Sources of Performance Risk in Complex System Development[1]

Tyson R. Browning
Lockheed Martin Corporation
P.O. Box 748 MZ 2222
Ft. Worth, TX 76101
Phone: +1-817-777-2043
E-mail: *tyson@alum.mit.edu*

Abstract. Developing complex systems is an exercise in uncertainty reduction and risk management. This article continues a previous paper, synthesizing research findings from literature and interviews with experienced product development managers and system engineers at a major aerospace company regarding the sources of uncertainty underlying one category of risk, product performance risk. Relationships between sources of performance uncertainty are shown using a causal framework. Consequences of performance uncertainty and shortfalls are also discussed. The goal of this preliminary work is to highlight areas for increased attention in planning and management processes, while providing a systems view of the effects of certain actions.

INTRODUCTION

Complex system product development involves enormous risk. This risk stems from uncertainty regarding product performance in the marketplace and the ability of the development process to deliver that product within a given schedule and budget—and the consequences of undesirable outcomes. From one perspective, product development is a process of uncertainty reduction and risk management. Markets and customers are studied to derive efficacious product design and pricing criteria and a product introduction window of opportunity; designs are developed to meet these goals; and programs are managed and controlled to keep cost and schedule within acceptable limits. Bettering our understanding of the sources of risk in the product development process is fundamental to improving it.

This paper extends an earlier paper on sources of schedule risk in complex system development (Browning 1998b). After providing definitions for several categories of risks, this paper focuses on the sources of product performance risk at the development program level. (One could also examine risks from a program portfolio perspective or at any of a number of levels within a program.) Practicing managers will hopefully find the approach taken useful, allowing them to consider the impacts of their decisions in new ways.

CATEGORIES OF RISK

Assimilating the risk taxonomies of many authors yields the following categories of product development risk: development cost risk, schedule risk, performance risk, technology risk, market risk, and business risk (AFSC/AFLC 1988; Boehm 1989; Brekka 1994; Conrow 1997; DoD 1992; Draves 1993; DSMC 1983; DSMC 1990; DSMC 1998; Hall 1998; Hoy 1996; Justice 1996; Michaels 1996; Reinertsen 1997; Roberts 1996; Sarbacker 1997; SEI 1996; Shishko 1996; Smith 1991; Souder 1993). Each is defined in Table 1.

METHODOLOGY

The identification of risk drivers for this research involved three stages:

1) Consultation of literature, examining the approaches others have taken to representing performance variance and its sources,
2) Numerous interviews with managers and system engineers in a variety of programs at a developer of large commercial and defense systems, soliciting insights on risk drivers, and
3) A Delphi-type survey of managers and system engineers at the same company, validating previous results as to significant sources of risk and their causes and supplementing them with additional views.

With a list of uncertainty drivers—many of them unaddressed in previous studies—the next step involved constructing causal diagrams to represent their relationships. These diagrams represent strawman frameworks for understanding the sources of performance risk in development projects. They do not purport to be finalized or airtight. They endeavor to be comprehensive at a reasonable level of

[1] This work draws from material in (Browning 1998a). Research funding by Lean Aerospace Initiative is gratefully acknowledged. The author would like to thank the managers and system engineers who contributed their observations, experiences, and comments to this paper. Charlie Boppe, David Grose, Ross McNutt, and Eric Rebentisch also provided helpful suggestions.

abstraction, but they no doubt fall short in terms of what factors and relationships should ultimately be included. Certainly, there is opportunity for future extension and improvement. The following section describes the resulting performance risk framework.

Performance risk	Uncertainty in the ability of a design to meet desired quality criteria (along any one or more dimensions of merit, including price and timing) and the consequences thereof
Schedule risk	Uncertainty in the ability of a project to develop an acceptable design (i.e., to sufficiently reduce performance risk) within a span of time and the consequences thereof
Development cost risk	Uncertainty in the ability of a project to develop an acceptable design (i.e., to sufficiently reduce performance risk) within a given budget and the consequences thereof
Technology risk	A subset of performance risk: uncertainty in capability of technology to provide performance benefits (within cost and/or schedule expectations) and the consequences thereof
Market risk	Uncertainty in the anticipated utility or value to the market of the chosen "design to" specifications (including price and timing) and the consequences thereof
Business risk	Uncertainty in political, economic, labor, societal, or other factors in the business environment and the consequences thereof

Table 1: Definitions of Dimensions of Product Development Risk

PERFORMANCE RISK

The design process consists of evolving a design to the point that it is able to meet the targets, goals, or specifications determined from perceptions of what will provide value to the market. As a product development effort proceeds, expending cost and schedule to perform design activities, the performance level or quality of the design concept improves in its estimated, analyzed, and verified ability to meet design goals. While design work occurs, unless actual design performance is somehow verified, uncertainty remains as to the ability of the design *in fact* to meet specifications. *Performance risk is uncertainty in the ability of a design in its current state to meet desired quality criteria (along any one or more dimensions of merit) and the consequences thereof.* In other words, performance risk is the degree of uncertainty in the development and deployment process that may keep the system from meeting its technical specifications or that may result in the system being unsuitable for its intended use (AFSC/AFLC 1988)—and the consequences thereof. As technology is often intrinsic to a design's ability to deliver performance criteria, technology risk is a large subset of performance risk and is dealt with separately (not covered here).

Here, design performance is defined broadly, including system functionality, capability, reliability, price (or life cycle operating cost), delivery timing, number of features, conformance to specifications, durability, serviceability, aesthetics, perceived quality, size, weight, speed, and other measures. The most important aspects of performance—those contributing most to performance risk—will vary with the needs and desires of the market, and should be weighted appropriately in decisions. These dimensions of interest can be thought of as the "whats" a design must satisfy, made explicit by contemporary design methodologies such as Quality Function Deployment (QFD—Clausing 1994; Hauser 1988). Key metrics for gauging aspects of performance include the technical performance measure (TPM—DSMC 1990; Shishko 1995), the measure of effectiveness (MOE—Shishko 1995), price, availability date, and number of defects or bugs. This paper will refer to all such metrics generically as *performance measures* (PMs). Most projects will have numerous PMs to monitor. Some stem directly from customer requirements or perceived market needs. Additional PMs derive from decisions made as the design unfolds (i.e., derived requirements).

From one perspective, *the design process is the course followed to reduce performance risk to acceptable levels*. Performance risk is practically zero when a product that *in fact* meets all requirements, specifications, targets, and goals is delivered. (Whether or not these goals are in fact the best ones, thereby producing a profit for the company, is a matter of market risk.) Performance risk is reduced by setting system parameters to levels that meet requirements, through trade studies and design decisions, and by verifying these results through performance engineering (test and validation). For complex systems, this process is iterative, with each successive iteration intended to reduce performance risk further.

CAUSES OF PERFORMANCE UNCERTAINTY

This section explores the drivers of performance uncertainty—i.e., the factors precluding exact knowledge of product performance level. A strawman framework assimilates these factors. Its purpose is to stimulate thinking about the sources of performance uncertainty, facilitating their identification and inclusion within new product development process models and management policies.

The performance uncertainty driver framework is shown as a causal diagram in Figure 1. A causal diagram shows cause and effect relationships between variables (represented by the directions of the arrows), assuming "all else being equal." Since the framework remains under development, some of the variables may not in fact represent direct causes but rather proxies for certain causes and/or effects. Also, the distinction between cause and effect becomes blurred in a dynamic model. For simplicity, Figure 1 has most of the feedback removed. (For example, performance uncertainty can cause performance corrective actions, which in turn have a variety of effects on the performance uncertainty drivers.) Variables in <variable> notation represent names used in more than one place, mainly in the other frameworks (schedule, development cost, and technology) not discussed in this paper. Some, such as <*Schedule Uncertainty*>, are also enclosed in boxes. These represent factors directly involved in the other risk categories; most of these variables are discussed in (Browning 1998b). The positive and negative signs in the diagram indicate the directionality implied in cause–effect relationships, all else being equal. For example, performing more design reviews increases the level of design evaluation, as indicated by the + alongside the arrow between them. Similarly, as the amount of design evaluation increases, performance uncertainty decreases, all else being equal.

Figure 1: Causal Framework of Factors Contributing to Performance Uncertainty

1.1 Design Work and Iterations

Design work (development, decisions, analyses, etc.) is performed by product development activities, many of which generate information for each other that, depending on when it is received, may cause rework in the process. Many complex system design problems require input and work from multiple activities and must expect several rounds of information exchange before converging on a solution. Both intentional and unintentional (planned and unplanned) repetitions of activities therefore occur in product development. This work and rework is intended to increase overall performance and to decrease performance uncertainty.

a. Number of Intentional Iterations. More available time and money can mean more planned iterations, and more iterations mean more opportunities to refine the design. Intentional iterations are enacted for just this reason. Design is fundamentally an iterative process (Kline 1985; Whitney 1990). Coupled activities cannot settle on a single set of parameter values and configuration choices that satisfy an entire set of complex requirements in one pass. Multiple configurations are put up as strawmen, and many virtual and sometimes physical prototypes are necessary to reveal issues initially overlooked. Indeed, most contemporary design process frameworks and flowcharts make explicit note of the iterative nature of design (e.g., Alford 1994; Blanchard 1990; Clausing 1994; DoD 1992; DoD 1994; Pugh 1991; Shishko 1995).

An explicit link between number of iterations and quality level of a design exists in theory but has not been conclusively demonstrated for a variety of design processes (Leong 1996; Safoutin 1996; Smith 1997). One could conceive of situations where additional iterations provide little or no increase, or perhaps even occasional decreases, in design quality. Research continues to validate the connection between iteration and design quality.

Although not shown in the diagram, *iteration productivity* is a rate variable for the amount by which successive iterations decrease performance uncertainty.

b. Number of Unintentional Iterations. While one should seek to eliminate unintentional iterations in a design process, they can have the positive side effect of refining the design. Yet this benefit comes at a much higher cost because the iterations are the result of mistakes, inadequate information from poorly sequenced activities, etc.

1.2 Design Evaluation

Performance uncertainty is reduced primarily through design evaluation efforts. After design decisions and/or prototypes are made (*design work and iterations*), analyses and tests are conducted to check the efficacy of those decisions and/or prototypes in meeting design goals. Several factors affect the quality and amount of design evaluation:

a. Design Reviews. More formal and informal reviews of the design provide additional opportunities to catch problems and determine if the design indeed meets expectations. This minimizes performance uncertainty. The frequency of design reviews (Ha 1995) and performance validation activities (Bell 1987) has a bearing on the rate of rework discovery (Cooper 1993a; Cooper 1993b; Cooper 1993c), which is an indirect measure of design quality. However, too frequent design reviews can choke the efficiency of a project. Design reviews must occur at the most advantageous times, when performance uncertainty can be reduced the most and the review is likely to discover the most problems early—i.e., when they will generate the most useful and valuable information (Reinertsen 1997). Many have noted the advantages of periodic, external, independent reviews in providing fresh, unbiased assessments.

b. Design Concept Initial Quality. The number of iterations required to converge the design parameters to acceptable levels is greatly dependent upon the quality of the design concept—i.e., on the starting points from which the search and convergence algorithm of design work begins (Ramachandran 1992). The ability of conceptual designers and system architects to propose quality system configuration propositions from which to begin design space exploration and concept refinement depends greatly on *product complexity* and the factors affecting it, *product and process novelty*, and *requirements quality, simplicity, and stability*. Of course, quality baseline configurations also depend on the experience and expertise of the architects and configurators.

c. Verification and Validation Testing. Activities to verify and validate design performance include developing and using simulations, prototypes, models, checklists, fault tree analysis techniques, failure modes and effects analysis (FMEA) techniques, etc. These can be implemented to provide more or less quality depending on the experience level and number of resources applied. A system can be designed to be more "testable" or more easily verifiable. To a very great extent, the amount and quality of verification and validation given to any design depends on the available time and budget, although a process of management decision actually makes the determination based on the amount of performance uncertainty. (This is an example of one of the interactions or feedbacks not shown explicitly in the framework.) In addition, the attitude of planners and managers towards testing can affect the quality of design evaluation and performance uncertainty: if planners assume that all tests will be successful, for instance, this will compromise plans and estimates of uncertainty in a variety of areas.

d. Communication, Coordination, and Integration Quality. The caliber of communication and coordination within the project has a bearing on the appropriate placement of design evaluation steps within the process and on the availability of the results of these studies so as to truly increase the known quality of the design. Exchange of information within the project can facilitate design issue discovery. For example, Northrop Grumman found that early "build-to packages" (BTPs) on one program were only 25% correct; later ones were 60% correct (so only four out of every ten required rework). Having even a rough BTP released for others to analyze facilitated rework discovery (and thus doing more iterations faster) (McIlroy 1997).

1.3 Product Complexity

Performance uncertainty increases with product complexity. As the number of components, functions, and interfaces increases (complexity), the ability of the design group to measure overall performance with certainty tends to decrease. Underestimation of product and process complexity by project planners exacerbates the issue (Conrow 1997). Factors influencing product complexity include:

a. Design Requirements Challenge Level. Design requirements can be more or less challenging to achieve. The level of challenge stems from several factors, including: the proximity of the design goals to state-of-the-art performance envelope boundaries; complex interrelationships between the specifications, such that increasing one decreases another; design freedom, in terms of requirements to integrate off-the-shelf components or utilize prior standards; hostile usage environments; ability to decompose and allocate system-level requirements to components; etc. Drivers of the challenge level include:

a1. Requirements Quality, Simplicity, and Stability. Much of the possibility of awareness of the design requirements challenge level results from having simple, stable, unambiguous requirements early on in a project. Design requirements are sometimes of poor quality themselves, particularly by being poorly defined or difficult to validate. Easily measurable requirements increase ability to understand the actual quality level of a design, thereby decreasing performance uncertainty. Sometimes the most challenging requirements to meet are the unknown or fuzzy ones, or ones that keep changing. "Creeping elegance" and other internal sources of change imparted to the design criteria also influence stability. Thus, simple and stable requirements can decrease the requirements challenge level.

a2. Requirements Negotiability. The challenge level can also be significantly reduced in key, difficult to meet areas if the requirements are negotiable. Sometimes requirements are set arbitrarily, and the extra effort needed to achieve them is not justified by the necessary time or cost.

a3. Product and Process Novelty. The challenge level is increased by *product and process novelty*. Difficult or ambiguous goals seem even more challenging when one is unsure of the exact means by which to approach them, and this is more likely when developing an unprecedented product or using a novel process. Product and process novelty effectively reduces the experience level of the work force, thereby debilitating their capabilities to identify important issues proactively and to have a system-level understanding of the effects of changes.

b. Robust Decomposition (Modularity). Product complexity and its resulting uncertainty can be reduced if the product architecture is intelligently decomposed into simpler subsystems. Many authors have written about this process (Alexander 1964; Baldwin 1999; Clausing 1994; Kusiak 1993; Pimmler 1994; Rechtin 1991; Sanchez 1997; Ulrich 1995). The general approach is to cluster system components into "chunks," with most interfaces existing between components within a given chunk and minimal interfacing between chunks. These chunks represent appropriate subsystems that can be developed relatively independently so long as the interfaces that remain with other chunks are carefully managed. One such management strategy, robust design, involves making the chunks as insensitive to changes in the designs of other chunks as possible. A well partitioned system architecture may also provide flexibility in terms of performance level modification ("killable chunks"), should performance need to be enhanced or depleted to satisfy market, development cost, technology, or schedule concerns.

1.4 Distribution of Risk Across System

If subsystem performance risks can be distributed such that most of the subsystems are of low risk and a relatively very few have higher risks, design performance uncertainty can be greatly reduced (Reinertsen 1997; Smith 1991, p. 107). For example, it is preferable to have four subsystems each with a 99% chance of success and one subsystem with a 90% chance of success—yielding a system success rate of 86%—than five subsystems each with a 95% chance of success (system success rate of 77%).

1.5 Technology Uncertainty

Performance uncertainty increases with *technology uncertainty*. Certain technologies are often key to delivering certain dimensions of product performance or design quality. Due to space constraints, technology uncertainty is not detailed in this paper.

1.6 Schedule Uncertainty

Since delivery timing is an aspect of overall product performance, *schedule uncertainty* contributes to performance uncertainty through that channel. (Browning 1998b) presents a framework describing sources of schedule uncertainty.

1.7 Development Cost Uncertainty

Product price, a dimension of overall product performance, is a function of, among other things, product development cost. Development cost uncertainty therefore causes product performance uncertainty. Due to space constraints, development cost uncertainty is not detailed in this paper.

CONSEQUENCES OF PERFORMANCE UNCERTAINTY AND SHORTFALLS

Consequences of performance shortfalls include:

- Additional design iterations, which increase schedule and development cost
- Late-breaking performance improvement initiatives required for specific areas
- Customer or market rejects product (reduced demand)
- Reputation of firm suffers
- Future contracts and sales jeopardized
- Other development, manufacturing, distribution, and support cost and schedule

impacts

In addition, the following are consequences of performance uncertainty itself in the design process:

- Required investment in backup plans
- Inefficient resource allocation throughout the design process (wasted time and money)
- Development cost and schedule uncertainty
- Inability to prioritize activities or interfaces
- Inability to make early guarantees to market
- “Fuzzy” design decision making; conservatism; misguided trade studies and analyses
- Much higher required investment in project planning and management

The key approach to managing uncertainty in the product design process is to purchase uncertainty reduction in appropriate areas at appropriate times. Bracing for the consequences of uncertainty requires flexibility. Flexibility has a price, and project planners and controllers should seek to ensure that the price is commensurate with the return in risk mitigation.

The design process has been described as a funneling procedure (Clark 1993; Pugh 1991). Design options and possibilities are removed until the remaining set hopefully satisfies the requirements. However, this procedure often requires taking a step backwards from time to time when decisions constrain the design (i.e., eliminate flexibility) too rapidly (Krishnan 1997). It is in this context that Ward *et al*. (1995) describe Toyota’s set-based design process as one where decisions are delayed as long as possible, thereby maintaining maximum design flexibility. (But maximum flexibility may not always be cost-effective.) Retaining as much design flexibility as possible, often simply by not eliminating options prematurely, can decrease performance risk. On the other hand, not making the appropriate decisions when they do become necessary increases the risk that the design will not converge within an acceptable time frame. The most appropriate timing of uncertainty reduction effort in the design process is a subject for future research.

CONCLUSION

This paper has addressed the sources of performance risk in complex system development programs. *Risk stems from uncertainty and consequences*. A key step in understanding risk consists of identifying the sources of uncertainty and deleterious consequences, the risk drivers. Then, it is useful to explore the relationships between sources of uncertainty: which ones cause or contribute to the others? These relationships are used in this paper to construct a causal framework. While the framework is based upon research and observations, some of the connections displayed are stronger than others. Some of the connections have been established through empirical research, whereas others to this point depend on theory, experience, and observation.

Thus, the framework should be viewed as a working model. Its primary purpose at this point is to direct further research and model development from a systems perspective. However, practicing managers will hopefully find the approach taken useful, allowing them to consider their options in new ways. A more advanced awareness of risk drivers improves *risk assessment* (attempting to quantify the probability of occurrence for each of several uncertain outcomes), *risk analysis* (examining the change in outcomes with the modification of risk drivers), and enlightened, proactive *risk management* policies and actions for the product development process.

REFERENCES

AFSC/AFLC (1988) "Software Risk Abatement" Air Force Systems Command/Air Force Logistics Command, Pamphlet 800-45.

Alexander, Christopher (1964) *Notes on the Synthesis of Form*, Cambridge, MA: Harvard University Press.

Alford, Mack (1994) "Issue Driven Design: Planning and Managing Design Iterations" *Proceedings of the Fourth Annual International Symposium of NCOSE*, San Jose, CA, Aug. 10-12.

Baldwin, Carliss Y. and Kim B. Clark (1999) *Design Rules: The Power of Modularity*, Boston, MA: M.I.T. Press (?).

Bell, Thomas E. (1987) "Performance Engineering: Doing it 'Later' on Large Projects" *CMG Transactions* (Winter): 75-81.

Blanchard, Benjamin S. and Wolter J. Fabrycky (1990) *Systems Engineering and Analysis*, Second Edition, Englewood Cliffs, NJ: Prentice Hall.

Boehm, Barry W. (1989) *Software Risk Management*, Washington, D.C.: IEEE Computer Society Press.

Brekka, Lawrence T. and George J. Vlay (1994) "Integrated Risk Management—Procedures and Tools" *Proceedings of the Fourth Annual Symposium of NCOSE*, San Jose, CA, .

Browning, Tyson Rodgers (1998a) *Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development* Unpublished Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA.

Browning, Tyson R. (1998b) "Sources of Schedule Risk in Complex System Development" *Proceedings of the Eighth Annual International Symposium of INCOSE*, Vancouver, July 26-30.

Clark, Kim B. and Steven C. Wheelwright (1993) *Managing New Product and Process Development*, New York: Free Press.

Clausing, Donald (1994) *Total Quality Development: A Step-by-Step Guide to World-Class Concurrent Engineering*, New York: ASME Press.

Conrow, Edmund H. and Patricia S. Shishido (1997) "Implementing Risk Management on Software

Intensive Projects" *IEEE Software* (May/June): 83-89.

Cooper, Kenneth G. (1993a) "The Rework Cycle: Benchmarks for the Project Manager" *Project Management Journal* **24**(1).

Cooper, Kenneth G. (1993b) "The Rework Cycle: How it *Really* Works… And Reworks…" *PMNETwork* (Feb.).

Cooper, Kenneth G. (1993c) "The Rework Cycle: Why Projects are Mismanaged" *PMNETwork* (Feb.).

DoD (1992) *Systems Engineering/Configuration Management Life Cycle Application* DoD, Military Handbook MIL-HDBK-499-3.

DoD (1994) *MIL-STD-499B Systems Engineering (Draft)* DoD, Military Standard Specifications MIL-STD-499B.

Draves, Joseph Paul (1993) *A Risk Management Process for Department of Defense Acquisition Programs Operating in an Integrated Product Development Environment* Unpublished Master's Thesis (Eng.), California State University, Long Beach, Long Beach, CA.

DSMC (1983) "Risk Assessment Techniques: Methods Overview & Technique Descriptions" (Ch. 4 and App. F) *Defense Systems Management College Handbook*. DSMC, iv-1—iv-25, F-1—F-13.

DSMC (1990) *Systems Engineering Management Guide*, : Defense Systems Management College.

DSMC (1998) *Risk Management Guide for DoD Acquisition*, Ft. Belvoir, VA: Defense Systems Management College Press.

Ha, Albert Y. and Evan L. Porteus (1995) "Optimal Timing of Reviews in Concurrent Design for Manufacturability" *Management Science* **41**(9): 1431-1447.

Hall, Elaine M. (1998) *Managing Risk: Methods for Software Systems Development*, Reading, MA: Addison-Wesley.

Hauser, John R. and Don Clausing (1988) "The House of Quality" *Harvard Business Review* **66**(3): 63-73.

Hoy, Kirk L. and David G. Hudak (1996) "Advances in Quantifying Schedule/Technical Risk" *Estimator* (Summer): 25-35.

Justice, Richard (1996) "Risk in the F-22 Program: A Defense Science Board Task Force Analyzes F-22 Concurrency and Risk" *Program Manager* (Jul.-Aug.): 68-74.

Kline, Stephen J. (1985) "Innovation is not a Linear Process" *Research Management* (Jul.-Aug.): 36-45.

Krishnan, Viswanathan, Steven D. Eppinger, and Daniel E. Whitney (1997) "Simplifying Iterations in Cross-Functional Design Decision Making" *Journal of Mechanical Design* **119**(4): 485-493.

Kusiak, Andrew and Juite Wang (1993) "Decomposition of the Design Process" *Journal of Mechanical Design* **115**(December): 687-695.

Leong, Alan and Robert P. Smith (1996) "Is Design Quality a Function of Time?" *Proceedings of the IEEE International Engineering Management Conference*, Vancouver, Aug. 18-20.

McIlroy, John (1997) (personal communication).

Michaels, Jack V. (1996) *Technical Risk Management*, Upper Saddle River, NJ: Prentice Hall PTR.

Pimmler, Thomas U. and Steven D. Eppinger (1994) "Integration Analysis of Product Decompositions" *Proceedings of the ASME Sixth International Conference on Design Theory and Methodology*, Minneapolis, MN, Sept.

Pugh, Stuart (1991) *Total Design: Integrated Methods for Successful Product Engineering*, Reading, MA: Addison-Wesley.

Ramachandran, Natarajan, Noshir A. Langrana, Louis I. Steinberg, and Vakram R. Jamalabad (1992) "Initial Design Strategies for Iterative Design" *Research in Engineering Design* **4**: 159-169.

Rechtin, Eberhardt (1991) *Systems Architecting: Creating & Building Complex Systems*, Englewood Cliffs, NJ: P T R Prentice Hall.

Reinertsen, Donald G. (1997) *Managing the Design Factory: A Product Developer's Toolkit*, New York: The Free Press.

Roberts, Barney B. and Ronald C. Winterlin (1996) "Integrated Risk Assessment: A Case Study" *Proceedings of the Sixth Annual International Symposium of INCOSE*, Boston, July 7-11.

Safoutin, Michael J. and Robert P. Smith (1996) "The Iterative Component of Design" *Proceedings of the IEEE International Engineering Management Conference*, Vancouver, Aug. 18-20.

Sanchez, Ron and Joseph T. Mahoney (1997) "Modularity, Flexibility, and Knowledge Management in Product and Organization Design" *IEEE Engineering Management Review* (Winter): 50-61.

Sarbacker, Shawn D. and Kosuke Ishii (1997) "A Framework for Evaluating Risk in Innovative Product Development" *Proceedings of the ASME Design Engineering Technical Conference (DETC)*, Sacramento, CA, Sept. 14-17.

SEI (1996) *Continuous Risk Management Guidebook*, Pittsburgh.

Shishko, Robert and Edward J. Jorgensen (1996) "Evaluation of Risk Management Strategies for a Low-Cost, High-Risk Project" *Proceedings of the Sixth Annual International Symposium of INCOSE*, Boston, July 7-11.

Shishko, Robert et al. (1995) *NASA Systems Engineering Handbook*, : NASA.

Smith, Preston G. and Donald G. Reinertsen (1991) *Developing Products in Half the Time*, New York: Van Nostrand Reinhold.

Smith, Robert P. and Primanata Tjandra (1997) "Experimental Observation of Iteration in

Engineering Design" University of Washington Dept. of Industrial Engineering, Working Paper .

Souder, William E. and David Bethay (1993) "The Risk Pyramid for New Product Development: An Application to Complex Aerospace Hardware" *Journal of Product Innovation Management* **10**: 181-194.

Ulrich, Karl T. and Steven D. Eppinger (1995) *Product Design and Development,* New York: McGraw-Hill, Inc.

Ward, Allen, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek II (1995) "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster" *Sloan Management Review* (Spring): 43-61.

Whitney, Daniel E. (1990) "Designing the Design Process" *Research in Engineering Design* **2**: 3-13.

AUTHOR'S BIOGRAPHY

Tyson Browning holds the position of Specialist Senior working on advanced process integration at Lockheed Martin Tactical Aircraft Systems in Ft. Worth, Texas. His current research and implementation activities focus on designing complex system product development processes for minimal cost, schedule, and performance risk; developing improved approaches to process modeling; and communicating the values of process definition for project planning, execution, and control. Formerly, Dr. Browning worked with the Product Development Focus Team in the Lean Aerospace Initiative <http://lean.mit.edu/public> at Massachusetts Institute of Technology. He has previous work experience at Honeywell, Inc. and at Los Alamos National Laboratory. He holds a Ph.D. in Technology, Management, and Policy (interdisciplinary systems engineering and technology management), a S.M. in Aeronautics and Astronautics, and a S.M. in Technology and Policy from M.I.T. and a B.S. in Engineering Physics from Abilene Christian University.