsored Research Topics

Acquisition Management

- Software Requirements for OA
- Managing Services Supply Chain
- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Portfolio Optimization via KVA + RO
- MOSA Contracting Implications
- Strategy for Defense Acquisition Research
- Spiral Development
- BCA: Contractor vs. Organic Growth

Contract Management

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

Financial Management

- PPPs and Government Financing
- Energy Saving Contracts/DoD Mobile Assets
- Capital Budgeting for DoD
- Financing DoD Budget via PPPs
- ROI of Information Warfare Systems
- Acquisitions via leasing: MPS case
- Special Termination Liability in MDAPs

Human Resources

- Learning Management Systems
- Tuition Assistance
- Retention
- Indefinite Reenlistment
- Individual Augmentation

Logistics Management

- R-TOC Aegis Microwave Power Tubes
- Privatization-NOSL/NAWCI
- Army LOG MOD
- PBL (4)
- Contractors Supporting Military Operations
- RFID (4)
- Strategic Sourcing
- ASDS Product Support Analysis
- Analysis of LAV Depot Maintenance
- Diffusion/Variability on Vendor Performance Evaluation
- Optimizing CIWS Lifecycle Support (LCS)

Program Management

- Building Collaborative Capacity
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Terminating Your Own Program
- Collaborative IT Tools Leveraging Competence

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

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

www.acquisitionresearch.org

Acquisition Research Program:
Creating Synergy for Informed Change

Modeling the Integration of Open Systems and Evolutionary Acquisition

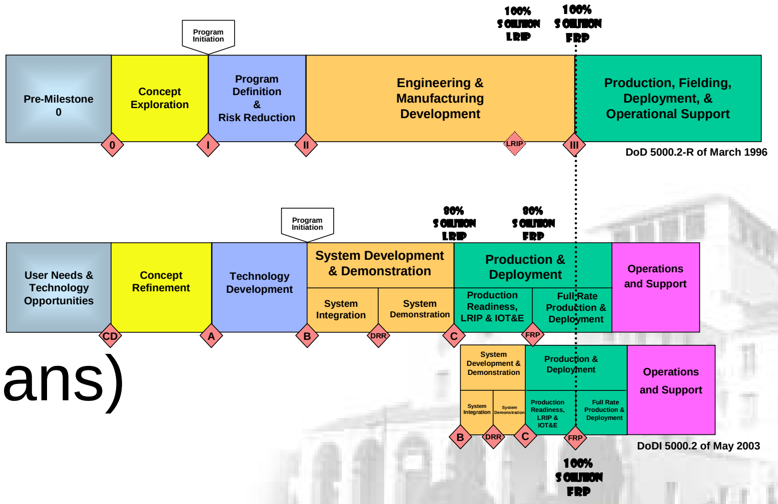
David N. Ford
Texas A&M University and
Naval Postgraduate School

John T. Dillard
Naval Postgraduate School

Introduction: Evolutionary Acquisition (EA)

- Multiple development blocks (vs. single step to full capacity)
- Concurrent development across development blocks (vs. sequential programs)
- Insert only adequately mature (TRL7) technologies
- Unspecified spirals are part of programs and become iterations (vs. independent development plans)

Single-step Acquisition (top) & Evolutionary Acquisition (bottom)



DoD 5000.2-R of March 1996

DoDI 5000.2 of May 2003

Introduction: Open Systems (OS)

- Select and commit to open (industry developed and maintained) standards – ***requires managing programs to external standards***

- Replace some customized design with COTS components, sub-systems, and systems

- Design open key interfaces to increase competition and allow systems and components to evolve with reduced impacts – ***requires managing interfaces***

- Program management shift from design to integration

Problem: Integrating Open Systems and Evolutionary Acquisition

- Great potential for open systems and evolutionary acquisition to synergistically support each other and improve acquisition program performance. Both…
  - Seek to reduce acquisition cycle time
  - Address interoperability
  - Provide flexibility to manage uncertainty in technologies & threats
- But benefits have not been fully captured – ***why?***
  - Both involve complex development processes that interact
  - Integration is difficult
- ***Simultaneous implementation of OS and EA is a major acquisition challenge…***
  - How do the requirements of OS and EA impact each other?
  - How do those interactions impact program performance?

Case Study: P-8 Poseidon Multi-Mission Aircraft Program

- Existing ASW P-3 fleet is approaching end of service life
  - Requires replacement
  - Continues to evolve and add capacity
- Opportunity to increase and improve capacities and performance (e.g. speed, altitude)
- Boeing selected in 2004 based on militarization of 737-800 aircraft
- Currently in SDD of baseline program

Case Study: P-8 Poseidon Program Evolutionary Acquisition and Open Systems

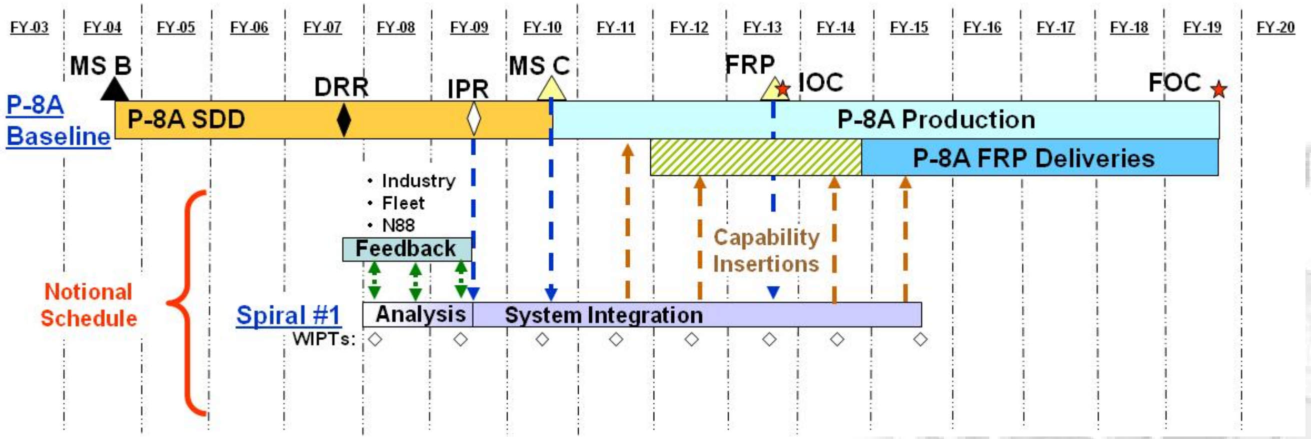
P-8 Acquisition Philosophy: "Design a baseline platform with significant physical and virtual capacity for future growth."

1) Excess power, cooling, and payload carrying capacity coupled with an open systems design allows for Spiral Acquisition of capability
2) Leverage on-going P-3 mission system development and design where possible to develop once and integrate twice

Evolutionary Acquisition Plan (Spirals)

1) **Baseline program** – integrate existing P-3 capabilities into P-8 aircraft (in progress)
2) **Spiral 1** – candidate list of capabilities identified, APB under development, WIPT formed and working, *No impact to baseline program*
3) **Spirals 2+** integrate evolving ASW/ASuW/ISR capabilities into P-8, in preliminary planning

How can the P-8 program integrate open systems and evolutionary acquisition to capture their potential benefits?

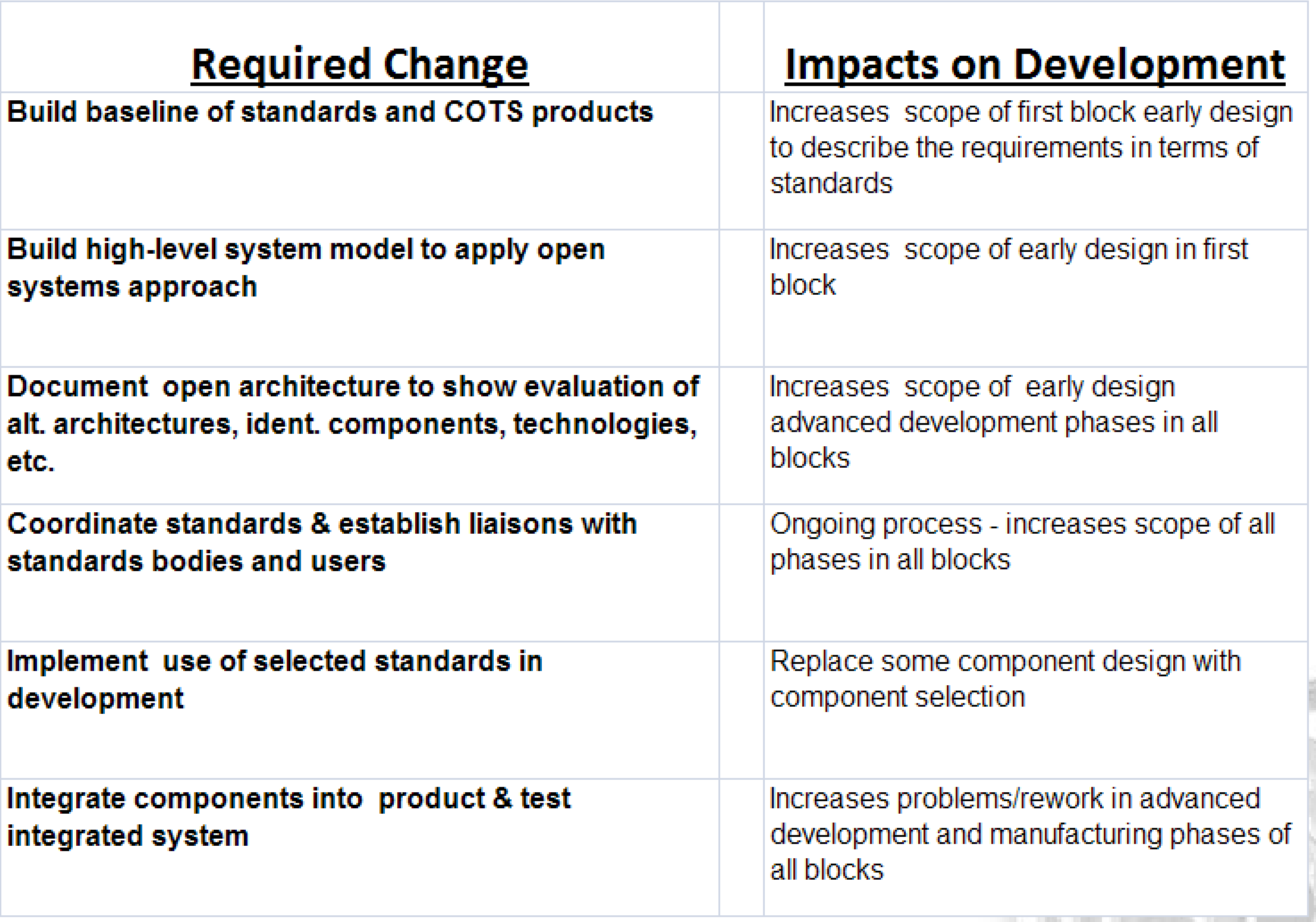


Research Plan

1. **Identify potentially important interactions and impacts with modeling. Use model to find initial lessons**
2. Look for modeling lessons in active program (P-8). Validate and improve model.
3. Use model to design and test EA/OS program management strategies

- Map changes required by open systems into impacts on an evolutionary acquisition process
- Model the impacts of open systems use on an evolutionary acquisition process
  - Map impacts onto changes in model variables
  - Simulate an evolutionary acquisition program with and without open systems
  - Compare behaviors of simulated programs

Changes Required by Open Systems and Impacts on Evolutionary Acquisition Programs



Required Change		Impacts on Development
Build baseline of standards and COTS products		Increases scope of first block early design to describe the requirements in terms of standards
Build high-level system model to apply open systems approach		Increases scope of early design in first block
Document open architecture to show evaluation of alt. architectures, ident. components, technologies, etc.		Increases scope of early design advanced development phases in all blocks
Coordinate standards & establish liaisons with standards bodies and users		Ongoing process - increases scope of all phases in all blocks
Implement use of selected standards in development		Replace some component design with component selection
Integrate components into product & test integrated system		Increases problems/rework in advanced development and manufacturing phases of all blocks

Based on Meyers and Oberndorf (2001):

Changes Required by Open Systems and Impacts on Evolutionary Acquisition Programs

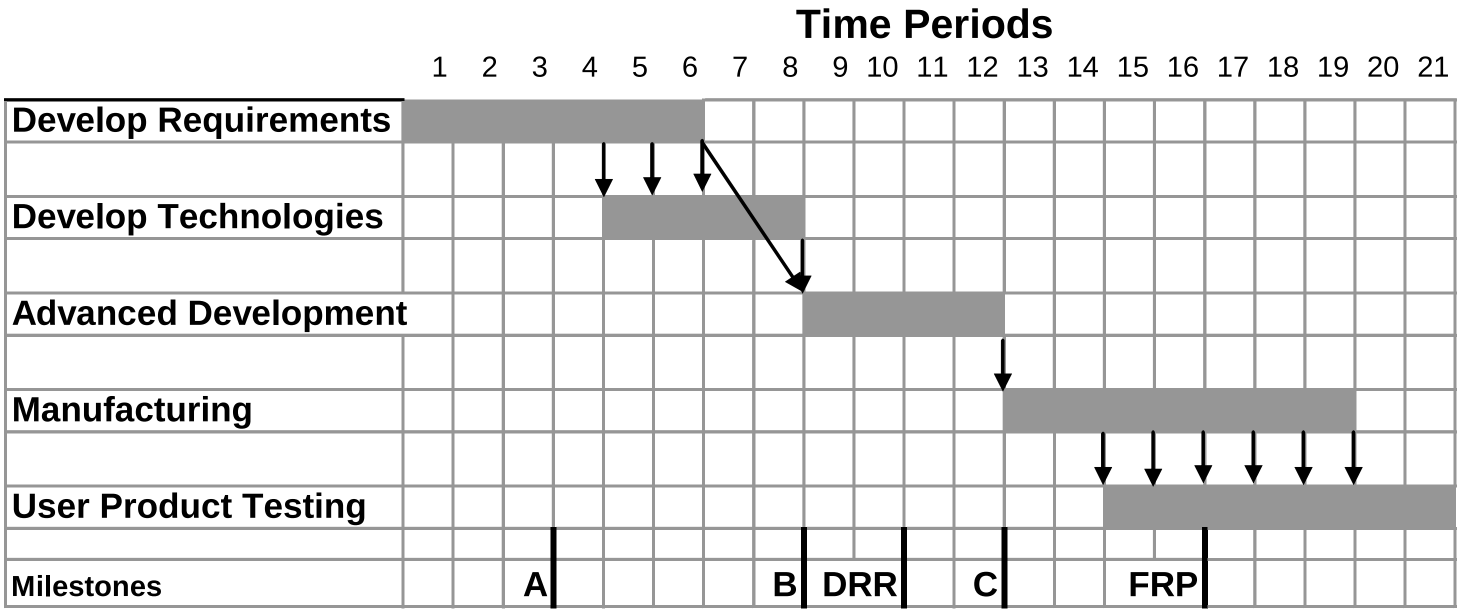
Required Change	Impacts on Development
Slower integration & testing of standards-based elements	Delays discovery of integration problems
Reduced DoD control over standards (Faster evolution of the standards in directions less likely to support the program)	Increases number & size of design problems
Standards evolve and chosen standards may not endure - increased standard choice risk. More frequent standard changes	Increases number & size of design problems & Increases testing and integration
More difficult to know when to shift from one standard to another - increased standards change and choice risk	Increases testing and integration & Increases number & size of integration problems to be discovered and resolved
Increased integration needs due to more and evolving commercial and non-developmental items	Increaseed and continuous testing requirements
Development of support concepts early in the acquisition cycle - increased standards selection risk	Increases standards research and planning early in acquisition & increased interface design and management.
Component design by industry based on industry-controlled standards - reduced control over detailed component design	Increases number & size of integration problems

Based on Hanraty, Lightsey, and Larson (1999):

Issues in Integrating Open Systems into Evolutionary Acquisition - *Summary*

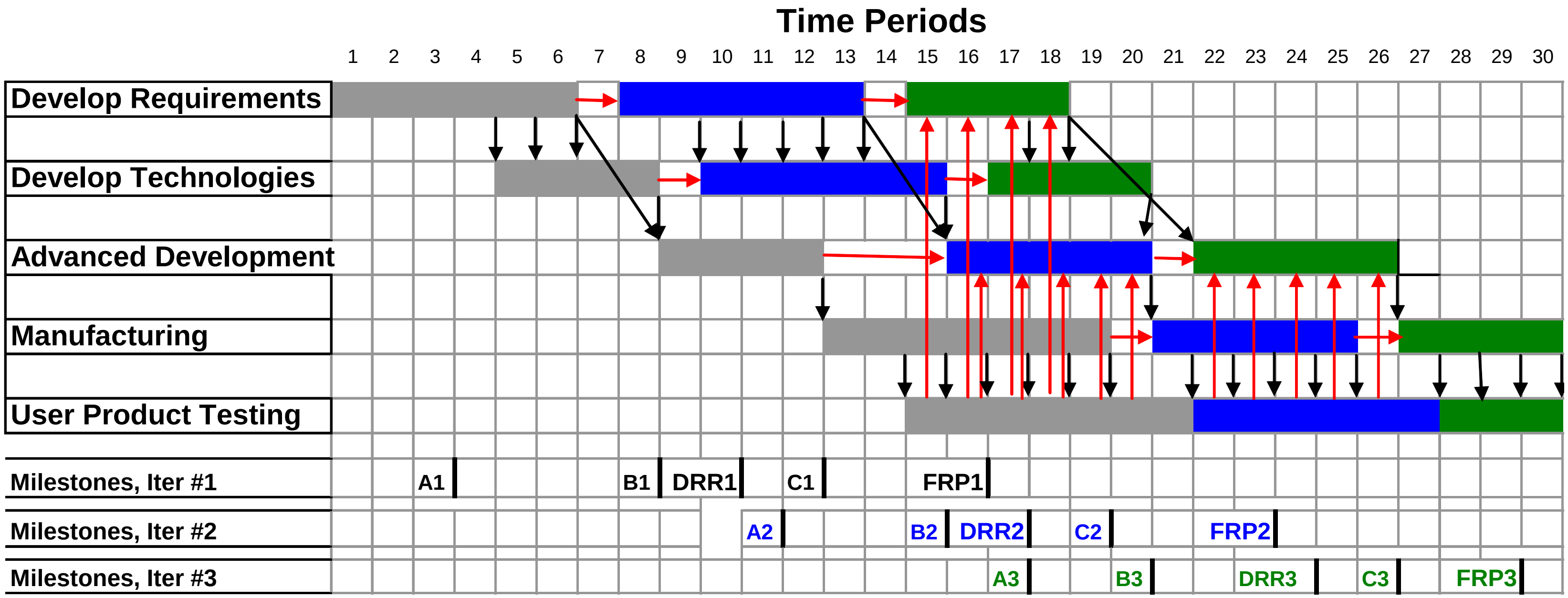
- **Shift in acquisition management from design to integration**
  - Reduced design capacity needed (e.g. for COTS components & systems)
  - Increased integration capacity needed (e.g. for testing)
  - Delays in discovery of problems
- **Program "openness" is a new and critical program management need**
  - Selection, monitoring, using, and documenting use of industry standards
  - Different and new opportunities and risks

Research Method: Simulation Model



Information Flows in a Single Block of an Evolutionary Acquisition Project

Research Method: Simulation Model



Time Periods

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Develop Requirements

Develop Technologies

Advanced Development

Manufacturing

User Product Testing

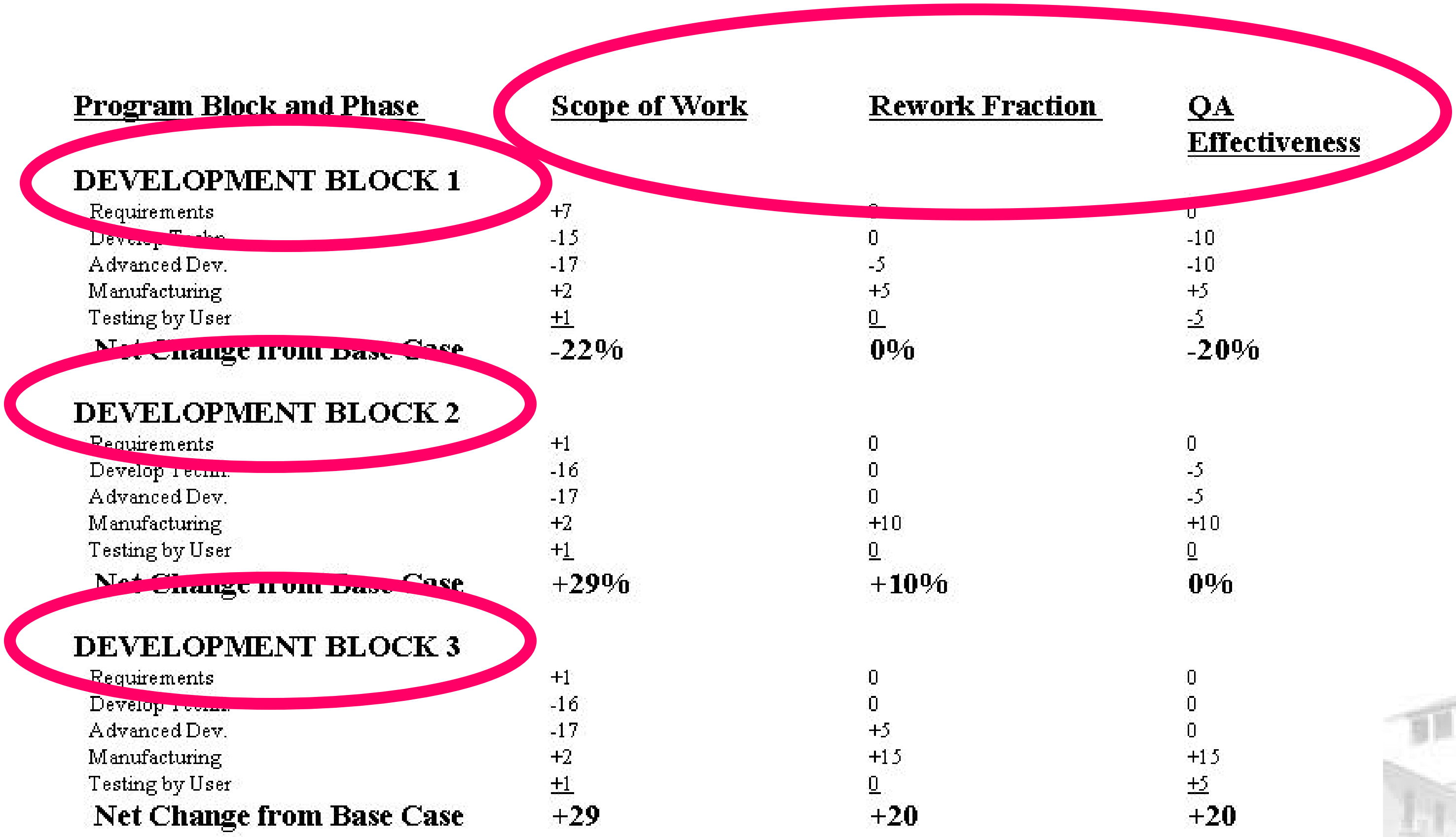
Milestones, Iter #1: A1, B1, DRR1, C1, FRP1

Milestones, Iter #2: A2, B2, DRR2, C2, FRP2

Milestones, Iter #3: A3, B3, DRR3, C3, FRP3

Information Flows in a Three-Block Evolutionary Acquisition Project

Mapping Open Systems Impacts into the Simulation Model

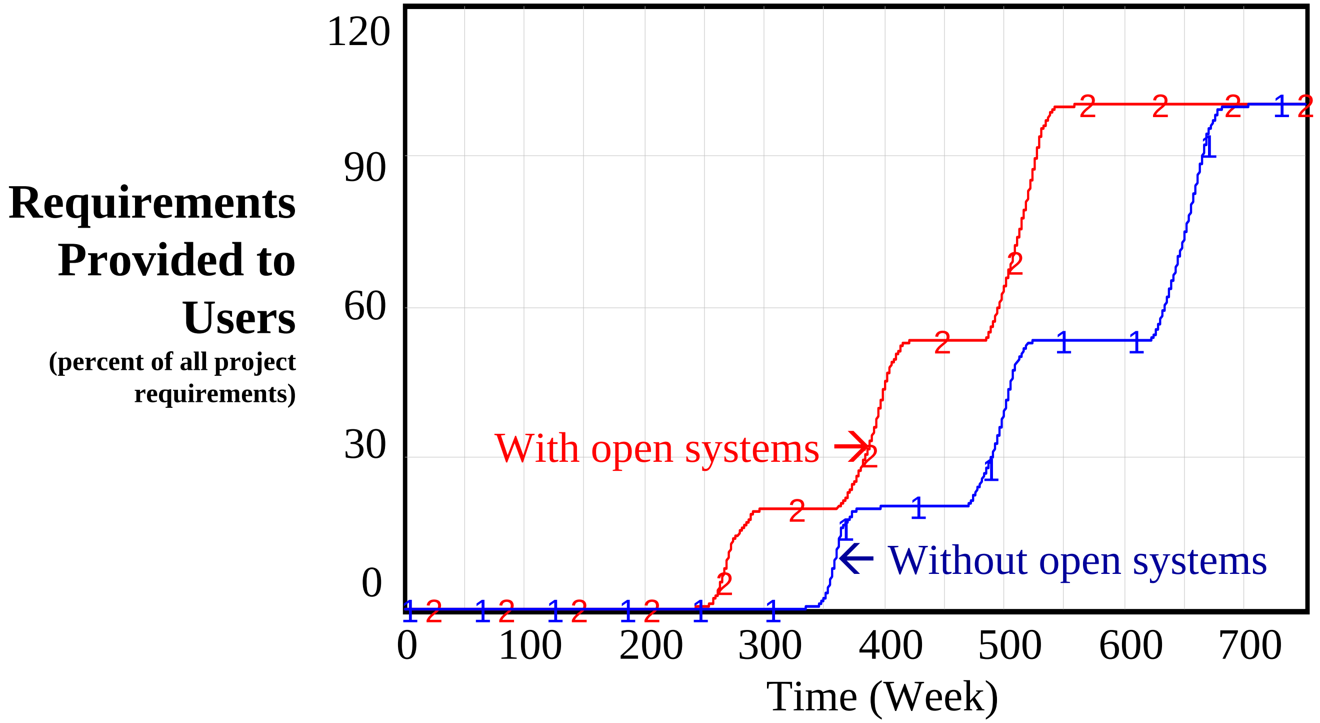


Program Block and Phase	Scope of Work	Rework Fraction	QA Effectiveness
DEVELOPMENT BLOCK 1			
Requirements	+7	0	0
Develop Techn.	-15	0	-10
Advanced Dev.	-17	-5	-10
Manufacturing	+2	+5	+5
Testing by User	+1	0	-5
Net Change from Base Case	**-22%**	**0%**	**-20%**
DEVELOPMENT BLOCK 2			
Requirements	+1	0	0
Develop Techn.	-16	0	-5
Advanced Dev.	-17	0	-5
Manufacturing	+2	+10	+10
Testing by User	+1	0	0
Net Change from Base Case	**+29%**	**+10%**	**0%**
DEVELOPMENT BLOCK 3			
Requirements	+1	0	0
Develop Techn.	-16	0	0
Advanced Dev.	-17	+5	0
Manufacturing	+2	+15	+15
Testing by User	+1	0	+5
Net Change from Base Case	**+29**	**+20**	**+20**

Estimated Changes in Evolutionary Acquisition Processes

to Reflect Open Systems

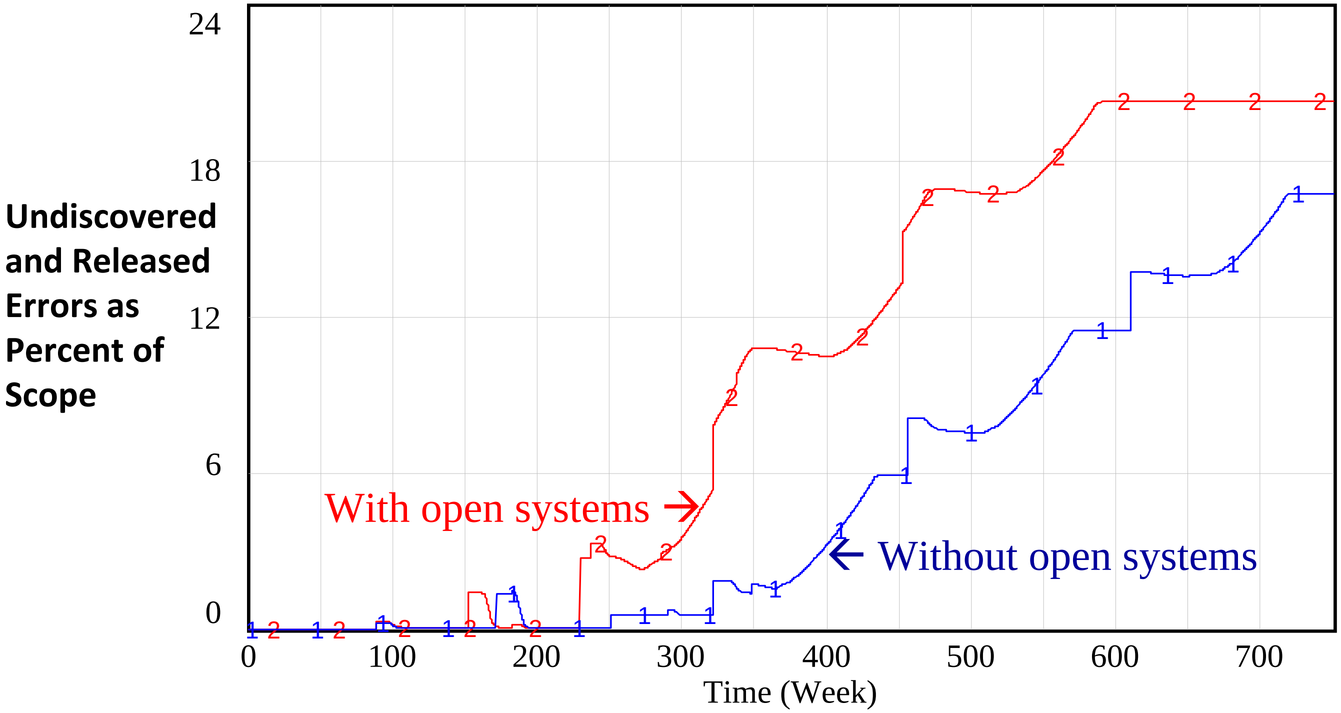
Simulation Results: Cycle Time and Cost



Open systems and Evolutionary Acquisition can reduce cycle time and time-based costs due to less and faster component design.

Requirement Fulfillment with Evolutionary Acquisition Without and With Open Systems

Simulation Results: Hidden Errors

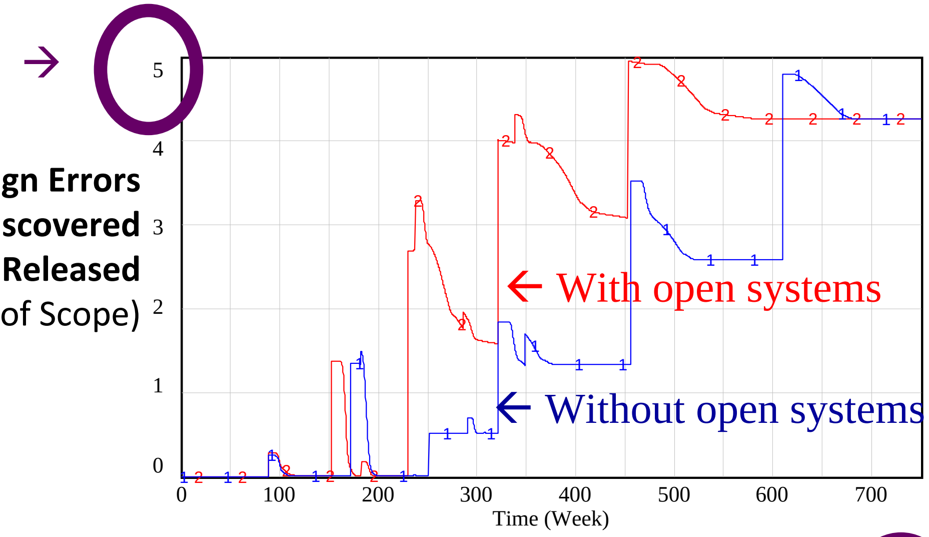


Open systems appear to generate more errors that are undiscovered and released – what types of errors?

Undiscovered Problems in Evolutionary Acquisition Without and With Open Systems

Simulation Results:

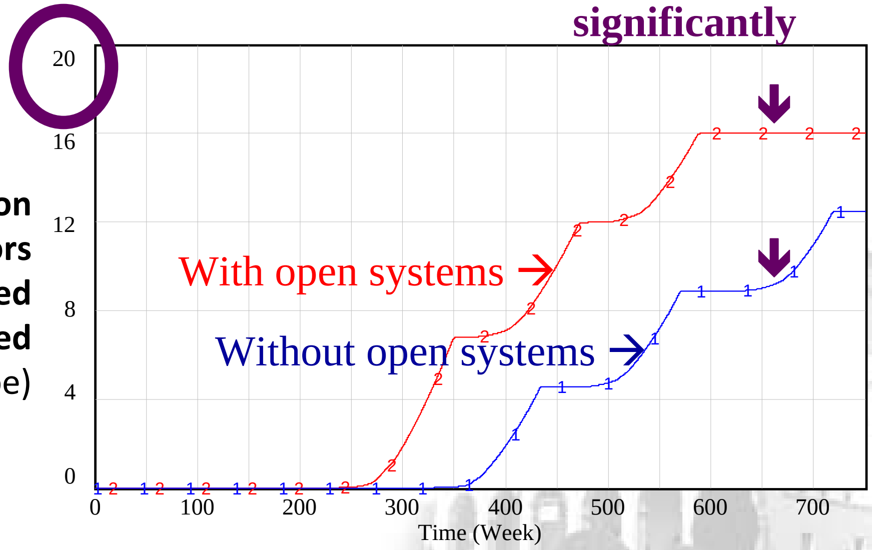
Undiscovered and Released Errors - Design vs. Integration



← 1) Design errors are about the same with and without open systems

3) Open systems can increase the number of integration errors significantly

2) There are 3-5 times more integration errors than design errors →



Open systems shifts some program management from design challenges (earlier, manifest) to standards and integration challenges (later, latent)

Implications for Practice (1 of 3)

Different Types and Amounts of Risks and Skills

- Shift in focus from design to standards and integration impacts the types and amounts of risk that programs accept and must manage.
- Open systems reduce design risks
- Open systems creates standards selection and standards change risks
- Different types of skills are needed to manage different types of risk…less detailed technical expertise will likely be needed and more integration and systems

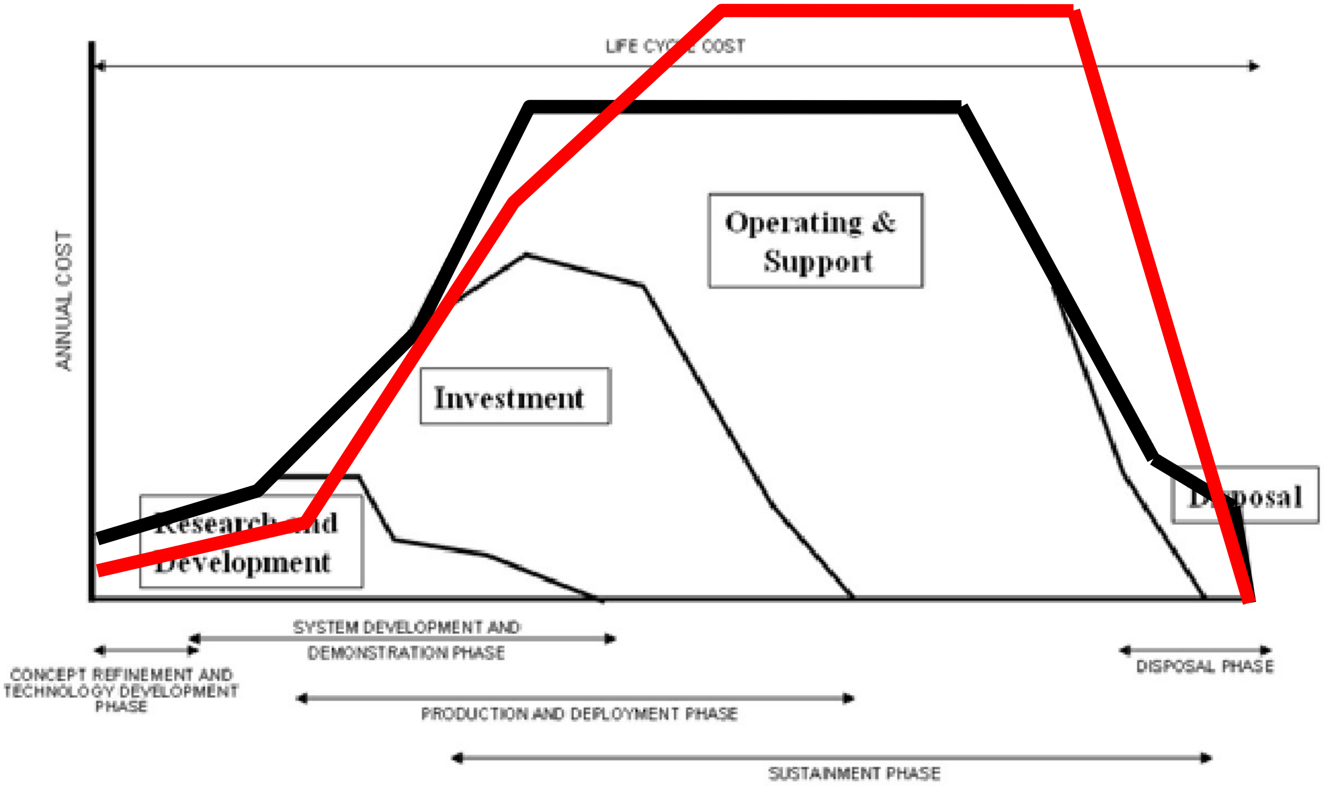
  Ex.: Detailed component design risk management requires technical expertise for design review and component testing, but integration risk management requires a broader systems understanding of the product, and how subsystems work together to fulfill requirements.

Integrating open systems and evolutionary acquisition, which repeats the development process over multiple blocks, will require significant, extended need for integration and systems expertise within acquisition programs.

Implication for Practice (2 of 3)

A Temporal Shift in Program Risks and Potential Costs

The interaction of open systems and evolutionary acquisition may reduce R&D costs but increase and delay integration costs



Relative Costs during a Product Life Cycle
(based on Defense Acquisition Guidebook, Nov. 2004, p. 43)

Implications for Practice (3 of 3)

Trading Design Obsolescence for Integration Obsolescence?

- Traditional acquisition processes commit programs to customized designs and therefore bear significant ***design obsolescence risk*** when threats and technologies evolve away from the design.
- Using open systems requires a program to commit to one or more standards early in a program and therefore bear significant ***standards obsolescence risk*** if and as standards evolve away from the needs of the program and integration problems increase.

Adding open systems to evolutionary acquisition may cause programs to trade away design risk for increased integration risk.

Closing

- **Open systems and evolutionary acquisition *<u>can</u>* interact synergistically. But, program managers must:**
  - Design programs to capture specific benefits
  - Design programs to manage (different) risks
- **Future work**
  - Test lessons in active acquisition programs
  - Learn from experience of multiple programs
  - Extend lessons into additional implications for practice and recommendations for management