This DoD Systems Engineering Research Center (SERC) briefing was commissioned as a quick-look study by Ms. Kristen Baldwin, Director of Systems Analysis within the ODDR&E Systems Engineering (SE) organization and Deputy Director of the SERC. The recent DoD Instruction 5000.02 and the Congressional Weapon System Acquisition Reform Act (WSARA) have recommended evolutionary acquisition as the preferred strategy for Major Defense Acquisition Programs (MDAPs). Since DoD SE has largely been performed under non-evolutionary acquisition practices, there are questions about what needs to be changed about DoD SE practices and SE-related acquisition management practices to enable SE to function more effectively. The study is presented as a briefing with notes to emphasize that it is not an exhaustive study, but one to set the context for more detailed analyses and initiatives.

The study found that there are several forms of evolutionary acquisition, and that there is no one-size-fits-all SE approach that is best for all situations. For rapid-fielding situations, an easiest-first, get something working, evolutionary SE approach is best. But for enduring systems, an easiest-first evolutionary SE approach is likely to produce an unscalable system whose architecture is incompatible with achieving high levels of safety and security. The study also found that evolutionary acquisition requires much higher sustained levels of SE effort, earlier and continuous integration and test, pro-active approaches to address sources of system change, greater levels of concurrent engineering, and achievement reviews based on evidence of feasibility vs. evidence of plans, activity, and system descriptions.

The study also found that many traditional acquisition practices are incompatible with effective SE of evolutionary acquisition. These include assumptions that full-capability requirements can be specified up front along with associated full-capability plans, budgets, schedules, work breakdown structures, and earned-value management targets; that most systems engineers can be dismissed after PDR; and that all forms of requirements change or “creep” should be discouraged. The study also found that other inhibitors to effective SE need to be addressed, such as underbudgeting (SE is the first victim of inadequate budgets); contracting provisions emphasizing functional definition before addressal of key performance parameters; and management temptations to show rapid progress on easy initial increments while deferring the hard parts until later increments.

Based on these findings, the study recommended that significant initiatives be undertaken to provide the necessary EvA infrastructure of SE-related acquisition and development processes, contracting and incentive structures, milestone decision criteria, financing, program management, and staffing, along with research support to address current gaps between EvA infrastructure needs and capabilities. Projects attempting to succeed at EvA while burdened by DoD’s current acquisition infrastructure could lead to sufficient numbers of projects “failing with EvA” to cause policymakers to give up on it before it has a chance to succeed.
**Summary: Evolutionary Acquisition (EvA)**

- EvA is the preferred DoD strategy for MDAPs (DoDI 5000.02)
  - It is more responsive to DoD’s current and future challenges
- There are several forms of EvA
  - With different strengths and shortfalls
- EvA requires significant advances in current practices
  - In systems engineering (SE) and development processes, particularly to avoid unscalable, easiest-first, sunny-day architectural commitments
  - In rapid, adaptive integration of SE and acquisition management (AM)
  - In financial and human resource allocation
  - In workforce capability and empowerment
  - In research and technology priorities and transition speedup
- Initiatives are needed to provide these advances in time for EvA to succeed

The report begins by emphasizing that evolutionary acquisition (EvA) is not only recommended policy via DoD Instruction 5000.02, but also that it is the best approach for most (but not all) DoD systems acquisition situations. It then summarizes the various forms of EvA, along with their primary strength and weakness areas, along with those of the traditional single-step-to-full-capability approach.

EvA has been successfully applied on DoD programs, but it has often been misapplied or unable to overcome current institutional inhibitors to its practice. Most often, the problem programs have proceeded with easiest-first initial increments without sufficient early systems engineering to provide evidence that the early-increment architectural commitments will enable subsequent achievement of full-system scalability, security, safety, and resilience in off-nominal situations. The study summarizes findings in five SE-related EvA areas where significant advances in current institutions and practices are needed to ensure effective SE results. These are:

- In systems engineering (SE) and development processes
- In rapid, adaptive integration of SE and acquisition management (AM)
- In financial and human resource allocation
- In workforce capability and empowerment
- In research and technology priorities and transition speedup.

The study concludes by recommending that significant initiatives be undertaken in these areas to provide the necessary EvA infrastructure of SE-related acquisition and development processes, contracting and incentive structures, milestone decision criteria, financing, program management, and staffing, along with research support to address current gaps between EvA infrastructure needs and capabilities. Otherwise, projects attempting to succeed at EvA will find DoD’s current acquisition infrastructure more of a hindrance than a help, which could lead to sufficient numbers of projects “failing with EvA” to cause policymakers to give up on it before it has a chance to succeed.
What is “Evolutionary Acquisition”?

• An acquisition approach that involves
  – Delivering mission capability in increments
  – Continuing reassessment of future-increment priorities

• It departs from the outdated assumptions underlying traditional acquisition
  – System requirements can be specified in advance
  – System requirements are largely stable
  – Full operational capability cost estimation is feasible
  – Full-development, up-front Integrated Master Plans, Integrated Master Schedules, and Earned Value Management Plans are feasible

• Its successful application requires rethinking traditional systems engineering (SE) and acquisition practices

DoDI 5000.02, December 2008 defines EvA as “the preferred strategy for rapid acquisition of mature technology for the user. An evolutionary approach delivers capability in increments, recognizing, up front, the need for future capability improvements. The objective is to balance needs and available capability with resources, and to put capability into the hands of the user quickly. The success of the strategy depends on phased definition of capability needs and system requirements, and the maturation of technologies that lead to disciplined development and production of systems that provide increasing capability over time.”

The key roles of SE and complementary initiatives such as Competitive Prototyping [Young, 2007] have also been emphasized in the Congressional Weapon System Acquisition Reform Act (WSARA) [WSARA, 2009]. Its Section 201 requires ensuring that “the process for developing requirements is structured to enable incremental, evolutionary, or spiral approaches.” Its Section 203 requires ensuring that “the acquisition strategy for each major defense acquisition program provides for competitive prototypes before Milestone B approval.”

Unlike traditional acquisition approaches, DoDI 5000.02 and WSARA recognize the 21st century realities that most future DoD systems will not be able to specify their full-capability requirements in advance, and thus will not be able to enter into a “total package procurement” agreement to develop such a full capability for a predetermined cost. They also recognize that the pace of change in threats, technology, interoperating systems, world affairs, and information infrastructure will require continual adaptation of DoD systems to rapid change.

Thus, DoD commitment to evolutionary acquisition brings with it a need to rethink many traditional systems engineering and acquisition practices to bring them in concert with 21st century realities.
Most DoD systems and the acquisition processes that developed them were organized to function in set-piece warfare situations, and have not been well-suited to function in current and future asymmetric, irregular, and hybrid warfare situations. Recent warfare experiences in Iraq and Afghanistan, and large-system response-time data shown on the next chart, illustrate this challenge to DoD SE and EvA.

In the early stages of the Iraq war, DoD forces were phenomenally successful. They could pick the time and place of attacks, and their highly superior command, control, communications, computing, intelligence, surveillance, and reconnaissance (C4ISR) capabilities enabled them to perform observe-orient-decide-act (OODA) loops well inside those of their conventional Iraqi army adversaries.

However, in the later stages of the Iraq war, and in Afghanistan, DoD faced much more serious challenges that are representative of many future conflict situations. Their adversaries were diffuse. They could pick the time and place of an attack. They had little to lose, and could use very agile, lightweight systems and processes, while being able to use powerful but hazardous hardware and software at relatively little risk.

On the other hand, DoD forces must be ready for anything at any time, and must be able to defend much more valuable assets. This requires more heavyweight systems and processes in order to develop and operate high-assurance systems, and restricts the use of untrusted components. Even for individual systems, this causes significant challenges in turning OODA loops inside those of one’s adversaries, but as seen next, the challenges will be even higher for complex systems of systems.
This chart shows data from two complex systems of systems that illustrates the challenge of executing tight OODA loops that involve coordinating changes across closely coupled systems of systems. The average time in workdays (longer than in calendar days) was 27 workdays to close a change request within an individual platform or capability group; 48 workdays if the change required coordination across multiple platform or capability groups; and 141 workdays if the change involved a change in performers’ contracts. These were typical high-content U.S. build-to-specification contracts, which [Hall, 1976] found averaged 10 times as long as high-context French contracts.

Corroborative data was provided at a workshop by [Schroeder, 2007], who found that the length of such changes was proportional to the number of contracts requiring changes. Other experienced sources have reported even longer contract change turnaround times for large complex systems. Clearly, improvements are needed in change-facilitating architectures, processes, and contracts if DoD is to stand any chance of performing evolutionary acquisition in ways that keep its change-adapting OODA loops within those of its adversaries.
EvA has been successfully applied on DoD programs, but it has often been misapplied or unable to overcome current institutional inhibitors to its practice. Most often, the problem programs have proceeded with easiest-first initial increments without sufficient early systems engineering to provide evidence that the early-increment architectural commitments will enable subsequent achievement of full-system scalability, security, safety, and resilience in off-nominal situations. The study summarizes findings in five SE-related EvA areas where significant advances in current institutions and practices are needed to ensure effective SE results. These are:

- In systems engineering (SE) and development processes
- In rapid, adaptive integration of SE and acquisition management (AM)
- In financial and human resource allocation
- In workforce capability and empowerment
- In research and technology priorities and transition speedup.

Findings in each area are described and illustrated by representative examples of the needs for improved SE-related EvA capabilities.
There is No One-Size-Fits-All EvA Model

<table>
<thead>
<tr>
<th>Type</th>
<th>Examples</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prespecified Sequential</td>
<td>Platform base plus PPPIs</td>
<td>Prespecifiable full-capability requirements, scalability when stable</td>
<td>Emergent requirements or rapid change, architecture breakers</td>
</tr>
<tr>
<td>Evolutionary Sequential</td>
<td>Small: Agile larger: Rapid fielding</td>
<td>Adaptability to change, need for usage feedback</td>
<td>Easiest-first; late, costly fixes; SysE time gaps</td>
</tr>
<tr>
<td>Evolutionary Overlapped</td>
<td>Stable development; Maturing technology</td>
<td>Mature technology upgrades</td>
<td>Emergent requirements or rapid change; SysE time gaps</td>
</tr>
<tr>
<td>Evolutionary Concurrent</td>
<td>Rapid, emergent development Systems of systems</td>
<td>Emergent requirements or rapid change, SysE continuity</td>
<td>Overkill on small or highly stable systems</td>
</tr>
</tbody>
</table>

Time phasing terms: Scoping; Architecting; Developing; Producing; Operating (SADPO)

Prespecified Sequential: SA; DPO1; DPO2; DPO3; ...
Evolutionary Sequential: SADPO1; SADPO2; SADPO3; ...
Evolutionary Overlapped: SADPO1; SADPO2; SADPO3; ...
Evolutionary Concurrent: SA; D1; PO1...
SA2; D2; PO2...
SA3; D3; PO3 ...

The Incremental and Evolutionary Acquisition models shown above are represented in terms of the various ways they sequence the primary system development activities: system scoping (determining the system boundaries), system architecting (concurrently defining the system’s requirements, architecture, and life cycle plans), and system development, production, and operation.

The Prespecified Sequential incremental development model is not evolutionary. It just splits up the development in order to field an early Initial Operational Capability, followed by several pre-planned product improvements (P3Is). When requirements are prespecifiable and stable, it enables a strong, predictable process. When requirements are emergent and/or rapidly changing, it often requires very expensive rework when it needs to undo architectural commitments.

The Evolutionary Sequential model rapidly develops an initial operational capability and upgrades it based on operational experience. Pure agile software development fits this model: if something is wrong, it will be fixed in 30 days in the next release. Rapid fielding also fits this model for larger or hardware-software systems. Its strength is getting quick-response capabilities in the field. For pure agile, it can fall prey to an easiest-first set of architectural commitments which break when, for example, it tries to add security as a new feature in a later increment. For rapid fielding, it may be expensive to keep the SE and development teams together while waiting for usage feedback, but it may be worth it.

Evolutionary Overlapped is the form chosen to represent EvA in DoDI 5000.02. It covers the special case of deferring the next increment until the desired new technology is mature enough to be added. In practice, DoDI 5000.02 allows any of the three types of EvA, but as seen in Chart 9, the figure can easily be misinterpreted as the single recommended process for EvA.

Evolutionary Concurrent has the systems engineers handling the change traffic and rebaselining the plans and specifications for the next increment, while keeping the development stabilized for the current increment. Its example and pros and cons are provided in chart 10.
This decision table provides criteria for deciding which of the four classes of incremental and evolutionary acquisition (EvA) defined in Chart 7 to use, plus the choice of non-incremental, single-step development.

The Single-Step-to-Full-Capability process exemplified by the traditional waterfall or sequential Vee model is appropriate if the product’s requirements are prespecifiable and have a low probability of significant change; and if there is no value in or opportunity to deliver a partial product capability. A good example would be the hardware portion of a geosynchronous satellite.

The Prespecified Sequential process is best if the product’s requirements are prespecifiable and have a low probability of significant change; and if waiting for the full system to be developed incurs a loss of important and deliverable incremental mission capabilities. A good example would be a well-understood and well-prioritized sequence of software upgrades to a programmable radio.

The Evolutionary Sequential process is best when there is a need to get operational feedback on a quick-response capability before defining and developing the next increment’s content. A good example is the need to save lives by rapidly fielding a capability to neutralize improvised explosive devices (IEDs).

The Evolutionary Overlapped process is best when one does not need to wait for operational feedback, but may need to wait for next-increment enablers such as technology maturity, external system capabilities, or needed resources. A good example is the need to wait for technology that is maturing via advanced technology demonstrations.

The Evolutionary Concurrent process is best when the enablers are available, but there is a great deal of change traffic to be handled that would destabilize the team developing the current increment. Examples may be new emerging threats, emergent user capability needs, external system interface changes, technology matured on other programs, or COTS upgrades.
Evolutionary acquisition is the preferred DoD strategy for rapid acquisition of mature technology for the user. An evolutionary approach delivers capability in increments, recognizing, up front, the need for future capability improvements. The objective is to balance needs and available capability with resources, and to put capability into the hands of the user quickly. The success of the strategy depends on phased definition of capability needs and system requirements, and the maturation of technologies that lead to disciplined development and production of systems that provide increasing capability over time. (See Figure 2).

Evolutionary acquisition requires collaboration among the user, tester, and developer. In this process, a needed operational capability is met over time by developing several increments, each dependent on available mature technology. Technology development preceding initiation of an increment shall continue until the required level of maturity is achieved, and prototypes of the system or key system elements are produced. Successive Technology Development Phases may be necessary to mature technology for multiple development increments.

Each increment is a militarily useful and supportable operational capability that can be developed, produced, deployed, and sustained. Each increment will have its own set of threshold and objective values set by the user. Block upgrades, pre-planned product improvement, and similar efforts that provide a significant increase in operational capability and meet an acquisition category threshold specified in this document shall be managed as separate increments under this Instruction.

In practice, DoDI 5000.02 allows any of the three types of EvA, but as seen above, the figure can easily be misinterpreted as the single recommended process for EvA. It should not be interpreted as a one-size-fits-all approach to DoD evolutionary acquisition, but only as the form that requires programs to defer the start of a new development phase until its necessary technology capabilities have been shown to be mature.
The frequent need to deliver high-assurance incremental capabilities on short fixed schedules means that each increment needs to be kept as stable as possible. This is particularly the case for large, complex systems, in which a high level of rebaselining traffic during development can easily lead to chaos. In keeping with the use of the spiral model as a risk-driven process model generator, the risks of destabilizing the development process make this center portion of the project into a stabilized, waterfall-like build-to-specification process, with unforeseeable change traffic routed elsewhere. The need for high assurance of each increment also makes it cost-effective to invest in a team of appropriately skilled personnel to continuously verify and validate the increment as it is being developed. (The dotted lines represent achievement strategies; the solid lines represent flows of data and resources.)

However, “routing the change traffic elsewhere” does not imply deferring its change impact analysis, change negotiation, and rebaselining until the beginning of the next increment. With a single development team and rapid rates of change, this would require a team optimized to develop to stable plans and specifications to spend much of the next increment’s scarce calendar time performing tasks much better suited to agile teams.

The appropriate metaphor for addressing rapid change is not a build-to-specification metaphor or a purchasing-agent metaphor but an adaptive “command-control-intelligence-surveillance-reconnaissance” (C2ISR) metaphor. It involves an agile team performing the first three activities of the C2ISR “Observe, Orient, Decide, Act” (OODA) loop for the next increments, while the plan-driven development team is performing the “Act” activity for the current increment. “Observing” involves monitoring changes in relevant technology and COTS products, in the competitive marketplace, in external interoperating systems and in the environment; and monitoring progress on the current increment to identify slowdowns and likely scope deferrals. “Orienting” involves performing change impact analyses, risk analyses, and tradeoff analyses to assess candidate rebaselining options for the upcoming increments. “Deciding” involves stakeholder renegotiation of the content of upcoming increments, architecture rebaselining, and the degree of COTS upgrading to be done to prepare for the next increment. It also involves updating the future increments’ Feasibility Rationales to ensure that their renegotiated scopes and solutions can be achieved within their budgets and schedules.

A successful rebaseline means that the plan-driven development team can hit the ground running at the beginning of the “Act” phase of developing the next increment, and the agile team can hit the ground running on rebaselining definitions of the increments beyond. Further details are provided in the [Boehm-Lane, 2009] discussion of Stage II of the Incremental Commitment Model.
“Evolution” and “Maintenance”

- EvA often includes maintenance of fielded increments
  - Changes to fielded increments must be coordinated with development of new increments
    - Even if a separate organization handles maintenance
- As above, there is no one-size-fits-all process for this
  - Can involve an evolutionary next-increment with a Milestone B gate
    - As in USD(AT&L) Designation of MDAP Subprograms memo [Carter, 2009]
  - Can also involve pools of maintainers performing low levels of as-needed maintenance not requiring Milestone Bs
- Can’t have predetermined process for change traffic
  - Best available process shown in next chart
- Situation even more complex with systems of systems
  - May need to integrate all types of evolution processes

Once an initial version of a system is deployed, it enters the “operation and maintenance” stage and must be maintained. Maintenance activities include defect fixing, periodic upgrades to embedded COTS products, technology refresh, high-priority user requests, and changes to accommodate external interface changes initiated by other systems.

As with evolutionary development, there are no one-size-fits-all modes of operations and maintenance (O&M). Major Defense Acquisition Programs (MDAPs) must undergo a MS B before committing to their next increment per the [Carter, 2009] memo. Also, however, there are some steady-state collections of systems in O&M (e.g., at Army, Navy, and Air Force logistics centers) with low-level but unpredictable change traffic for which a pool of O&inters just handle the prioritized backlog as time is available, and for which no formal milestones per O&Med system are necessary or useful. The next chart shows that this is also the case for the increment startup of the agile rebaselining team whose operation was described in the previous chart. This team usually needs to cope with unpredictable and asynchronous change requests such as security breaches, new terrorist threats, COTS and/or externally evolving system change or decommitment surprises, new technology opportunities, or leadership priority changes during the next increment. Given the unpredictability of the nature and magnitude of these change requests, there is no way to have them go through a MS A covering a predictable set of upcoming activities. About the best way that one can cope with such change traffic is to perform a continuing triage to perform either a quick fix, a plan for including the change request in the next increment, or a deferral or rejection, as shown in the next chart. On the other hand, a P3I for a major new-technology upgrade (vs. a straightforward COTS-upgrade improvement) should go through a MS A and B, with a PDR before MS B as per 5000.02 and Chart 9.

The situation becomes even more complex with systems of systems, which may have all types of evolutionary acquisition going on within their constituent systems.
To handle unpredictable, asynchronous change traffic, it is not possible to do a sequential observe, then orient, then decide process. Instead, it requires handling unforeseen change requests involving new technology or COTS opportunities; changing requirements and priorities; changing external interfaces; low-priority current increment features being deferred to a later increment; and user requests based on experience with currently-fielded increments (including defect fixes).

In the context of the agile rebaselining team in Chart 10, this chart shows how the agile team interacts with the change proposers, current-increment development team, and managers of future increments to evaluate proposed changes and their interactions with each other, and to negotiate rebaselined Milestone B packages for the next increment. Again, there is no precise way to forecast the budget and schedule of this team’s workload. Within an available budget and schedule, the agile team will perform a continuing “handle now; incorporate in next increment rebaseline; defer-or-drop” triage in the top “Assess Changes, Propose Handling” box. Surges in demand would have to be met by surges in needed expertise and funding support.
The relatively clean and simple processes in charts 9 and 10 are challenged when dealing with the realities of complex, multi-mission, multi-stakeholder systems of systems (SoS). The complexities are clear even for the relatively straightforward “Directed” SoS form [DoD, 2008a], in which a lead organization “controls” the overall budget, supplier selection, and SoS architecture framework. However, this control is often undermined by independently evolving interfaces with success-critical external systems, COTS products, or net-centric services; and by obligations of suppliers who are also parts of other SoSs.

For the more typical “acknowledged” SoS [DoD, 2008a], many of the constituent systems are already operating and will feed incremental upgrade information into the SoS SE analysis process. The SoS system engineers are continually tracking changes in the constituent systems (some of which are initiated by other SoS in which the constituent system participates) while trying to guide the development of new SoS capabilities and enhancement to the overall architecture and performance of the SoS. As the figure shows, seldom are the constituent system upgrade cycles well-aligned and this means that it is difficult to test new capabilities until the last constituent system contributing to the capability completes its upgrade. As a result, many of the capabilities “pieces” are deployed before the capability is tested. Add to this complexity the incorporation of new systems currently under development. In addition, it is not atypical for a new system to be identified as a constituent for multiple SoSs.
EvA is the preferred DoD strategy for MDAPs (DoDI 5000.02)
  – It is more responsive to DoD’s current and future challenges

There are several forms of EvA
  – With different strengths and shortfalls

EvA requires significant advances in current practices
  – In systems engineering (SE) and development processes, particularly to avoid unscalable, easiest-first, sunny-day architectural commitments
  – In rapid, adaptive integration of SE and acquisition management (AM)
  – In financial and human resource allocation
  – In workforce capability and empowerment
  – In research and technology priorities and transition speedup

Initiatives are needed to provide these advances in time for EvA to succeed

EvA has been successfully applied on DoD programs, but it has often been misapplied or unable to overcome current institutional inhibitors to its practice. Most often, the problem programs have proceeded with easiest-first initial increments without sufficient early systems engineering to provide evidence that the early-increment architectural commitments will enable subsequent achievement of full-system scalability, security, safety, and resilience in off-nominal situations. The study summarizes findings in five SE-related EvA areas where significant advances in current institutions and practices are needed to ensure effective SE results. These are:

  In systems engineering (SE) and development processes
  In rapid, adaptive integration of SE and acquisition management (AM)
  In financial and human resource allocation
  In workforce capability and empowerment
  In research and technology priorities and transition speedup.

Findings in each area are described and illustrated by representative examples of the needs for improved SE-related EvA capabilities.
EvA Differences:
SE and Development Processes

• EvA not a “one-size-fits-all” process
  – As shown in charts above
• No clear boundary between “development,” “operations and maintenance”
  – Don’t release your key SE people at PDR
• Short prioritized increments; Schedule as independent variable (SAIV)
  – Architect to add or drop borderline-priority features to meet schedule
  – Exceptions for non-subsettable deliverables
• Earlier testing, verification, and validation (V&V)
  – Architecture, infrastructure V&V to avoid unscalable easiest-first increments
  – Earlier/continuous test, V&V
• Rapid, concurrent, pro-active SE and acquisition management
  – Of product and process; requirements and solutions; development and evolution;
    hardware/software/human factors
  – Proactive vs. reactive with respect to technology, NDI, external interfaces
• Evidence/risk as decision criterion, first class deliverable
  – Process for evidence preparation
  – Risk/opportunity-driven: avoid overkill, underkill (balance risk, opportunity)
  – Competitive prototyping as a way of buying information to reduce risk

As seen in Chart 7, EvA is not a “one-size-fits-all” process, but has several forms, each with different strengths and weaknesses. This implies the need to choose among the candidate forms based on top-level criteria such as the ones in Chart 8. However, the realities of complex systems shown in Chart 13 further imply that hybrid and evolving mixes of the candidate forms are increasingly likely to be needed. Thus, SE has an expanded role in evolving its own processes, as well as those for development and evolution.

As warfighting strategies have come to accept that often there is no Forward Edge of the Battle Area (FEBA) separating friendly and adversary forces, acquisition strategies need to adapt to the fact that with EvA, there will be no clear line separating Development (and its funding) from Operations and Maintenance (and its funding), and that many systems will continually evolve rather than being handed to separate maintenance organizations directly after initial fielding. The traditional up-front nature of SE, which has the SEs departing the project soon after its Preliminary Design Review (PDR), will no longer be sustainable.

The accelerating pace of change, along with the need for high assurance levels, also implies that the evolutionary increments need to be as short and stable as possible, leading to evolutionary overlapped processes such as in Chart 10. Stability in a world of systems being parts of multiple systems of systems also means that keeping to synchronized schedules and a schedule-as-independent-variable (SAIV) approach become increasingly important in keeping capabilities synchronized and stabilized. In some cases, this will involve synchronizing a longer hardware increment with the result of several shorter software increments.

For the overall system, early system-level V&V needs to ensure that adequate SE has been performed to avoid unscalable easiest-first architectural commitments. Within each increment, V&V needs to find defects as close as possible to the insertion point so that they can be quickly fixed and not cause extensive late rework, along with earlier test planning, preparation, and execution.

The pace of change in threats and solution technology implies that projects no longer have the luxury of proceeding sequentially through requirements and solutions levels. The risks of fixing on infeasible or obsolete requirements or solutions implies that SE, and its enabling acquisition management processes, must become much more rapid, concurrent and proactive vs. reactive, and must use risk (and opportunity) as primary decision criteria for determining how a process should proceed. Risk is proportional to the lack of evidence of a decision option’s feasibility and appropriateness before committing to it, further implying the need for sufficient SE budget, schedule, and qualified personnel to produce the evidence. Charts 17-19 elaborate on rapid acquisition management and evidence-based milestone reviews. Charts 20-21 provide an example of competitive prototyping as a way of reducing risk.
The SAIV* Process Model


1. Shared vision and expectations management
2. Feature prioritization
3. Schedule range estimation and core-capability determination
   - Top-priority features achievable within fixed schedule with 90% confidence
4. Architecting for ease of adding or dropping borderline-priority features
   - And for accommodating post-IOC directions of growth
5. Incremental development
   - Core capability as increment 1
6. Change and progress monitoring and control
   - Add or drop borderline-priority features to meet schedule

*Schedule As Independent Variable; Feature set as dependent variable
– Also works for cost, schedule/cost/quality as independent variable
  – Also known as timeboxing, time-determined development, time-certain development

Best commercial practices (e.g., the Microsoft scalability approach in [Cusumano-Selby, 1995]) recognize that cost and schedule cannot be estimated precisely, and that buffers can be used to synchronize and stabilize multiple concurrent developments. The SAIV process, described in detail in CrossTalk [Boehm et al, 2002], is a scaled-up version of the timeboxing approach used in agile methods that can be used on government procurements to use dropping of lower-priority features as a way of providing such buffers. Versions of it have been recommended in some recent DoD studies in such terms as time-certain development or time-determined development. It applies when there is a strong need to produce incremental capabilities by well-established dates, but as with evolutionary development, the exact nature of the capabilities is not well specifiable in advance.

The SAIV process operates as follows:

- Work with stakeholders in advance to achieve a shared product vision, realistic expectations, and prioritized requirements;
- Estimate the maximum capability buildable with high confidence within the available schedule;
- Define a core-capability increment content based on priorities, end-to-end usability, and need for early development of central high-risk infrastructure;
- Architect the system for ease of dropping or adding borderline-priority features;
- Monitor progress; add or drop features to accommodate high-priority changes or to meet schedule.

Versions of the SAIV process have been used successfully to date on numerous commercial projects needing to meet highly leveraged introduction dates such as product rollouts, trade show demonstrations, and events such as the Olympic Games. An example of its use on one recent very large evolutionary-development DoD system involved dropping lower-priority requirements originally within the IOC set such as automatic real-time natural language translation.
Rapid, Adaptive Acquisition Management

- Need to rework traditional guidance documents
  - No longer prespecified, total-package, sequential IMP, IMS, WBS, EVMS
  - Concurrently engineered with evidence-based decision milestones

- No “one-size-fits-all” contracting
  - E.g., for build-to-spec, agile rebaselining, V&V teams
  - SE budgets a function of system size, criticality, requirements volatility
  - Award fees rewarding collaborative, effective change adaptation

- Continuous monitoring: INCOSE leading indicators; SERC effectiveness measures, or equivalent
  - Evidence preparation; schedule; exit criteria
  - Success-critical stakeholders participation; validation by experts

- Competitive prototyping critical success factors
- Acquisition corps empowerment
  - Next generation of tools, education, and training; career paths

Many traditional acquisition practices are incompatible with effective SE of evolutionary acquisition. They include assumptions that full-capability requirements can be specified up front along with associated full-capability plans, budgets, schedules, work breakdown structures, and earned-value management targets; that most systems engineers can be dismissed after PDR; and that requirements change or “creep” should be discouraged. Examples include total-package fixed-price, fixed schedule build-to-specification contract instruments, including associated contract exhibits for sequential functions-first reviews and audits, and fully prespecified integrated master plans, schedules, work breakdown structures, and earned value management plans.

EvA needs to cope with rapid change while preserving high assurance. One successful approach for doing this involves the multi-mode, concurrent-engineering processes illustrated in charts 10 and 12. These are incompatible with traditional one-size-fits-all, fixed-price, build-to-spec contracts. In chart 10, the traditional contract approach is appropriate for the stabilized, build-to-spec development team, but not for the agile rebaselining and rapid V&V teams working with the build-to-spec teams. Nor are there fixed budget percentages for these two teams, as discussed partly in chart 12 and in charts 22-26 under Finance.

For evidence of feasibility to succeed as a milestone decision criterion and as a way to synchronize and stabilize concurrent engineering of requirements and solutions, the evidence needs to become a first-class deliverable instead of an optional appendix. As such, its production needs to be planned and budgeted for, and progress vs. plans continuously monitored via earned value management, leading indicator tracking [INCOSE, 2007], and expert validation of evidence preparation adequacy and cost-effectiveness [Boehm et al., 2009].

Competitive prototyping (CP), as recommended in [Young, 2007] and [WSARA, 2009], can be a powerful way of stimulating innovation and generating evidence of feasibility. But similarly, it needs complementary efforts to prepare for prototype evaluation via testbeds, representative test scenarios, instrumentation, data analysis, and expert reviewers. Other CP critical success factors identified in a 2008 survey [Ingold, 2008] included continuity of funding to keep competitor teams productively on-board during evaluation and downselect periods, use of evidence of preparation and ability to succeed in development as well as prototype capability as downselect criteria, and exit ramps should none of the prototypes demonstrate sufficient maturity to proceed.

Many of these capabilities do not exist, leaving the acquisition corps to struggle to succeed with obsolete traditional acquisition instruments, or with a lack of education and training to determine when and how to use a variety of instruments instead of a one-size-fits-all approach. Initiatives are also needed to address these challenges.
Types of Milestone Reviews

- **Schedule-based reviews (contract-driven)**
  - We’ll hold the PDR on April 1 whether we have a design or not
  - High probability of proceeding into a Death March

- **Event-based reviews (artifact-driven)**
  - The design will be done by June 1, so we’ll have the review then
  - Large “Death by PowerPoint and UML” event
    - Hard to avoid proceeding with many unresolved risks and interfaces

- **Evidence-based commitment reviews (risk-driven)**
  - Evidence provided in Feasibility Evidence Description (FED)
    - A first-class deliverable
    - Based on concurrently engineered ConOps, specs, and plans
  - Shortfalls in evidence are uncertainties and risks
  - Should be covered by risk mitigation plans
  - Stakeholders decide to commit based on risks of going forward

A major recent step forward in the management of outsourced government projects has been to move from schedule-based reviews to event-based reviews. A schedule-based review says basically that, “The contract specifies that the Preliminary Design Review (PDR) will be held on April 1, 2011, whether we have a design or not.”

In general, neither the customer nor the developer wants to fail the PDR, so the project goes into development with the blessing of having passed a PDR, but with numerous undefined interfaces and unresolved risks. As we will show below, these will generally result in major project overruns and incomplete deliveries.

An event-based review says, “Once we have a preliminary design, we will hold the PDR.” Such a review will generally consist of exhaustive presentations of sunny-day PowerPoint charts and UML diagrams, and focus on product-oriented deliverables and reviews, rather than addressing the tradeoffs among key performance parameters (KPPs), where most of the risks are.

As we will see in charts 24 and 25, this lack of risk resolution is the cause of large overruns on large projects. Most contract DIDs cover function, interface, and infrastructure considerations, but place demonstration of their KPP feasibility in optional appendices where they are the first to go when time and effort are running out. As discussed above, this often leads to easiest-first, unscalable, unsecurable architectural commitments in the early increments of an evolutionary development.

Evidence-based milestone reviews require evidence of feasibility to be produced by the developer and reviewed by independent experts. The evidence is thus a first-class deliverable, and can be used to determine whether the technology is sufficiently mature, and whether the plans and architecture are feasible enough to proceed. They have been used successfully in commercial projects for over 20 years [Maranzano, 2005]
Content of Evidence-Based Reviews

- **Evidence** provided by developer and **validated by independent experts** that:

  If the system is built to the specified architecture, it will
  - Satisfy the specified operational concept and requirements
    - Capability, interfaces, level of service, and evolution
  - Be buildable within the budgets and schedules in the plan
  - Generate a viable return on investment
  - Generate satisfactory outcomes for all of the success-critical stakeholders

- Shortfalls in evidence are uncertainties and risks
  - Should be resolved or covered by risk management plans

- Assessed in increasing detail at major anchor point milestones
  - Serves as basis for stakeholders’ commitment to proceed
  - Serves to synchronize and stabilize concurrently engineered elements

  *Can be used to strengthen current schedule- or event-based reviews*

Evidence-based reviews focus on the content of a first-class deliverable often called a Feasibility Evidence Description (FED). The FED contains **evidence** provided by the developer and validated by independent experts that, **if the system is built to the specified architecture it will:**

- Satisfy the specified operational concept and requirements, including capability, interfaces, level of service, and evolution
- Be buildable within the budgets and schedules in the plan
- Generate a viable return on investment
- Generate satisfactory outcomes for all of the success-critical stakeholders
- Identify shortfalls in evidence as risks, and cover them with risk mitigation plans

A FED does not assess a single sequentially developed system definition element, but the consistency, compatibility, and feasibility of several concurrently-engineered elements. To make this concurrency work, a set of anchor point (AP) milestone reviews are performed to ensure that the many concurrent activities are synchronized, stabilized, and risk-assessed at the end of each phase. Each of these AP reviews is focused on developer-produced and expert-validated **evidence**, documented in the FED, to help the success-critical stakeholders determine whether to proceed into the next level of commitment.

The evolving FED and its review at major DoD acquisition milestones can also be added straightforwardly to traditional reviews such as a Preliminary Design Review.
An example to illustrate the benefits of competitive prototyping and incremental commitment approaches to system definition is the Unmanned Aerial Vehicle (UAV) (or Remotely Piloted Vehicles (RPV)) system enhancement discussed in Chapter 5 of the National Research Council’s Human-System Integration report [Pew and Mavor, 2007]. The RPVs are airplanes, helicopters, or other vehicles operated remotely by humans. These systems are designed to keep humans out of harm’s way. However, most current RPV systems are human-intensive, requiring two people to operate a single vehicle.

As RPVs have become increasingly cost-effective, the demand for them has outrun the supply of experienced RPV operators. This has created a strong desire to modify the 2:1 (2 people to one vehicle) ratio to allow for a single operator for 4 aircraft (e.g., a 1:4 ratio). If a proof-of-principle agent-based prototype demo shows a 1:4 performance for some RPV tasks, how should one proceed with acquiring such an agent-based capability for field use?
Total vs. Incremental Commitment -- 4:1 RPV

This slide outlines two approaches to the RPV question: total (non-evolutionary) commitment and incremental (evolutionary) commitment. While this is a hypothetical case for developing a solution to the RPV manning problem, it shows how a premature total commitment without significant modeling, analysis, and feasibility assessment will often lead to large overruns in costs and schedule, and a vehicle-to-pilot ratio that is considerably less than initially desired.

However, by “buying information” early via increasing incremental commitments to competitive prototyping (CP) and validating high-risk elements, the more technologically viable options are evolutionarily explored and identified much earlier, and can be provided at a much lower cost and much closer to the desired date. The incremental commitment approach leads to the same improved vehicle-to-pilot ratio as the total commitment approach, but sooner and at a much reduced cost.

The approach also employs a competitive downselect strategy, which both reduces risk and enables an evidence-based buildup of trust among the acquirers, developers, and users. It is also important to note that not all of the funds go to prototype development. Some must go to technical evaluation of the prototypes, including preparation of testbeds and representative scenarios, and some must go to question-answering and technology studies to keep CP competitor core teams productively together during evaluation and downselect periods.

At $5M per competitor in round 1, $20M each in round 2, and $100M each in round 3, expenditures on downselected competitors are 3*$5M+2*$20M+ $100M= $155M, and expenditures on evaluation are $5M+$15M+$25M=$45M, but the total “extra” CP investment of $200M and 2 months is small compared to the $2 billion and 38 months of savings relative to the total-commitment approach.
Finance, Human Resource Allocation

- Need balance of short-term, long-term budget commitments
  - Budget for long-term continuity but build in short-term off-ramp options
  - E.g., for technology maturity as in the DoDI 5000.02 example of EvA in chart 9
  - Prespecified total-package capabilities, budgets, and schedules generally infeasible
- Higher up-front SE schedules, costs for larger, more critical programs
  - Earlier infrastructure and test plans and preparations
  - Heavier penalties for late rework; generally high payoffs for competitive prototyping
  - Program-specific budgets, schedules, staffing established via mix of parametric models and expert judgment
- Need to address total cost of ownership, lifecycle ROI (DOTMLPF)
  - Lifecycle success-critical stakeholder involvement
  - Mission effectiveness analysis with respect to alternatives;
  - Use to help prioritize increments
- Use as basis for monitoring progress vs. plans
  - Continuing update of resources required

Budgeting for evolutionary acquisition SE (and other activities) needs to be clearly available for the long term, but not guaranteed, as earlier increments may establish that the technology is too immature to be developed and fielded, in which case the right course of action is for the program to take the off ramp, or to defer the next evolutionary increment until the necessary technology is mature, as in the DoDI 5000.02 example of EvA in chart 9. As discussed in chart 3, current and future systems have too many sources of change and emergent requirements to enable total-package up-front budgeting and contracting.

As will be shown in charts 24 and 25, the larger and more critical the program is, the more investment in up-front and continuing SE will be needed. Parametric models such as COSYSMO [Valerdi, 2010] provide ways for programs to determine the effect of program-specific cost drivers on their SE effort needs, but as they are still undergoing full calibration, they should be complemented by expert-judgment estimates. If competitive prototyping is planned, its costs (evaluation preparation, criteria, facilities, bridging competitors during evaluation) need to be determined, but as shown in chart 21, its payoffs generally outweigh the costs.

Full understanding of the life-cycle resource requirements of the program should be addressed, such as sources of cost resulting from new doctrine, organization, training, materiel, leadership, personnel, and facilities (DOTMLPF), plus logistics and continuing SE. The business case for benefits resulting from the costs should be developed; besides providing budget justification, it will help prioritize the backlog of future features. The estimated costs can then be used as a basis of earned-value monitoring of progress with respect to plans. For evolutionary development, as in the [Carter, 2009] memo, this should be done one evolutionary increment at a time.
[Blanchard and Fabrycky, 1998] conclude from their studies of system engineering on numerous projects that, as illustrated in this figure, there are often major decisions and commitments prematurely made in the early stages of a program with respect to things such as technologies, materials and potential sources of supplies and equipment, manufacturing processes, and maintenance approaches. These often encumber the program with architectural and contractual commitments that make changes difficult and expensive to accommodate, once more system-specific knowledge about preferred solutions becomes available. The bottom line result often plays out as a major overrun, not only on DoD projects in GAO reports [GAO, 2008] but also on commercial projects in Standish reports [Johnson, 2006].

This figure can also be used as a roadmap for improving the situation. Strategies such as evolutionary acquisition, competitive prototyping, and evidence-based milestone reviews defer architectural and technology commitments by investing more up-front cost into SE efforts to build up system-specific knowledge and avoidance of unscalable easiest-first EvA commitments. Investments in technology, marketplace, and threat trend analysis provide information about likely directions of change that can be accommodated by more change-adaptive architectures (e.g., [Parnas, 1979; Baldwin-Clark, 2000]) that reduce the steep dropoff in ease of change vs. time.

Ideally, one would like to know more about the shapes of these curves; how they vary with parameters such as system size, criticality, and volatility; and how to reason about them to determine how much SE investment should be made to avoid rework overruns on the not-enough side or analysis paralysis on the too-much side. In general, there is insufficient data to support such quantitative analysis, but fortunately sufficient data has been analyzed to support such decisions for software-intensive systems, as shown in the next two charts.
Insufficient data exist for determining the general return on investment (ROI) of project investments in systems engineering. However, for software-intensive systems, quantitative data from 161 DoD-representative projects show that the ROI of SE investments, a measured by the degree that they have fully resolved the project’s architecture and risk issues by the project’s Preliminary Design Review, is very high for large, critical projects, but relatively low for small, less-critical, volatile projects.

The graph above shows the results of a “how much architecting is enough” analysis, based on the COCOMO II Architecture and Risk Resolution (RESL) factor. This factor was calibrated along with 22 others to 161 project data points. It relates the amount of extra rework effort (the black dashed curves) on a project to the percent of project effort devoted to software-intensive system architecting (the red dotted curve), and to the resulting sum of these costs (the green curves). The analysis indicated that the amount of rework (or project delivery delay at full project staffing) was an exponential function of project size.

A small (10 thousand equivalent source lines of code, or KSLOC) could fairly easily adapt its architecture to rapid change via refactoring or its equivalent, with a rework penalty of 14% between minimal and extremely thorough architecture and risk resolution. However, a very large (10,000 KSLOC) project would incur a corresponding rework penalty of 91%, covering such effort sources as integration rework due to large-component interface incompatibilities and critical performance shortfalls, as often encountered in unscalable easiest-first approaches to EvA. As shown in the chart, this corresponds to a very high initial ROI for reducing total costs by further investments in large-project SE (the slope of the left end of the highest green curve) which decreases to a diminishing-returns “sweet spot.” This analysis was performed for the US Army Future Combat Systems program, and resulted in an additional 18 months being added to the SE schedule [Boehm et al., 2004]. Further details are provided in [Boehm-Valerdi-Honour, 2008]

Actually, the RESL factor includes several other architecture-related attributes besides the amount of architecting schedule investment, such as available personnel capabilities, architecting support tools, and the degree of architectural risks requiring resolution. Also, the analysis assumes that the other COCOMO II cost drivers do not affect the project outcomes. Results for two such factors are shown in the next chart.
The effects of rapid change (volatility) and high assurance (criticality) on the sweet spots are shown in the graph above. Here, the solid black lines are the same as the black, red, and green curves from the average-case cost of rework, architecting, and total cost for the 100-KSLOC project shown in the previous chart. The dotted red lines show the effect on the average-case cost of architecting and total cost if rapid change adds 50% to the schedule of architecture and risk resolution. Quantitatively, this moves the sweet spot from roughly 20% to 10% of effective architecture investment (but actually 15% due to the 50% cost penalty). Thus, high investments in architecture and other documentation have a lower return on investment when rapid change causes high costs of documentation rework for rapid-change adaptation.

The dashed blue lines at the right represent a conservative analysis of the effects of failure cost of architecting shortfalls on the project’s effective business or mission cost-effectiveness sweet spot. It assumes that the costs of architecture shortfalls are not only added rework, but also losses to the organization’s operational effectiveness. These are conservatively assumed to add 50% to the project-rework cost of architecture shortfalls to the organization. In most cases for high-assurance systems, the added cost would be considerably higher.

Quantitatively, this moves the sweet spot from roughly 20% to over 30% as the most cost-effective investment in architecting for a 100-KSLOC project. It is good to note that the “sweet spots” are actually relatively flat “sweet regions” extending 5-10% to the left and right of the “sweet spots.” However, moving to the edges of a sweet region increases the risk of significant losses if some project assumptions turn out to be optimistic.

The bottom line of these two curves is that, subject to an eventual analysis-paralysis investment level, the larger and more critical the project, the higher are the payoffs of investing in SE efforts resulting in architecture and risk resolution. However, the higher the level of requirements volatility and architecture breakage, the lower will be the payoffs in architecture investments.

Again, these results assume that the other factors determining the cost of SE are being held constant. The COSYSMO model described on the next chart helps to determine a more accurate cost of SE for a particular project’s cost driver parameter values.
There are other cost factors to consider in addition to those illustrated on charts 25 and 26. The strongest available capability for estimating their impact on systems engineering costs is the Constructive System Engineering Cost Model (COSYSMO) [Valerdi, 2005]. It has been calibrated to over 60 aerospace-community SE projects. Its size parameters include the numbers of requirements, interfaces, scenarios, and algorithms, each weighted by complexity. Size adjustment parameters include requirements volatility and reuse of SE artifacts, recently added in [Fortune, 2009].

Its application parameters include the project’s levels of Requirements understanding, Architecture understanding, Level of service requirements complexity, Migration complexity, Technology risk, Documentation, Number and diversity of installations/platforms, and Number of recursive levels of design. Its team-oriented parameters include the project’s levels of Stakeholder team cohesion, Personnel experience/continuity, Process capability, Multisite coordination, and Tool support.

Academic versions of the COSYSMO tool are available from MIT and USC; commercial versions are available from Galorath, Price Systems, and Softstar Systems.
DoD is not only challenged by an insufficient number of SEs. As shown in recent presentations such as [Gelosh, 2009], many of the current DoD SEs will retire soon and deprive DoD of their scarce defense-domain knowledge, skills, and abilities (KSAs). Particular needs stemming from the trends toward rapid change and complex, net-centric systems of systems are the need for “T-shaped” SEs with depth in particular specialties along with breadth of skills across such areas as hardware, software, and human factors, and the ability to think strategically at the systems or systems of systems level. Without this expertise, the risks of commitments to unscalable, easiest-first EvA increase significantly. The rapid pace of change also implies the needs to unlearn obsolete practices and learn new ones. As discussed in chart 17, this also applies to an acquisition corps that struggles to succeed with obsolete traditional acquisition instruments, or with a lack of education and training to determine when and how to use a variety of instruments instead of a one-size-fits-all approach.

Both a challenge and an opportunity is the emergence of a next generation of personnel having extensive familiarity with mobile, interactive, social-networked devices and capabilities for dealing with massive amounts of data and for distributed collaboration. Rather than providing these future warfighters, SEs, and acquisition personnel with lock-step, computer-knows-best user interfaces (a good example is WYTINWYG (what you type is not what you get) Microsoft word-processors, which have been changing SEs to Ses as I type this), future systems-engineered user interfaces should empower the future DoD workforce with user-programming and adaptive interfaces (a good example is Microsoft Excel, which empowered many warfighters in Iraq to adapt to new situations more effectively than they could with their prepackaged applications).
EvA Research and Technology Needs

- Incremental vs. start-over tools and methods (e.g., for formal methods)
- Ambiguity-tolerant methods, processes, and tools
- Powerful, flexible, composable models and simulations
- Rapid change-impact analysis
- Scalable methods, processes, and tools
- Concurrent engineering support
  - Collaboration technology; quality attribute tradeoff analysis tools
  - Value-based architecting and design methods, processes, and tools
- Continuous V&V; early, continuous testing; reliable autonomy
- Use of multi-core computing processors for rapid multiple-option analysis
- Integration of hardware, software, and human factors solution approaches and architectures
- Next-generation process maturity models
- Evolutionary acquisition contract and incentive structures
- Rapid technology maturation and adoption

Some analysis tools, such as formal proof techniques, do not work unless everything has been completely specified, and have to start over again if any change is made to the inputs. Such tools do not fit well in an evolutionary-acquisition world of emergent requirements and rapid change. Some emerging methods enable one to break a problem statement up in ways that produce best-effort partial solutions that can be composed with other partial solutions as they become available. However, having tools that can deal with the general problem of whether components satisfying certain properties such as those involving security or safety will still hold when the components are composed is a much-desired but often unachievable capability. A related much-desired but still unsatisfied capability is for scalable change-impact-analysis tools that can rapidly determine which parts of a complex system are affected by a proposed change in the system. In some cases, this can be done for small systems, but does not scale up to complex systems of systems.

Other much-desired capabilities for dealing with rapid change in complex situations involve support for multi-discipline, distributed collaboration support for rapid mutual understanding of complex phenomena and multiple problem and solution views. Particular capabilities are the ability to perform rapid tradeoff and sensitivity analysis of the interactions among key performance parameters such as security, safety, reliability, response time, usability, evolvability, scalability, and affordability. A key opportunity area is value-based architecting and design aids; frequently, technology-based architecting approaches will overfocus on solutions that affect only 2% of the problem rather than searching for solutions with the highest mission-value leverage. Related opportunity areas involve early and continuous V&V and testing, including exploration of potential failure modes and safeguards for autonomous equipment.

Some new opportunities may also be afforded by multi-core computing processors. These are largely being investigated for performance speed, but also provide additional degrees of freedom for concurrently checking assurance assertions, or for concurrently modeling or simulating the prospective outcomes of various proposed courses of action.

Better technology is also needed for dealing with complex systems of systems such as those depicted in chart 13. A particular challenge is to reconcile traditional hardware-oriented functional system decomposition methods with layered and service-oriented software system decompositions, as discussed in charts 29 and 30. Similar challenges involve architecture-based integration of hardware, software, and human factors considerations, as discussed in [Pew and Mavor, 2007]. Other research and technology needs associated with EvA are process maturity models more attuned to agility; contract and incentive structures rewarding team performance and adaptability to change; and capabilities for accelerating technology maturation and adoption, such as testbeds, technology knowledge management capabilities, and rapid prototyping and composition capabilities.
A valuable perspective on the mismatches between traditional hardware-oriented systems engineering architectural structures and modern software architectural structures has been provided in [Maier, 2006]. First, traditional systems engineering methods functionally decompose the systems architecture into one-to-many “part-of” or “owned-by” relationships, while modern software methods organize system capabilities as layers of many-to-many service-oriented relationships. Forcing software into many disjoint parts makes for slow and cumbersome software support of evolutionary acquisition.

Second, traditional systems engineering approaches collect data into a single data dictionary and store it for common use. This was OK for sequential software, but for interrupt-driven concurrent software, it could cause data being used by a program to be modified while the program was interrupted, causing losses in software integrity. Modern object-oriented software approaches address this problem by using “information-hiding” techniques to store data inside objects, requiring more up-front data engineering.

Third, hardware interfaces tend to be static: sensor data flows down a wire, and the sensor-system interface can be represented by its message content, indicating the data’s form, format, units of measure, precision, frequency, etc. In a net-centric world, interfaces are much more dynamic: a sensor entering a network must go through a number of protocols to register its presence, perform security handshakes, publish and subscribe, etc. The next chart describes how this can result in integration problems.

Fourth, hardware relations are assumed to be static and subject to static functional-physical allocation: if the engines on one wing fail, an engine cannot migrate from the other wing to rebalance the propulsion. But in software, modules frequently migrate from one processor to another to compensate for processor failures or processing overloads.
Examples of Architecture Mismatches

- Fractionated, incompatible sensor data management

- “Touch Football” interface definition earned value
  - Full earned value taken for defining interface dataflow
  - No earned value left for defining interface dynamics
    - Joining/leaving network, publish-subscribe, interrupt handling, security protocols, exception handling, mode transitions
  - Result: all green EVMS turns red in integration

A frequent consequence of traditional systems engineering methods that functionally decompose the systems architecture into one-to-many “part-of” or “owned-by” relationships is to create incompatible and hard-to-modify (i.e., incompatible with EvA) software. This is exacerbated by the owned-by relations being propagated into work breakdown structures (WBS) and WBS-oriented management structures. For example, an aircraft has wings as parts, which have ailerons as parts, which have aileron controls as parts, which have sensors as parts, which have sensor software as parts. Further, an aircraft using active controls to stabilize an otherwise unstable aircraft will rely on atmospheric, propulsion and attitude sensors owned by other parts of the aircraft hierarchy to accomplish the stabilization objectives.

In such situations, best-of-breed hardware subsystem contractors are often selected whose sensor data management system (SDMS) software is incompatible, causing major problems as compared to layered software providing unified data management services. And if sensor data management changes need to be coordinated across a wide variety of subsystem management chains, very slow OODA loops result, as discussed in chart 5. These sources of delay are reinforced by current hardware-oriented work breakdown structures such as the current rework of MIL-STD-881 on Work Breakdown Structures, which continues to use hardware-driven WBS structures in its platform appendices, with software fractionated at levels 4 and below. That this is not necessary has been shown by the National Reconnaissance Office, which has been successfully using parallel hardware and software WBS structures for several years. The hardware-driven WBS frequently leads to similar hardware-driven management and earned value structures, one of whose risks is discussed next.

A frequent consequence of hardware-oriented static interface definitions being represented by their message content, as discussed on the previous chart, is that system engineering will credit itself with full interface-definition earned value when it defines the interface message and its content. If no further earned value can be generated for defining the dynamic software interface aspects, they will tend to lose management priority and be deferred until the deficiencies are found during integration, when numerous incompatible subsystem software interface assumptions are found that are expensive and time-consuming to fix. This is a frequent explanation for all-green earned value management system (EVMS) indicators turning red during integration.
Summary: Evolutionary Acquisition (EvA)

- EvA is the preferred DoD strategy for MDAPs (DoDI 5000.02)
  - It is more responsive to DoD’s current and future challenges
- There are several forms of EvA
  - With different strengths and shortfalls
- EvA requires significant advances in current practices
  - In systems engineering (SE) and development processes, particularly to avoid unscalable, easiest-first, sunny-day architectural commitments
  - In rapid, adaptive integration of SE and acquisition management (AM)
  - In financial and human resource allocation
  - In workforce capability and empowerment
  - In research and technology priorities and transition speedup

Initiatives are needed to provide these advances in time for EvA to succeed

The study concluded that EvA approaches are necessary for DoD to sustain its ability to keep its observe-orient-decide-act (OODA) loops inside those of its adversaries. However, the study also found that many traditional acquisition practices are incompatible with effective SE of evolutionary acquisition. These include assumptions that full-capability requirements can be specified up front along with associated full-capability plans, budgets, schedules, work breakdown structures, and earned-value management targets; that most systems engineers can be dismissed after PDR; and that all forms of requirements change or “creep” should be discouraged.

The study also found that other inhibitors to effective SE need to be addressed, such as underbudgeting (SE is the first victim of inadequate budgets); contracting provisions emphasizing functional definition before addressal of key performance parameters; management temptations to show rapid progress on easy initial increments while deferring the hard parts until later increments; and acquisition infrastructure that continues to stimulate sequential hardware-first approaches where agile concurrent hardware-software-human factors SE is needed for rapid fielding and rapid responses to new threats and opportunities. A good example is the current revision of MIL-STD-881, which continues to use hardware-driven WBS structures in its platform appendices, with software fractionated at levels 4 and below. Such EvA-incompatible WBSs also tend to spawn EvA-incompatible management and EVMS structures as well.

The study further found that there are wide gaps between the current technology available to support SE and the technology needed to perform effective SE of complex net-centric, multi-stakeholder systems of systems with emergent requirements, high assurance needs, rapid change, and continuing sources of attractive but disruptive technology. The study recommends a coordinated program to accelerate SE research, technology maturation, and technology adoption in such much-needed areas as systems architecture modeling, evolvability, and tradeoff analysis; rapid, scalable change impact analysis; rapid, distributed, multi-discipline system definition collaboration technology; value-based architecting and design aids; reliable autonomy; innovative uses of multicore processors; powerful, flexible, and composable models, simulations, and testbeds; and incrementally-applicable and ambiguity-tolerant SE tools.

Further, the study recommended that significant initiatives be undertaken to provide the necessary EvA infrastructure of SE-related acquisition and development processes, contracting and incentive structures, milestone decision criteria, financing, program management, and staffing, along with research support to address current gaps between EvA infrastructure needs and capabilities. Projects attempting to succeed at EvA while burdened by DoD’s current acquisition infrastructure could lead to sufficient numbers of projects “failing with EvA” to cause policymakers to give up on it before it has a chance to succeed.
References - I


AM Acquisition Management
AP Anchor Point
AT&L Acquisition, Technology, and Logistics
C2ISR Command, Control, Intelligence, Surveillance, and Reconnaissance
C4ISR Command, Control, Communications, Computing, Intelligence, Surveillance, and Reconnaissance
COCOMO Constructive Cost Model
COSYSMO Constructive Systems Engineering Cost Model
COTS Commercial Off the Shelf
CP Competitive Prototyping
DID Data Item Description
DoD Department of Defense
DoDI Department of Defense Instruction
DOTMLPF Doctrine, Organization, Training, Materiel, Leadership, Personnel, and Facilities
EvA Evolutionary Acquisition
EMD Engineering and Manufacturing Development
EVMS Earned Value Management System
FED Forward Edge of the Battle Area
FED Feasibility Evidence Description
GAO Government Accounting Office
IED Improvised Explosive Device
IMP Integrated Master Plan
IMS Integrated Master Schedule
INCOSE International Council on Systems Engineering
IOC Initial Operational Capability
KPP Key Performance Parameter
KSA Knowledge, Skills, and Abilities
KSLOC Thousand Source Lines Of Code
References - II

Young, J., “Prototyping and Competition,” USD(AT&L) Memorandum, September 19, 2007

MDAP Major Defense Acquisition Program
MS Milestone
MSA Material Solution Analysis
O&M Operations and Maintenance
O&S Operations and Support
OODA Observe-orient-decide-act
PD Production and Deployment
PDR Preliminary Design Review
PPPI (P3I) Pre-Planned Product Improvement
RESL Architecture and known Risk Resolution (from COCOMO II)
ROI Return on Investment
RPV Remotely Piloted Vehicle
SADPO Scoping, Architecting, Developing, Producing, Operating
SAIV Schedule as an Independent Variable
SDMS Sensor Data Management System
SE Systems Engineering
SERC Systems Engineering Research Center
TD Technology Development
UAV Unmanned Aerial Vehicle
UML Unified Modeling Language
USD Under Secretary of Defense
V&V Verification and Validation
WBS Work Breakdown Structure
WSARA Weapon System Acquisition Reform Act
WYTINWYG What You Type is Not What You Get