Risk Management for Systems of Systems©

Dr. Edmund H. Conrow
Risk-Services.Com

Systems of systems (SOS) present some unusual risk management challenges that often are not explicitly addressed, yet can impact the resulting degree of system effectiveness. Potential risks associated with integrating a diverse set of systems and associated hardware/hardware, hardware/software, and software/software often exist; these are made all the more difficult by individual systems at different levels of maturity and potential risks that do not exist at the individual system level. Established risk management processes may be in place for different systems, yet process steps and associated tools and techniques may not be compatible. This article briefly examines some key SOS risk management process issues together with methods for better implementing risk management on such programs.

Most of the literature and government guidance to date on project risk management has been focused on individual programs or systems. Yet systems of systems are becoming more complex and more commonplace in the United States and abroad.

One system of systems that many people in the United States have used without recognizing it is the air traffic control system. In the United States, according to the General Accounting Office (GAO):

> … the en route centers (of the Air Traffic Control system) alone rely on over 50 systems to perform mission-critical information processing and display, navigation, surveillance, communications, and weather functions. [1]

A current Department of Defense (DoD) example of a complex system of systems under development is the Future Combat System (FCS). In building FCS, the GAO says:

> … Army leaders decided to include interoperability with other systems in the FCS design, and design the individual FCS systems to work as part of a networked system of systems with a first-of-a-kind network. [2]

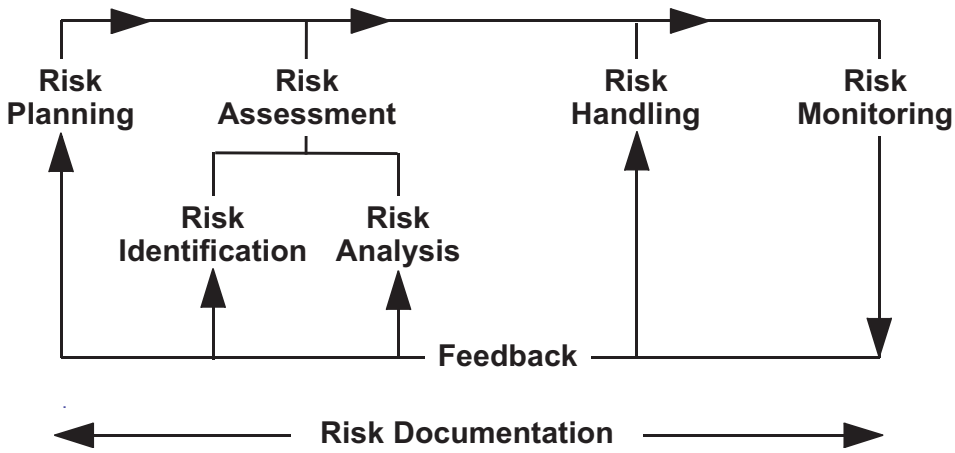
For FCS, the GAO writes:

> … 14 major weapon systems or platforms have to be designed and integrated simultaneously and within strict size and weight limitations in less time than is typically taken to develop, demonstrate, and field a single system. At least 53 technologies that are considered critical to achieving critical performance capabilities will need to be matured and integrated into the system of systems. And the development, demonstration, and production of as many as 157 complementary systems will need to be synchronized with FCS content and schedule. [3]

In this article, I will provide an overview of the risk management process and explore risks that are common to many systems of systems (SOS) implementations along with recommendations for addressing each risk.

© Copyright 2005 by Edmund H. Conrow. All Rights Reserved.

Figure 1: *Risk Management Process*



Risk Management Introduction

Risk management is the act or practice of dealing with risk. It includes planning for risk, assessing (identifying and analyzing) risk issues, developing risk handling options, monitoring risks to determine how they have changed, and documenting the overall risk management program. A simplified risk management process flow is given in Figure 1 [4, 5].

- **Risk planning** is the process of developing and documenting an organized, comprehensive, and interactive strategy and methods for identifying risk issues, performing risk analyses, developing and implementing risk handling plans, and monitoring the performance of risk handling actions.
- **Risk assessment** is the process of identifying and analyzing program areas and critical technical process risks to increase the likelihood of meeting cost, performance, and schedule objectives. *Risk identification* is the process of examining the program areas and each critical technical process to identify and document the associated risk. *Risk analysis* is the process of examining each identified risk issue or process to refine the description of the risk, isolating the cause and determining the effects.
- **Risk handling** is the process that identifies, evaluates, selects, and implements options in order to set risk at acceptable levels given program constraints and objectives. This includes the specifics on what should be done, when it should be accomplished, who is responsible, and what are the associated cost and schedule. Risk handling options include assumption, avoidance, control (also

known as mitigation), and transfer. The most desirable handling option is selected and a specific implementation approach is then developed for this option and documented in a risk-handling plan.

- **Risk monitoring** is the process that systematically tracks and evaluates the performance of risk handling actions against established metrics throughout the acquisition process, and provides inputs to update risk-handling strategies as appropriate. Risk monitoring also provides risk-related information to the other processing steps via the feedback function (as illustrated in Figure 1).
- **Risk documentation** is recording, maintaining, and reporting assessments; handling analysis and plans; and monitoring results. It includes all plans, reports for the program manager and decision authorities, and reporting forms that may be internal to the program.

While the above items (with the exception of risk documentation) are related to specific process steps, it is equally important that risk management is properly implemented following appropriate human and organizational behavioral considerations. For example, both *top-down* (program manager lead) and *bottom-up* (worker-level daily performance) are necessary to provide a suitable environment for effective risk management. It is all too common that upper management is disinterested in risk management or sends mixed messages to working-level personnel. Yet, without working-level personnel assimilating risk management principles into their daily job function, it will be difficult at best to have successful risk management.

In general, it is more important and more difficult to create the proper culture on a program to inculcate risk management than it is to master the tools and techniques for the process steps.

System-of-System Issues

I will now briefly discuss seven relatively common issues for SOS risk management [6], which are given in Table 1. (See Boehm, et. al. [7] for a discussion of some software-intensive SOS risks.) The format used in each case first describes and frames the issue and is then followed by recommended approaches for addressing each issue.

Number	Issue	Issue Summary
1	Multiple Stakeholders	Differences in stakeholder's behaviors will often lead to contention and potentially sub-optimal design solutions, funding allocation, schedule priority, and increased risk.
2	Multiple Risk Management Processes	Differences in risk management processes and their implementation can lead to the omission of risks as well as exaggeration of other risks.
3	Long Life Cycles	Non-uniform acquisition maturity potentially complicates risk management.
4	Common Technical Risk Classes	Technical risks are often examined, evaluated, and managed separately, which may not provide insight into potential strengths/surpluses and weaknesses/shortfalls.
5	Integration Risk	Integration risk is often not explicitly evaluated.
6	Functional Performance Risk	Functional performance risk is often not explicitly evaluated.
7	Interface Complexity	It is generally difficult to evaluate interface complexity and accurately relate it to risk.

Table 1: *Common Systems of Systems Risk Management Issues*

1. Multiple Stakeholders

Multiple buyers, sellers, and other stakeholders will generally exist, and the behavior of each group is not homogeneous. The objective function associated with cost, performance, and schedule (CPS) will be different for different parties. These differences will often lead to contention and potentially sub-optimal design solutions, funding allocation, schedule priority, and increased risk [8].

For SOS, multiple prime contractors may exist at the individual system level; these contractors are both buyers from lower-level contractors on an individual program and sellers to both the systems of systems lead contractor and the government. Hence, a variety of objective functions will typically exist at the systems of systems level and reflect different preferences for CPS and associated risk.

In the development of government systems, both the buyer (e.g., government) and seller (e.g., contractor) typically favor increased levels of performance, while the buyer often favors decreased cost and schedule, and the seller favors increased cost and schedule. A common result of this imbalance in both DoD and NASA programs is that performance is the dominant variable and cost and/or schedule are adjusted during the course of the development phase to meet performance requirements [8]. Issues resulting from sub-optimal CPS trades often translate to considerable risk when they are discovered late in the development phase because there is limited ability to efficiently modify designs, etc.

One method to alleviate such problems is to systematically investigate CPS and associated risk in all CPS trades, not just one or two of the three dimensions. Furthermore, the three dimensions of risk should be integrated along with CPS trades to yield a cohesive representation of the potential solution space. In systems of systems, it is common to find marginal risk management focused on technical risk, and weak cost- and schedule-related risk management.

An aid to effective risk management is to have suitable CPS risk management implementation and integration through a central risk management process for each program, as well as at the SOS level. (It is surprisingly common to find separate pockets of CPS risk management within a large-scale program, often with limited program-level integration. This behavior is counter-productive and can lead to weak risk management.)

In addition, differences in the party's objective functions and resulting behaviors should be recognized to avoid *surprises*, balance risk across systems, and to help facilitate mutual awareness and the development of potential solutions prior to risk issues becoming problems later in the program. (Note: a problem is defined here as a risk issue that has occurred [probability = 1].)

For example, a system under development by one government organization often reported risk levels for challenging subsystems that were lower than similar subsystems under development by a different organization. (Here, both systems were in competition with each other and only one would potentially be deployed.) After some time, the government organization responsible for SOS integration instructed the two other government organizations that credible risk analysis results and risk handling plans were far more important than artificially low risk scores.

This *message* was received by both government developmental organizations, and helped to level the playing field between them. This led to increased risk management effectiveness at the SOS organization because fewer resources were needed to evaluate and correct the imbalance in risk analysis results.

2. Multiple Risk Management Processes

Multiple risk management processes will generally exist for SOS. These processes should be, but are often not highly compatible. Without particular attention, this can contribute to weak or ineffective risk management. Risk management differences between systems – and possibly organizations associated with a given system – will likely occur in risk identification and analysis methodology, and the development of risk-handling plans. These process and associated implementation differences can lead to omission of some risks as well as the exaggeration of other risks.

I will now briefly address some common issues associated with different risk management process steps when multiple risk management processes exist.

Unstructured and Incomplete Risk Identification

Risk identification is often performed in an unstructured manner and typically uses a small subset of available approaches. The result of these shortcomings is that potential risk issues can be missed and may become problems later in the program.

At least six different risk identification approaches exist such as those based upon the work breakdown structure (WBS), requirements flow-down, and key process evaluation; each of these approaches should be considered [5]. Typically only a few of these methods are used on a large-scale program, yet each should be considered. This shortfall may in part be related to an organization's risk management heritage. That is, organizations with a strong focus on process-level risks (e.g., design and test) may have limited experience with the WBS approach.

In addition, for SOS, a methodology should be used to perform a top-level risk evaluation for each individual program as well as across the programs for items not associated with lower WBS levels. For example, a program is top-level (WBS 1) for its system. However, a particular program is likely second- or third-level WBS for SOS (whose top-level is WBS 1). Some candidate risks may exist at higher WBS levels (e.g., 1-3) and may not manifest in an easily recognizable manner or even exist at lower WBS levels.

Other potential risks may be better addressed across programs at a top level within the SOS and not at lower levels within the individual systems. For example, networking architectures should be addressed at the SOS level (top-down).

Differences in Risk Analysis Methodologies

A variety of risk analysis methodologies will typically exist for SOS. This can be problematic since variations in the resulting risk levels for the same item evaluated by different organizations or across different programs may be non-trivial (e.g., vary by one or more risk levels). When different organizations evaluate the same risk issue, a significant difference in estimated risk level may result due to differences in how they perceive risk (e.g., risk tolerance) as well as from using different methodologies.

The organization(s) responsible at the SOS level may have to develop a *Rosetta stone* to compare risk analysis results between organizations and translate results at a lower WBS level to a higher WBS level. Likewise, such organizations should evaluate key risks at the individual program level as well as across programs when possible to ensure that appropriate and consistent risk levels exist.

> *"In general, it is more important and more difficult to create the proper culture on a program to inculcate risk management than it is to master the tools and techniques for the process steps."*

Unfocused Risk Handling Strategies

Risk handling strategies are often developed in an ad hoc manner and without regard to strategies in place for other risks. A focused risk handling strategy should be used for each risk that management (e.g., risk management board) chooses to address. The strategy should evaluate possible options (assumption, avoidance, control, transfer), select the best option, and then develop the most appropriate implementation approach for that option.

This approach should be used at both the individual program and SOS level. In addition, a *top-level* examination of risk handling strategies across programs should be performed to identify resources that may be applied from one strategy to another, as well as potential constraints across strategies on the quantity and timing of resources available. (See the related discussion in the "Common Technical Risk Classes" section.)

3. Long Life Cycles

SOS can be expected to have long life cycles, ranging from many years to decades. The individual programs may have different levels of maturity varying from early development to operations/support. The resulting non-uniform acquisition maturity potentially complicates risk management at the SOS level. For example, the resulting interactions and integration of some programs in early to mid development and others fielded (thus in operations and maintenance) are often with risk.

Conversely, fielded systems often pose constraints on developmental systems from a SOS perspective because of the integration and operations framework that is developed. However, developmental systems may impact fielded systems within a system of systems due to unanticipated programmatic and/or technical issues that may result.

The risk management process should be tailored to each program within the systems of systems and each corresponding program phase. In addition, the risk management process at the SOS level should not be static but should evolve over time as individual program maturity and the overall level of integration increases, as new systems are added and as additional data is available.

Risk issues that exist and the level of information available about specific risks will vary from early development to operations/support. For example, non-trivial, architecture-level design and technology problems may manifest in early to mid development, while manufacturing and integration problems may be present in mid to later development, and support-related problems may follow system deployment.

Each of the resulting risk issues should be evaluated in the early development phase as part of the trade process and in later program phases as appropriate in order to address them before they become problems. The risk handling plan content and implementation schedule will vary with acquisition, resource availability, and time-urgency considerations during the course of the acquisition cycle. In addition, relatively little information may exist for some risk issues early in the development phase, and the result-

ing uncertainty in the estimated risk level may be non-trivial. The quality of information available and the level of certainty should increase during the course of the program and lead to improved risk handling actions (all else held constant).

4. Common Technical Risk Classes

While technical risks are often examined, evaluated, and managed separately, a finite number of technical risk classes often exist in a given program. Grouping technical risks into risk classes can provide program decision-makers with insight into potential strengths/surpluses and weaknesses/shortfalls associated with processes, personnel, other resources, etc.

Some common technical risk classes often include but are not limited to design, functional performance, integration, resource availability, support, and technology. Broadly speaking, many types of risk outside of pure programmatic entities (e.g., cost and schedule) may be classified as technical risk. Technical risk classes can exist from low WBS levels to the program level (WBS level 1) or SOS level. It is common that several of these risk classes are not explicitly evaluated during the course of the program. I will now briefly discuss how common technical risk classes can be addressed in different risk management process steps.

Risk Planning

At the individual program level as well as the SOS level, potential risk classes should be explicitly identified as part of the risk planning process, included in the Risk Management Plan (or equivalent), and updated as warranted. This is important since the common practice of selecting risk classes during risk identification oftentimes leads to some risk classes and corresponding candidate risks being omitted.

Risk Identification

A risk identification framework should be used that incorporates standard techniques (e.g., WBS level, requirements flow-down, and key processes) that are selected and adjusted by risk class and program phase. For example, an initial review of key processes (e.g., design, manufacturing, and test) should be performed early in the development phase to identify potential risks. This review should be updated and expanded during the development phase to provide sufficient opportunity to address shortfalls and increase maturity prior to critical program need.

Technology risk, however, is better addressed at the WBS level. This evaluation should be initiated early in the development phase and continued during the development phase until the technology has matured to a satisfactory degree.

Risk Analysis

Tailored risk analysis methodologies should be available for specific risk classes. For example, it is generally not sufficient to use a single, generic, probability of occurrence scale (e.g., very high = *E* to very low = *A* where *E* > *A*) when performing a technical risk analysis because many risk issues (e.g., development maturity) cannot be readily framed into a question associated with probability level.

> *"... the risk management process at the SOS level should not be static but should evolve over time as individual program maturity and the overall level of integration increases, as new systems are added and as additional data is available."*

For example, if a hardware unit in the early developmental stage exists and a fully operational unit is desired using a generic probability of occurrence scale (as above), this can lead to substantial uncertainty as to what level should be selected, and potentially erroneous results. In this particular example, ordinal probability of occurrence scales tailored to unit maturity (e.g., scientific research = *E* to fully operational = *A*) and other potential risk classes (e.g., manufacturing) are often much better suited and can help reduce the level of misscoring and provide more consistent results.

(Note: maturity-based scales, such as Technology Readiness Levels [TRL], *do not* estimate risk, but only one component of the probability of occurrence term. Risk is the product or combination of probability of occurrence and consequence of occurrence. Since TRL and other such scales are unrelated to consequence of occurrence, they do not in and of themselves provide an estimate of risk.)

Risk Handling

Risk handling strategies should be overlaid for common risk classes across WBS levels at the individual program level and the SOS level to identify potential resource issues in a timely manner. For example, if high-performance custom microelectronic components are needed there may be a limited number of suppliers capable of developing and fabricating such parts. If individual orders are examined within a program, the resulting number of different devices may be small, but when examined across programs the quantity may lead to supplier resource shortfalls (e.g., workstations, software licenses, trained personnel, and fabrication, test, and screening capacity) and contention for these resources.

At the individual program level, there may be no apparent risk, but when viewed at the SOS level the resource-related risk may be considerable. This is all the more important if the supplier has fundamental process difficulties in design, testing, or manufacturing because an issue affecting parts for one program may also impact the SOS level or in some cases an entire industry. In such cases, it may be necessary to understand common resources at the supplier level and prioritize potential needs across the program, SOS, or even industry to reduce the level of potential risk whenever possible.

5. Integration Risk

Integration risk is present on many types of programs and is pervasive on SOS by its very nature, yet is often not explicitly evaluated. Hardware/hardware, hardware/software, and software/software are common forms of integration risk. Multiple layers of integration risk are also common, from low to high WBS levels (e.g., 5 to 1) but also across programs for systems of systems. In addition, new forms of integration risk such as net-based integration issues not commonly seen at the individual program level may occur at the SOS level.

The potential level of integration risk is often substantial because of a tendency to underestimate integration difficulty, and simultaneously overestimate the maturity of items that require integration. This is all the more problematic when integration risks manifest late in a program because the ability to trade CPS is typically limited versus manifesting earlier in the program. The result for govern-

ment programs (e.g., DoD and NASA) is often non-trivial cost and/or schedule growth, while performance degradation are typically small [8].

One helpful strategy for alleviating integration risk is to increase attention to potential integration issues throughout the life cycle – beginning in early development rather than focusing on them late in the development process. This can include using adaptable acquisition models (e.g., spiral), carefully developed interface control documents, and early prototyping and perceptive testing to identify potential issues early when there is greater flexibility to trade CPS.

In addition, the transfer risk handling option should be considered for integration issues – do not simply default to the control (mitigation) option. Oftentimes, the transfer option is thought to be limited to insurance, guarantees, warranties, and similar approaches when it also encompasses a variety of other methods such as transferring risk between interfaces, hardware and software, different organizations (e.g., prime versus subcontractor), and even programs. In some cases, this option may alleviate the level of potential risk (e.g., an inexperienced contractor passing real-time software development to a teammate with considerable experience in this area), so long as the recipient actively works the potential risk rather than passively accepting it.

6. Functional Performance Risk

SOS level functional performance risk may include the ability to demonstrate that desired functions or requirements can be met to a specified performance level. This is a different and somewhat converse concept than design risk, which generally assumes that a requirement *can be met* by the nature of the design. Functional performance risk is rarely estimated, yet functional performance shortfalls can translate to problems late in the program if insufficient progress has been made in demonstrating the performance level of key functions that can be achieved.

The probability of occurrence term of functional performance is often maturity based – and scales that incorporate, for example, unverified analytic modeling to in-field testing from less to more mature might represent a coarse ordinal sequence for use. Initial modeling, simulation, and emulation followed by appropriate incremental demonstrations, prototyping, and testing can be helpful throughout the development and integration cycle to potentially reduce functional performance risk to an acceptable level. Whenever possible, avoid an *all or nothing* demonstration and testing approach late in the program since this will often fall short of achieving necessary performance levels and permit little time for recovery versus an incremental approach maintained during the development phase.

7. Interface Complexity

Complex hardware and software interfaces will often exist within individual programs as well as in SOS. While there may be a desire to explicitly treat complexity in a risk analysis, it is generally difficult to accurately relate complexity to risk. Furthermore, efforts to estimate the risk of interface complexity directly may lead to uncertain, subjective, and/or erroneous results.

Interface complexity is typically related to the probability of occurrence term of risk and unrelated to consequence of occurrence. However, it is generally very difficult to develop specific relationships between complexity and probability of occurrence. While the notion that more complex interfaces should have a higher probability of occurrence (all else held constant) is often reasonable from a qualitative or ordinal sense, it may not be possible to confidently say how much higher the resulting probability level is than an interface with a lower complexity level, and inaccurate and/or uncertain estimates may result. Instead, the analyst should consider whether or not interface complexity could be mapped to other technical risk classes that can then be more readily evaluated. These risk classes can include, but are not limited to, design, integration, and support risk. (See the discussion associated with integration risk.)

Conclusion

Complex technical and implementation issues will exist for SOS that may be far more difficult to deal with than for simpler implementations or individual programs. Risk management can play a key role in addressing many such issues. The seven risk management issues and recommendations for addressing them presented here are applicable to a variety of SOS and provide a starting point to the reader to apply to their programs.◆

References

1. United States General Accounting Office. Air Traffic Control. GAO/AIMD-97-30. Washington, DC: GAO, Feb. 1997: 20.
2. United States General Accounting Office. FCS Program Issues. GAO-03-1010R. Washington, DC: GAO, 13 Aug. 2003: 2.
3. United States General Accounting Office. The Army's Future Combat Systems' Features, Risks, and Alternatives. GAO-04-635T. Washington, D.C.: GAO, 1 Apr. 2004: 9.
4. Department of Defense. Risk Management Guide to DoD Acquisition. 5th ed. Vers. 2.0 Ft. Belvoir, VA: Defense Acquisition University, June 2003 <www.dau.mil/pubs/gdbks/risk_management.asp>.
5. Conrow, Edmund H. Effective Risk Management: Some Keys to Success. 2nd ed. Reston, VA: American Institute of Aeronautics and Astronautics, 1 June 2003.
6. Conrow, Edmund H. "Risk Management for Systems of Systems." 2004 Systems and Software Technology Conference, Salt Lake City, UT, 21 Apr. 2004.
7. Boehm, Barry, A. Winsor Brown, Victor Basili, and Richard Turner. "Spiral Acquisition of Software-Intensive Systems of Systems." CrossTalk May 2004: 4-9 <www.stsc.hill.af.mil/crosstalk/2004/05/0405boehm.html>.
8. Conrow, Edmund H. "Some Long-Term Issues and Impediments Affecting Military Systems Acquisition Reform." Acquisition Review Quarterly 2.3 (Summer 1995): 199-212.

About the Author

Edmund H. Conrow, Ph.D., is a risk management consultant to government and industry with more than 20 years experience. He has helped develop much of the Department of Defense's best practices on risk management and has also served as a risk manager on a variety of programs. Conrow is the author of "Effective Risk Management: Some Keys to Success." He has doctorate degrees in both general engineering and policy analysis.

Risk-Services.Com
P. O. Box 1125
Redondo Beach, CA 90278
Phone: (310) 374-7975
E-mail: info@risk-services.com