FEDSOFT

# WHITE PAPER:

## SYSTEMS ENGINEERING COMPLEXITY IN THE GROWING ENTERPRISE

The purpose of this document is to review some current discussions of large system integration and a possible architecture management approach to cope with the emerging complexities. It suggests that the problem lies with the challenges associated with the emergence of highly complex systems combined with a lack of customer understanding of what constitutes "good" Systems Architecture and Systems Engineering.

## ABSTRACT

As the components and sub-components, systems and sub-systems, and systems-of-systems that are part of larger systems-of-systems become larger and more complex; the management process and baseline architecture behind the entire effort will also become more complex. Any approach to large systems development and architecture management must adjust with the technology behind the elements of the system to an amount equal to or greater than the advances in technology, processors, hardware, and software. This paper is a review of some current discussions of large system integration and a possible architecture management approach to cope with the emerging complexities. I suggest that the problem lies with the challenges associated with the emergence of highly complex systems combined with a lack of customer understanding of what good Systems Architecture and Systems Engineering is. Without a collaborative and cooperative architecture design and development management approach, successful program execution and product delivery will remain an elusive target.

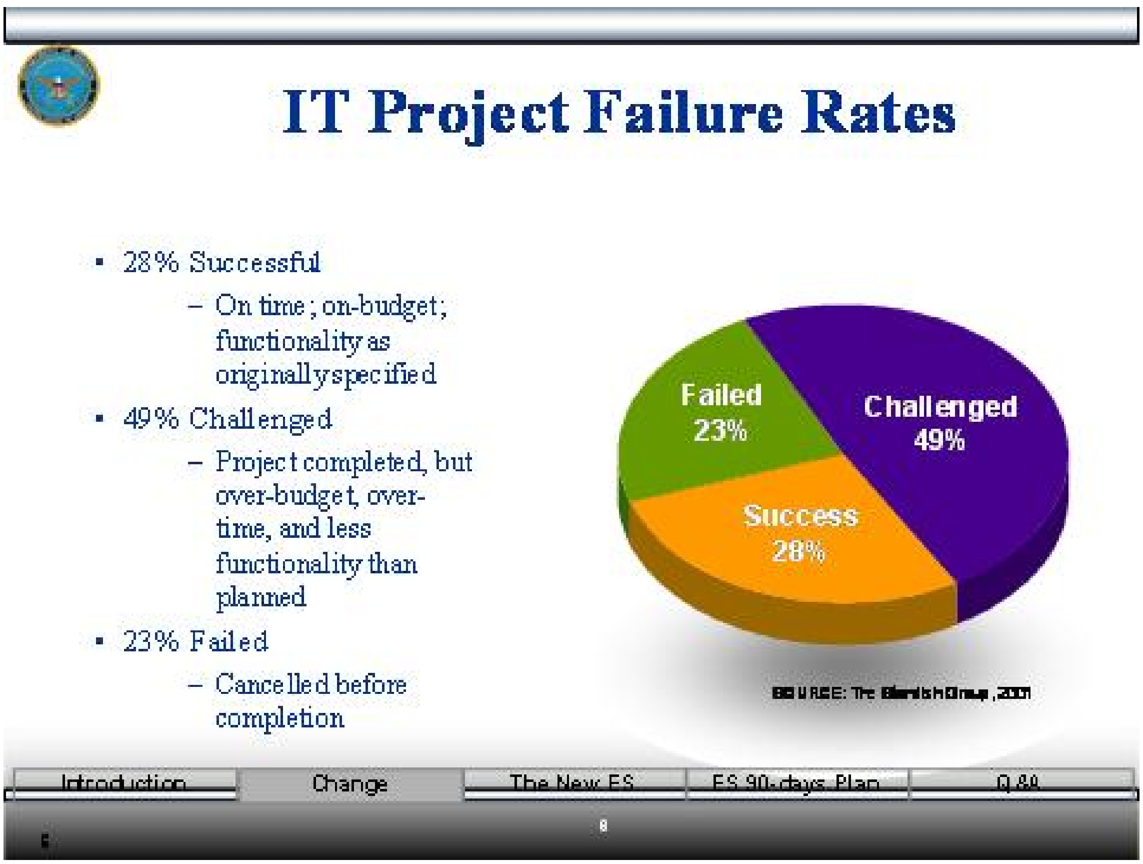


**Keywords:** Architecture, Management, Federal Procurement, customer collaboration, Complex Systems, Collaborative Architecture


## 1. INTRODUCTION

As an attendee at a number of Industry Day presentations by several government agencies from DISA and DIA to the Veterans Administration and the USDA, honest and frank discussions have led to statements like, "Over 25% of our organizations development projects are either a total failure or will be before the end of the fiscal year. Another 50% are severely over-budget, behind schedule, and delivering substantially less then what is needed." This came from the head of the DIA and he said it to the attendees at a conference wherein he was discussing where the agency was going with their procurement dollars in FY2009 through FY2014. Most of us in the audience sat in disbelief as he stated what a lot of us felt but were unable to quantify. This same general sentiment was repeated in three other conferences for three other agencies. The names were different, but the story was the same. As I contemplated the gravity of these pronouncements, it became apparent that the problem was noticeable across organizational boundaries from DoD to the VA to the USDA. In evaluating the current state of the industry, the processes and methods used to determine who will be selected for a procurement of small or large complex systems, and the methods and processes used to execute those programs, the issue seems to be epidemic in the Federal space.

Einstein once said, "The definition of insanity is doing the same thing over and over again while expecting different results." As I have worked with these agencies and seen this cycle repeat itself year after year, I look at the literature that highlights the challenges of designing, developing, and deploying complex systems and wonder when the Einstienian light will illuminate. I believe that the problem lies with the challenges associated with the emergence of highly complex systems combined with a lack of understanding of what good systems architecting and systems engineering really is. This combined with a broken government procurement system leads to the chaos and budget crises faced by all or most of the federal government. LT GEN Charles Croom, the Director of the Defense Information Systems Agency stated during the DoDIIS Industry Day Conference in Orlando back in April 2008, that Procurement at DISA is broken. He stated that contracts are written in a way that the chance of success from day one approaches zero and the successful program is the exception rather than the rule.




An initial analysis of the issues behind the breakdown in the release and fielding of useful systems in the federal government, two major focus areas emerge. The most critical foundation to a successful deployment of a complex system starts with an effective architecture. To get to that solid, effective successful architecture, nothing can be developed in a vacuum, cannot be based on abstract, ill-defined requirements, it must be developed in a collaborative, cooperative environment that respects the role that each side of the contractor-government buyer continuum.

## 2. EFFECTIVE COMPLEX SYSTEMS ARCHITECTURE DEVELOPMENT

The Defense Acquisition Guide (DAG) defines system integration as the, "…process of incorporating lower level systems elements into higher-level system elements in the design solution or physical architecture." Designing and developing complex systems starts with defining the architecture from lower levels of complexity, building up to more complex levels, higher into the architecture.

What is needed is a new way of approaching and managing complex systems design, development, and deployment. I would contend that there are few simple or non-complex systems in the Enterprise today. As an example, e-mail started out as a simple method for communications, but today there are Exchange Servers, digital signatures, encryption, Collaboration Services, Chat, file sharing, desktop sharing, Blackberry, Treo, and other Personal Digital Assistant interfaces, making e-mail a fulltime management task within any company with international reach.

Some of these applications or systems were small, one of a kind band-aids to a specific problem or need, but a lot of the systems were large, complex applications and systems of systems that had enterprise reach and the potential for life or death impact. As the war on Terror continues to intensify, systems and Information Technology support is not decreasing. As the push to manage budgets increases, every agency will need to get better at the delivery of effective, appropriate systems capability with useful, well defined architectures, and break this epidemic of disasters.

The research seems to point to a fairly flat, simplistic approach to complex system architecture development. I would postulate that the most basic approach to complex system architecture development focuses on a set of iterative process completed in series with a repeat cycle for each segment and then for each integration pass. The literature repeats often that complex systems should arise from an existing baseline or product instances, growing and evolving from there (Sheard, 2007). The first step in this evolution from simple system or sub-system is a logical segmentation of purpose or function. Taking a customers need as the starting point and what is known or existing, and dividing the project into manageable segments of meaning and intent. These actionable parts of the whole are then each evaluated, dissected, and molded into a picture that will be used to represent that portion of the whole.

## 3. ARCHITECTING THE SOLUTION - ARCHITECTURE IS A COLLABORATION SPORT

The body of evidence supporting successful programs starts with or is sustained by a healthy understanding of heuristics. “A team producing at the fastest rate humanly possible spends half its time coordinating and interfacing.” (Rechtin, 2002). Any aspect of a program, project, or process that consumes one half of the time has to be considered a critical factor. Add to this volatile mixture a customer who wants value on their investment and is looking to the contractor to deliver, and it becomes apparent that coordinating and interfacing should rapidly evolve into cooperation, communication, and collaboration.

### 3.1 Collaborate on the Need Statement

One of the first things taught to a new System Engineering student is the identification and communication of the customers need. Too often during a review of failed projects, it becomes apparent in hindsight that the customer and the contractor never achieved synergy and understanding about what was to be built, how it was to come to being, and the measures to confirm success.

The first step in developing a team that is capable of completing and delivering what is needed, on time, and on budget, Customer and Contractor need to collaborate and communicate on the need. Most programs develop communications plans and communications and contact plans, but often neglect to execute. Some government customers feel that their responsibility is to point their hired guns (their contractors) at their interpretation of the problem and then dog the effort till the results begin to flow. Some contractors feel that government customers are a nuisance and do nothing but get in the way. None of this is expressed openly, but the results are the same, failed programs and missed opportunities.

Customers and contractors need to decide early that collaboration, communication, and cooperation are best included at the front end of the program rather than during lessons learned. I do not accept that the objective of any customer or contractor is to find a way to fail. If the initial objective is to fail, the quickest route to that end would be not to compete for the work or never start. I believe that with a little extra effort and a non-combative understanding that customer and contractor are both looking for success, reaching a common understanding of the need up front can go a long way in setting the stage for victory.

### 3.2 Adversarial Project Management and the CIA

Some time ago, some agencies within the US Government accepted and taught a form of project management called, ‘Adversarial Project Management’. The idea behind this interaction methodology was to tell the contractors that they were not measuring up to their obligations, not meeting expectations. This communication approach was practiced regardless of the status of the program, successful or otherwise. Proponents of this management philosophy felt that this environment would force or cause the contractor to perform beyond expectations in order to recover. They hoped that the contractor would work harder, creating greater value for the government. What actually happened was a severe decrease in team morale, teamwork, cooperation, and collaboration decreased, causing the results of the project to live up to these false expectations of poor performance.

During this brief period, the Central Intelligence Agency (CIA) chose not to accept this radical management philosophy. They felt that there had to be a better way to manage a project. In 1988 the CIA selected Len Malinowski to develop a management training course that would for a time, revolutionize how the CIA did business.

The objective behind the training was to create a shared experience at all levels between the government and the contractor. A project team paired off with their opposite, one from the contractor side and one from the government. They focused on learning to see the project objectives, methods and biases from each other’s perspective. During this cooperative shared time, they learned to use a common vocabulary and view the project from a common perspective and use a common approach. This shared, cooperative and collaborative approach to the project led to a significant and dramatic increase in productivity and had an effect on the whole agency. (Forsberg, Mooz and Cotterman, 2005)

As the assistant test lead on a CIA program during this time working on an open source image and text storage and retrieval, I saw firsthand the effect of a working relationship with the government that was focused on that common set of experiences and expectations. The time spent on the STARS program was incredibly productive. Communications up the chain, across organizational units, and within each working unit was excellent. Progress was expected and realized in all areas. New technology advances or the use of existing technology in new and innovative ways became a way to advance the solution space and improve the architecture. Within this supportive and sharing setting, progress was a constant and turned out to be the path to the delivery of a high quality, mission critical complex system.

**3.3 Set a Joint Course**

Collaboration is defined on Wikipedia:

> *Collaboration is a recursive process where two or more people or organizations work together toward an intersection of common goals. … Structured methods of collaboration encourage introspection of behavior and communication. These methods specifically aim to increase the success of teams as they engage in collaborative problem solving. Forms, rubrics, charts and graphs are useful in these situations to objectively document personal traits with the goal of improving performance in current and future projects.*

In the INCOSE article *Integration Challenges of Complex Systems* by Bill Haskins and Jack Striegel a discussion was presented on lessons learned from failed complex systems programs. Their premise was that the integration of the multiple layers of a complex architecture fails because there is a lack of ownership of the architecture and solution, a lack of attention, and a lack of communication. (Haskins & Striegel, 2002) In 'The Art of Systems Architecting', the authors stated, "Communications with the client has two goals. First, the architect must determine the client's objectives. Second, the architect must insure that the system to be built reflects the value judgments of the client…". (Rechtin, 2002)

Most people do not think in abstract ways or in a manner that is consistent with today's complex systems. Most people think along a timeline within the three distinct segments of time contained in the past, the present, and the future. Some might be able to further divide that timeline into distant past, resent past, with equal views about the future, but most people do not think in a way that is consistent with what is needed to see what a system is, what it was, and what it needs to become.

In the 1970's the scientific community looked at the complexity of life, considering things like snowflakes, weather, plaques, and national disasters a freak or random events. With further study and analysis it became apparent that there was actually order and understanding in what they saw. They named this new order 'chaos', creating a new science that focused on the order and complexity of these new unpredictable events. (Ollhoff and Walsheski, 2003)

The challenge for system designers and architects is to bring order and understanding to this chaos. As the architect designs and develops the solution for the multiple segments of the end-product, because the end product is being developed with the customer's needs in mind, the customer must be a part of the process and must be able to see the order in the chaos regardless of their predilection or pre-project perceptive abilities. It is the job of the project team, and a primary responsibility of the System Architect to create a cooperative and collaborative environment where understanding and order can be perceived and ambiguity and doubt are satisfied.

### 3.4 Define Success

As part of the plan for success, there must be a clear, common, understanding of what success will look like. As part of the definition of what the customer needs, there must be a measureable set of criteria as defined in a concise architecture that both the customer and contractor accept. This needs to be unambiguous and a clear view of what will be the common understanding of completion.

### 3.5 Enforce Team Communication

Just as with defining success there have to be clear lines of communication between both sides. The project leaders and the project architect need to work with everyone on open communications and focus on –

- A common set of goals - Clearly define what the team objectives are. Include measures of success and make sure that everyone has that common understanding (Kevin Forsberg, Hal Mooz, Howard Cotterman – 2005)
- Acknowledge interdependencies and mutual respect – Too often customers consider contractors the enemy and contractors consider customers impediments. The most effective path to an effective program is for everyone to acknowledge that both sides are needed for an effective team, and all of this must be done in an atmosphere of respect and trust. Clearly defined roles and responsibilities can lead to a broad understanding of the end point.
- A Common Code of Conduct – Define for everyone what is expected. The less that is left to interpretation the better. Expect and define what is acceptable professional behavior for all parties, and there is less of a chance for confusion or misunderstandings.

- Shared Rewards – Corporate management seeks sales, orders, and profit. That is the measure used to gauge the success of a program at top from the boardroom. Community recognition of a job well done can often go a long way to keeping everyone engaged.
- A clearly understood and shared architecture – Define success in the form of a solid, documented architecture. Make it the common working artifact that drives all aspects program from initial concept to system retirement.
- Team Spirit and Energy – Teamwork is not an automatic outgrowth of effective teams. It must be cultivated and developed proactively. Successful teams encourage teamwork, have plans to work the issues constantly, and management sets the example. If the Government PM and the Contractor PM can work together and encourage teamwork, their energy will influence the outcome.

### 3.6 Evolve a Collaborative Architecture

If the architecture is not moving forward, adapting to change, and evolving, consider the future bleak. Stagnation and inflexibility can kill a project by way of the architecture as fast as problems with hardware, software, or people. “If at first you do not succeed, but the architecture is sound, try, try again. Success sometimes is where you find it.” (Rechtin, 2002) This heuristic suggests that you need to go looking for success. Looking and going are action verbs. If you start with a sound viable architecture, work to support an evolution of the ideas and concepts that make up that architecture and allow it to grow from good to great, from simple to complex, from start to finish. An outgrowth of communication, cooperation, and collaboration is a flow of knowledge and ideas that can feed the architecture and push it forward.

“Don’t assume that the original statement of the problem is necessarily the best, or even the right one.” (Rechtin, 2002). In this heuristic it is once again imperative and demonstrable that the plan and architecture must remain fluid and flexible. To allow the problem space surrounding the system architecture to stagnate or remain static will lead to increased opportunities for failure. Revisit the Problem Statement, the Customer Need Statement, and the requirements often with a critical eye towards progress and motion.

In working complex systems like a 787, program discussions, decision making, and architectural changes must be made collaboratively. Boeing has learned from experience on a number of complex aircraft development programs, that communication and decision making must be kept at a level that maintains short paths for approvals. There is a heuristic that defines the exponential amount of time needed to gather approvals that says, “The time in days, $T$, to obtain an approval requiring $N$ signatures is approximately $T = 2^{N-2}$.” (Rechtin, 2002).

In small systems programs, if ***N*** stays small, the time will remain small. As systems grow more complex and risk becomes more expensive with slipping schedules and increasing costs, it has been my experience that more people are inserted into the decision-making and approval process, adding days to approvals, further adding to the risk that budgets and schedules can slip. In the table below you see that if you start with a simple approval chain of 4 people, the growth can rapidly get to the point where the program revolves around decision-making and approvals rather than the intended focus of the program

| Number of Approvals | Days |
|---|---|
| 4 | 4 |
| 5 | 8 |
| 6 | 16 |
| 7 | 32 |
| 8 | 64 |
| 9 | 128 |

**3.7 Architect an Understanding of Risk from the Start**

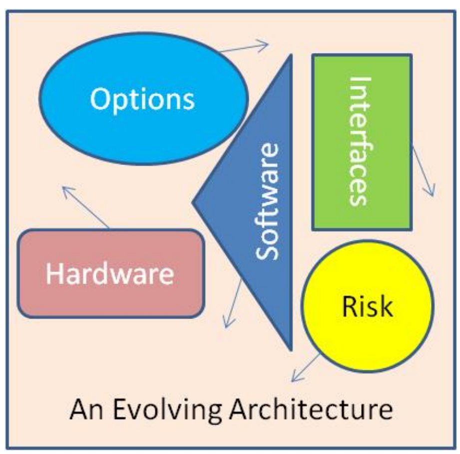
"Risk is in the eye of the beholder, not the Architect." (Rechtin, 2002) Risk needs to be part of the architecture from the earliest discussions, maybe as early as definition of customer need. In defining zero tolerant systems, risk is assumed upfront and addressed at every turn with the end goal of designing risk out of the architecture and the eventual solution.

In considering risk from the start of the program, there are a number of appropriate heuristics as listed below. (Rechtin, 2002)

1. Murphy's Law – If anything can go wrong, it will – In the average software program or hardware design effort, regardless of the amount of planning and preparation, something will go wrong. Work to develop that flexible, evolving architecture that will be able, in the hands of a well trained, collaborative team, to adjust and accommodate disaster as it surfaces.
2. In new missions and markets, expect the unexpected – As with Murphy's Law, success is not the problem, the unexpected is. Remaining flexible as with Murphy can provide the agility needed to adjust and adapt when faced with these now expected challenges.
3. A system optimized for a particular situation is unlikely to encounter that situation – Expecting the unexpected and planning for new situations and opportunities is the most effective method for ensuring that you have it covered. The more you plan, the more flexibility is built into the architecture, the less risk the program will be faced with.

4. You can’t avoid redesign. It is a natural part of design – Architectures and system design are fluid and must remain so for the life of the program. As soon as the architecture solidifies, it is time for redesign or the start of system obsolescence.
5. Predicting the future may be impossible, but ignoring it is irresponsible – No one can claim to know the future. Even knowing that Murphy will become a concern during the life a project does not mean we should not plan for that eventuality. A good architecture and a good architect is looking forward, planning for eventualities as best they can in the most responsible and conscientious manner possible. Understanding all the way that this is a reasonable response to an impossible problem.
6. Don’t ever stop talking about the system – This heuristic can be written to say, ‘Don’t ever stop talking about the architecture.’ A permanent part of every meeting, status report, and review should be a restating of the focus of the architecture with a call for options, concerns, criticism and recommendations. Never let a day go by without expending energy and collaborative effort on ensuring that the architecture is moving and evolving.

Architecting a healthy understanding of risk and planning for adjustments and adaptation is a smart course. Risk needs to a significant part of the architecture as presented below.

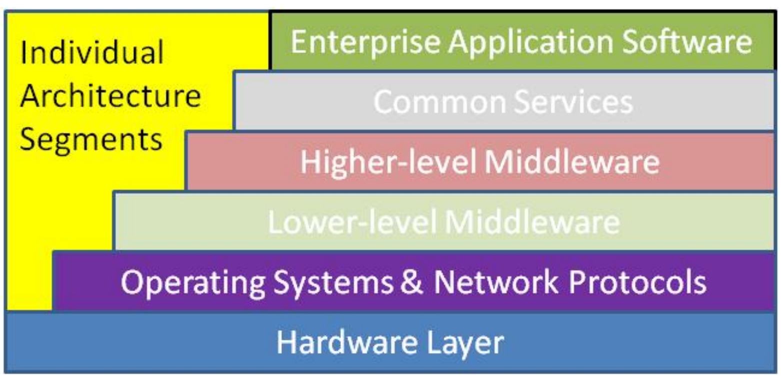



In this graphic, risk is one element of the evolving architecture. It is not more or less important to the design of the solution space. As the system changes and migrates from one state to the next, so must a team’s understanding and appreciation for the role that Risk plays in the answer.

### 3.8 The Architecture Must Consider the Parts as Well as the Whole

The Systems Engineering literature all seems to support the idea that the goal of complex system development is defining reasonable and logical sections or segments of the entire problem set that can be worked, designed, and architected. During the course of this segmentation of the 'whole', the architect must keep foremost in their mind that the end game or end point is in fact the whole. The parts are critical to understanding the whole, but the customer need will not be satisfied by some of the parts, but will be by the sum of the parts that will eventually equal the whole. Jim Ollhoff and Michael Walsheski in Stepping in Wholes says, "Wholes are so related to their parts that not only does the existence of the whole depend on the orderly cooperation and interdependence of its parts, but the whole exercises a measure of deterministic control over its parts." (Ollhoff and Walsheski, 2002 - As quoted in Mayr 1997). The design behind the architecture or the architecture behind the design of each part or segment will have a profound impact on the resulting whole architecture, system and result.

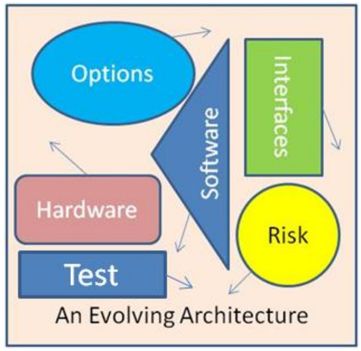
As in the definition of the Evolving Architecture and risk above, the technical details of an architecture must be defined and parceled out in a logical determined fashion. The parts must be organized and managed with the 'big picture' in mind as well as at the parts level.




The customers' needs and desires must be accomplished in the sum of the parts, completed in the whole.

**3.9 Design Test into the Architecture**

Too often with some customers, test is assumed to be a part of development, or not considered at all. "To be tested, a system must be designed to be tested." (Rechtin, 2002) Test cannot be an afterthought. Test must be considered from the start and evolve as part of the architecture at each step. Each segment must be developed and architected with test in mind, and the parts of the whole must be testable from the lowest layer to the interface. In some cases the ability to test and the testability of the whole could become reliant on a system itself. "The test setup for a system is itself a system." (Rechtin, 2002) In complex systems it is common to have a test architecture that is itself a system that can be as complex and difficult as the delivered system will be. Adding test to the Evolving Architecture produces -




Considering test as part of the whole ensures that measures of effectiveness or performance are planned and incorporated into the architecture from the very start.

**3.10 Plan for the Unexpected**

As referenced in the Risk section above, planning for the unexpected needs to become the mantra and watch word for the successful Architect. Options and alternatives should be at the forefront of consideration during the life of a project. "Build in and maintain options as long as possible in the design and implementation of complex systems. You will need them." (Rechtin, 2002) In this heuristic again, the architect can be the focus of the statement and become the driving influence that will determine success.

**4. SUMMARY**

A Systems Architecture needs to emerge from the very start as a product of a collaborative team design. The Architect needs to avoid repeating the same mistakes and errors that their predecessors made. Contractors and customers need to work on a flexible, adaptable, architecture that is never static. Customers and contractors must work at communicating across all segments of the system and architecture. Both sides of the system development program lifecycle need to work towards that common end goal, working with an evolving, always improving architecture.

**References**

1. Bill Haskins, Jack Striegel; Integration Challenges of Complex Systems; INCOSE 2006 – 16th Annual International Symposium Proceedings
2. Gerrit Muller ; Coping with System Integration Challenges in large Complex Environments; INCOSE 2007 – 17th Annual International Symposium Proceedings
3. Sarah A. Sheard ; Principles of Complex Systems for Systems Engineering; INCOSE 2007 – 17th Annual International Symposium Proceedings
4. Kurt A. Richardson, Graham Mathieson, and Paul Cilliers; Complexity Thinking and Military Operational Analysis, 2003
5. Kurt Richardson; The Hegemony of the Physical Sciences: An Exploration in Complexity Thinking, 2004

6. Kurt A. Richardson & Michael R. Lissack; On the Status of Boundaries, both Natural and Organizational: A Complex Systems Perspective, 2003
7. Jim Ollhoff and Michael Walsheski; Stepping in Wholes, 2002
8. Kevin Forsberg, Hal Mooz, Howard Cotterman; Visualizing Project Management – Models and Frameworks for Mastering Complex Systems, 2005
9. Mark W. Maier, Eberhardt Rechtin; The Art of Systems Architecting, 2002
10. E. Mayr, This is biology: The science of the living world", 1997