Sources of Schedule Risk in Complex System Development

Tyson R. Browning

Lockheed Martin Corporation, P.O. Box 748, M2 2222, *Fort Worth*, TX 76101 (*e-mail*: *tyson@alum.mit.edu*)

Received March 3, 1999; accepted May 13, 1999

ABSTRACT

Schedule risk is an important category of risk in complex system product development. This paper presents a framework that facilitates understanding schedule risk from a systems perspective. Research findings from literature and a Delphi-type survey of experienced product development managers and system engineers at a major aerospace company are synthesized into a framework characterizing sources of schedule uncertainty. The framework includes not only key uncertainty drivers but also the hypothesized or theorized relationships between them. Since risk is more than just uncertainty, consequences of schedule overruns and of schedule uncertainty itself are also discussed. This research contributes a more comprehensive, systems view to the studies of product development and risk management and to the practice of both in industry. The paper also examines potential paths for future research. © 1999, John Wiley & Sons, Inc. Syst Eng 3: 129–142, 1999

1. INTRODUCTION

Complex system product development inevitably involves risk. A number of authors have documented the importance of risk management to successful product development endeavors [e.g., Blanchard, 1997; Boehm, 1989; Conrow, 1997; Draves, 1993; DSMC, 1998; Hall, 1998; Reinertsen, 1997; SEI, 1996; Smith and Reinertsen, 1991]. An important category of risk in this context is schedule risk. This paper presents a framework that facilitates understanding schedule risk from a systems perspective. Bettering our understanding of the sources of risk in the product development process is fundamental to improving it. This paper presents the results of identifying essential sources or "drivers" of schedule uncertainty, determining or proposing their relationships, and using these to derive a framework for considering approaches to mitigating schedule risk.

Risk stems from uncertainty surrounding potential future states and the consequences of those states should they occur. Schedule risk stems from uncertainty regarding the ability of the development process to deliver a quality product within a given schedule and the consequences of schedule overruns. An important step in managing risk consists of identifying the main

Contract grant sponsor: Lean Aerospace Initiative at Massachusetts Institute of Technology

contributors to uncertainty and deleterious consequences, the risk drivers. (Typically, this is addressed on specific projects though the use of general risk driver checklists.) However, it is also important to explore the relationships among the sources of uncertainty, asking questions such as "How do they affect each other?" and "Can more than one risk be mitigated through a single process change?" This paper uses the relationships among sources of uncertainty as the basis for a causal framework. Currently, the framework is based upon research and observations and should only be viewed as a working model. Nevertheless, practicing managers can find the approach useful, since it allows them to consider the impacts of their decisions 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.

Much of the relevant literature describes the risk management process as iterative, with built in feedbacks. Risk management actions serve as controls, limiting the effects of risk drivers. An effective risk management process requires continuous monitoring of project risks and effective control mechanisms for identifying and reacting to process instabilities. Without a systems view, however, many risk management actions serve only to push schedule risk into another category—such as cost or performance risk—rather than truly reducing overall risk.

The framework in this paper provides a systems perspective by illuminating cause and effect relationships among risk driver variables. Understanding these ties can help one pinpoint high leverage areas and avoid unwanted side effects. To illustrate, consider several footballs stuck in the outer branches of a dense tree. If the goal is to reduce overall schedule risk by making a change in one area of the project, without increasing risk in other project areas, this can be likened to trying to shake the branches of the tree to dislodge only one ball without causing the others to fall. One should think carefully about which branch to shake, and one who discerns the structure may realize that only a small tussle in just the right place will achieve the desired results.

This paper focuses on schedule risk at the complex system development project level. One could also examine risks from the perspective of a portfolio of projects or at any of the levels of activity breakdown within a project. The framework developed herein is designed to be useful at any level from the project level down, but not necessarily up: Other factors would then need to be included. However, with an understanding of the schedule risk for one project, such knowledge would figure into an informed analysis of multiproject risks for an enterprise. This paper also focuses on complex system development, where the identification and management of risk drivers is more challenging and valuable. However, the framework also applies to the development of "simple" products.

After a brief presentation of methodology in Section 2, Section 3 discusses the factors causing schedule risk—first, the sources of uncertainty and their relationships, and, second, the consequences of schedule overruns and of schedule uncertainty itself. The paper concludes with an examination of the implications for the practice of risk management and directions for further research.

2. METHODOLOGY

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

1. Consultation of literature, examining the factors others have identified as sources of schedule risk.
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.
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.

Stages 1 and 2 provided an initial list of schedule risk drivers. Before a new factor was added to the growing list, an effort was made to determine that the factor was indeed different and not a restatement of a previously identified factor. The stabilized list was used to build the survey in stage 3. Stage 3 served mostly to validate the list, to weight the importance of the factors, and to provide some information regarding causality. Stage 3 confirmed that almost all of the factors had been identified in stages 1 and 2. The survey used in stage 3 was not administered to a statistically significant number of respondents, so factor weightings are not included in this paper. However, the survey and a summary of responses is available in [Browning, 1998b]. (Collecting a significant amount of empirical data about each factor would constitute an interesting research project, as discussed at the end of the paper.)

With a list of uncertainty drivers—many of them unaddressed in previous studies—the next step involved constructing causal diagrams to represent their relationships. This process led to some refinement of the list of factors as it became clearer which factors influenced others. The resulting diagrams represent a strawman framework for understanding the causes of schedule risk in development projects. The framework does not purport to be finalized. It endeavors to be comprehensive at a reasonable level of abstraction, but it no doubt falls short in terms of what factors and relationships should ultimately be included. Certainly, there is opportunity for future extension and improvement. This paper will hopefully generate discussion along these lines. The following section describes the schedule risk framework resulting from this work.

3. SCHEDULE RISK

3.1. Definition of Schedule Risk

Schedule risk is the uncertainty associated with 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. From a mathematical perspective, a measure of schedule risk is the product of the probability of certain project durations and the consequences of each. Schedule uncertainty can be represented mathematically as variation from the expected duration. Schedule uncertainty drivers cause variation in the distribution of possible schedule duration outcomes. The wider the distribution of possible schedule durations, and the greater the probability of the deviant durations, the greater the schedule uncertainty, and—especially when consequences are proportional to deviation—the greater the schedule risk.

3.2. Causes of Schedule Uncertainty

Variation in the overall duration for a product development project can be attributed to a number of factors. The goal of this section is to explore the causes of schedule risk in terms of sources of schedule uncertainty. The framework herein should be viewed as a preliminary, working model. Its purpose is to stimulate thinking about schedule risk drivers, facilitating their identification and inclusion within new product development process models and management policies. The connections in the framework vary in strength in two ways: (1) Some factors are more influential than others, and (2) some of the indicated relationships have been established through empirical research, whereas others to this point depend on theory, observation, and experience.

Figure 1 provides an overview of the schedule risk driver framework in the form of a causal loop diagram.

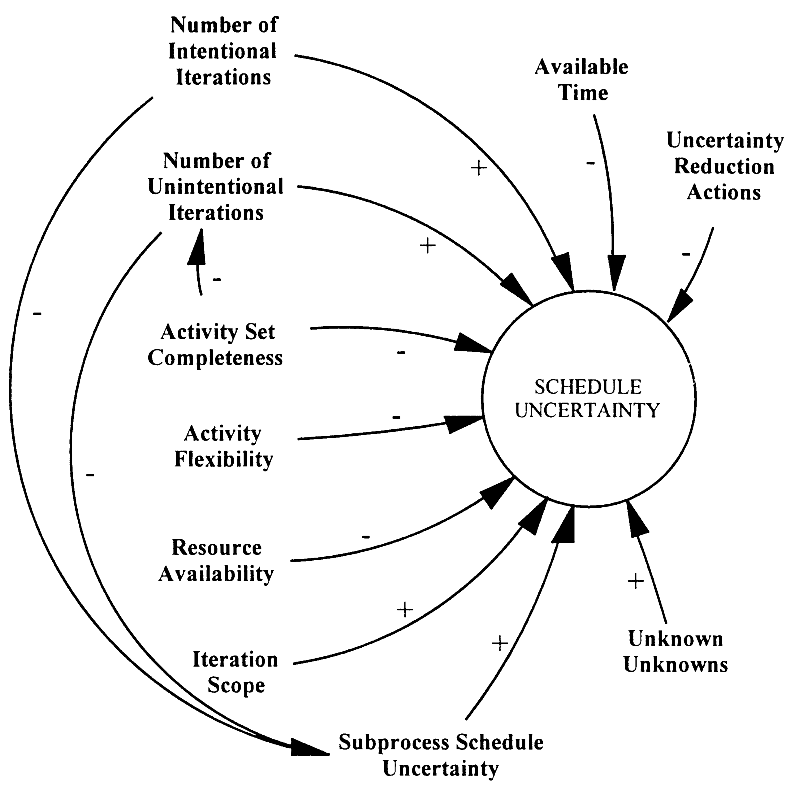


Figure 1 Categories of sources of schedule uncertainty.

A causal loop diagram shows cause and effect relationships among variables (represented by the directions of the arrows), under the assumption of "all else being equal." Some of the variables shown (and many others, added below) may not in fact represent direct causes but rather proxies for certain causes and/or effects. Also, note that in a dynamic model the distinction between cause and effect becomes blurred. For simplicity, Figure 1 does not include feedback. (For example, *Schedule Uncertainty* may precipitate corrective actions, which in turn have a variety of effects on the schedule uncertainty drivers.) The positive and negative signs accompanying most causal arrows indicate directionality, all else being equal. For example, having to perform additional iterations of product design increases the risk that the schedule will deviate from anticipated duration, as indicated by the "+" alongside the arrows from both "number of iterations" variables to schedule uncertainty. Similarly, as confidence about the completeness of the activity set constituting the product development process increases, schedule uncertainty decreases, all else being equal. The main drivers of schedule uncertainty and their causes are discussed in the following subsections.

3.2.1. Number of Intentional Iterations within the Process

Iteration of design activities is fundamental to complex product development processes in at least two ways:

1. Iterations are intentional (planned), allowing design processes containing interdependent activities to refine their designs and converge to a desirable solution.
2. Iterations are unintentional (unplanned), the result of new information arriving late in the process. This information alters assumptions and causes upstream activities to do rework.

As a conceptual construct, iteration accounts for design changes and rework. Osborne [1993] found that iteration typically accounted for an average of 30% (ranging from 13% to 70%) of total project effort (usually the unanticipated part) for the semiconductor development projects he studied at Intel. Iterations are a major contributor to schedule uncertainty. This effect is compounded by their occurrence along a process's critical path of activities. The number of intentional iterations is affected by a number of variables, shown in Figure 2 and discussed below.

a. Performance Uncertainty: Successive iterations are intended to move the design closer to desired targets [Bell, 1987; Eppinger et al., 1994; Singh et al., 1992; Smith and Eppinger, 1997b; Whitney, 1990]. Any assumption that performance increases monotonically with each iteration generally becomes more valid the higher the level of abstraction. However, some studies disprove the monotonicity assumption under certain circumstances [e.g., Cusumano and Selby, 1995; McDaniel, 1996].

b. Iteration Productivity: Iteration productivity is the effectiveness of each iteration in decreasing the performance gap by increasing actual design quality. The rate at which additional iterations close the performance gap is the productivity, quality, or gain of the iterations. If doing additional iterations within an allotted amount of time (by doing them faster) compromises the productivity of each iteration, the additional iterations may not be helpful.

c. Degree of Activity Coupling: Highly interdependent activities will converge more slowly to a multidimensional performance target than will sparsely coupled activities [Clark and Fujimoto, 1991; Eppinger, Nukala, and Whitney, 1997; Eppinger et al., 1994; Smith and Eppinger, 1997a, 1997b]. *Amount* of coupling refers to the relative number of dependencies among activities. *Tightness* of coupling refers to the

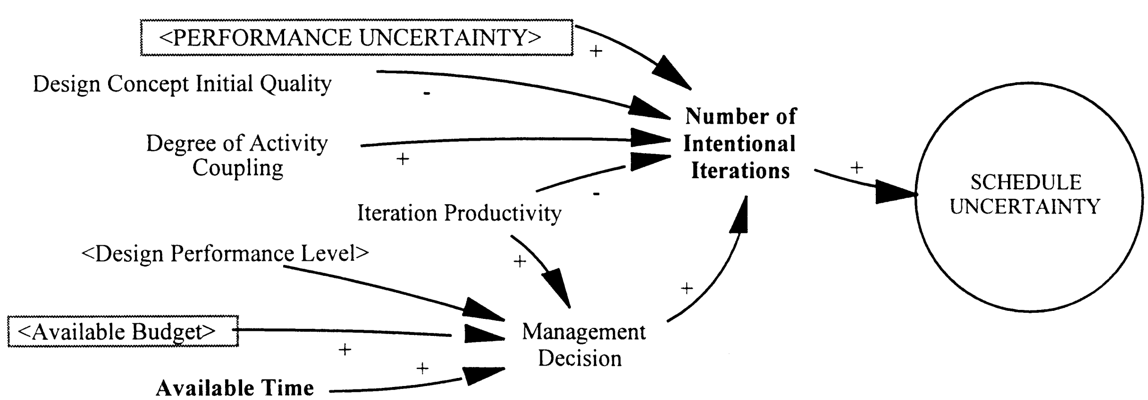


Figure 2 Factors affecting number of intentional iterations.

extent that the dependent activities rely on each other's outputs as their own inputs. Together, amount and tightness are represented as *degree* of activity coupling. A greater degree of coupling means more "chicken and egg" design problems[1] that must be solved iteratively. The ramifications of design coupling can be reduced to an extent through robust design practices [e.g., Clausing, 1994] and set-based design [e.g., Ward et al., 1995]. They can also be reduced through improved *activity sequencing quality* (Section 3.2.2b, below). Activity resequencing cannot remove coupling, but it can minimize its impact.

Complex systems typically exhibit coupling or interconnectedness among their components. Complex system architectures then imply complex development processes (coupled activities) and organizations (coupled teams). Complexity—operationalized as the number of components, the number of features per component, and the number of interactions between components—can be reduced through intelligent decomposition or partitioning into subsystems with minimal interaction [Alexander, 1964; Baldwin and Clark, 1999; Eppinger, 1997; Gulati and Eppinger, 1996; Rechtin, 1991]. The quality of this partitioning can have great impact on the coupling between activities and therefore the ability of the design process to converge rapidly to an acceptable solution.

d. Design Concept Initial Quality: The number of iterations required to converge on acceptable design parameter values is greatly dependent on the quality of the design concept—i.e., on the starting points from which the search and convergence algorithm of design work begins [Ramachandran et al., 1992].

e. Management Decision: Project planners and controllers decide on a number of intentional design iterations to do based on current design performance level, available time (Section 3.2.8), and available budget. The current performance level of the design, as determined through design evaluation, should explicitly influence management decisions regarding whether or not to do additional iterations. Often, the influence is only implicit.

Management decision also has a bearing on the general approach to managing intentional iterations—faster or fewer iterations [Smith and Eppinger, 1997a]. In general, fewer iterations can mean less design quality, or at least a greater chance that the design will not converge to the desired targets (performance risk). Since more iterations reduce the chance that the design will not converge (all else being equal), faster iterations can reduce performance risk. In both cases, however, the quality or productivity of each iteration is important. Fewer iterations might make sense if each one is of sufficient quality to ensure acceptable output. Faster iterations will only be advantageous if each activity can be accelerated while continuing to produce satisfactory outputs.

3.2.2. Number of Unintentional Iterations within the Process

The number of unintentional or unplanned iterations is affected by a number of variables, shown in Figure 3 and discussed below.

a. Performance Uncertainty: Performance uncertainty, representing an unanticipated performance shortfall, causes unwanted iterations. For instance, an unanticipated test failure may cause unplanned rework for several activities. Although performance uncertainty is usually addressed through intentional iterations, the iteration in this example is unintentional, the result of a realized process failure mode.

b. Activity Sequencing Quality: The activities in the design process can be sequenced based on their input and output requirements such that the need to send information backwards or upstream in the process and the scope of these feedbacks are minimized. Processes not organized on this basis are more likely to experience unintentional iteration simply because the right information is not available at the right place at the right time [Browning, 1998b; Eppinger et al., 1994; Smith and Eppinger, 1997b, 1998]. Therefore, activity sequencing quality not only affects the number of unintentional iterations, it also affects the number and scope of activities comprising any iterative cycle (iteration scope; see Section 3.2.6). Activity sequencing is constrained by:

- *Long lead time activities*
- *Resource availability* (*Section 3.2.5*)
- *Knowledge of activity content* (a function of *product and process novelty*)
- *Requirements quality, simplicity, and stability* (*see d.*)

Activity coupling can inadvertently increase iteration when downstream activities are brought upstream (increased concurrency). Enlightened activity overlapping requires considering the sensitivities of activities to changes in their inputs, the volatilities of their outputs, and the amount and frequency of their preliminary information exchange [Browning and Eppinger, 1998; Carrascosa, Eppinger, and Whitney, 1998; Krishnan, Eppinger, and Whitney, 1997; Loch and Terwiesch, 1998; Yassine, Chelst, and Falkenburg, 1998].

"Activities" include decisions as well. Just as analyses, trade studies, tests, etc. must be sequenced with an understanding of their information needs and products,

[1] Or design "circuits" [Stewart, 1981].

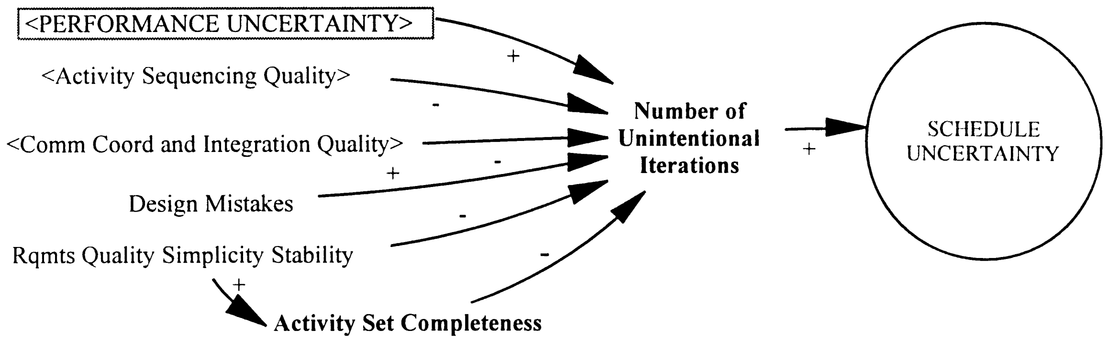


Figure 3 Factors affecting number of unintentional iterations.

decisions must also be arranged so that maximum information is available to make them and that their outcomes create minimum rework. (This includes decisions taking into account downstream requirements and constraints, such as design for manufacturing and assembly, etc.) Poor decision sequencing leads to reopening issues and a cascade of unintentional iteration.

c. Quality of Communication, Coordination, and Integration in the Process: Successful communication helps information be available at the right place, at the right time, and in the right format. Less than perfect communication contributes to finding out things "later," when the only option is to go back and redo something [Browning, 1998a; Christiansen, 1992; Hoedemaker, Blackburn, and Wassenhove, 1995; Morelli, Eppinger, and Gulati, 1995]. The synchronization of information exchange, decisions, and iterations across processes is important to ensure that all activities are working with the latest, most useful information. Thus, this source of schedule uncertainty includes aspects of the effectiveness of change management, configuration management, information management, role and responsibility assignments, relations with customers and suppliers, etc. In this vein, Clausing [1994] notes how design decisions must be progressively frozen for the design to converge. Clark and Fujimoto [1991: 240] discern that a condition for effective integration is making upstream decisions in a timely and "downstream friendly" fashion. Likewise, downstream activities should improve their ability to forecast upstream results and make time-risk tradeoffs. Organization modularity significantly impacts the ease of organizational integration [Browning, 1997; Sanchez and Mahoney, 1997]. Changes in leadership can also affect the quality of coordination within a process.

d. Design Specifications and Requirements' Quality, Simplicity, and Stability: Requirements are a major topic of interest to the systems engineering community because of their tremendous impact on many facets of projects. Gupta and Wilemon [1990] and Mello[2] [1997] found poor definition of requirements to be the number one cause of delays in new product development. Complex and/or equivocal requirements increase the likelihood that something will be missed on the first pass and cause rework later. Unstable requirements result in moving design targets; changing prioritizations result in new design information and rework. Vacillating requirements enable requirements "creep" and design concept instability. It is hard to converge on a poorly understood target or on one that keeps changing. Requirements can otherwise be deficient because of incompleteness, conflict, invalidity, or infeasibility. Incomplete requirements, once completed, are likely to create new information and rework. Ironing out conflicting requirements and discovering invalid or infeasible requirements often requires multiple, inintentional iterations.

While a closed set of simple requirements is unlikely for a novel, complex system (*product and process novelty*), appropriate effort can diminish complications [Smith and Reinertsen, 1991: 66–69]. Development efforts should also address the possibility of insuring that design decisions are robust against inevitable requirements changes.

Market uncertainty drives requirements quality, simplicity, and stability as well. The better the market (customer) and the regulatory environment (certification basis) are understood, the more likely the organization can concisely write and correctly interpret stable product requirements. Furthermore, *available time* can also impact requirements stability: The longer the duration of the design process, the greater the chance that customer needs will change during that time [Reinert-

[2]Mello quotes findings from the "Product Development Best Practices survey: Report of Findings," undertaken by PDC, Inc., March 1996.

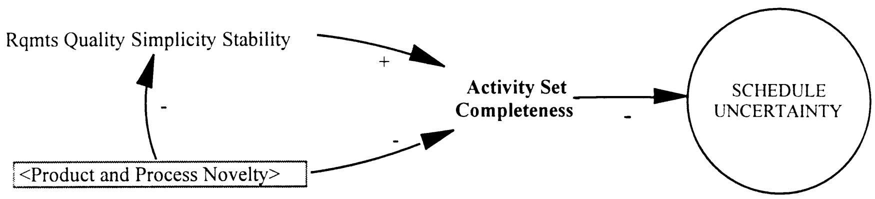


Figure 4 Factors affecting activity set completeness.

sen, 1997]. Hence, shorter cycle time projects can be less likely to experience requirements changes.

e. Design Mistakes: Every design process, especially an unprecedented one or one developing an unprecedented system (*product and process novelty*), will have its share of errors and mistakes [Rechtin, 1991: 129f].[3] Mistakes imply unintentional rework for upstream activities. Improved *communication*, *coordination*, and *integration quality* can minimize the impact of mistakes by helping to discover and correct them quickly.

f. Activity Set Completeness (Section 3.2.3): The information created by activities in the product development process should add value to the product design. However, if activities are missing, some of the necessary value will not be added. New activities, added late in the process to correct the situation, create new information and rework for other activities (unintentional iteration).

3.2.3. Activity Set Completeness

Schedule certainty cannot be achieved without a complete activity set. If the activities comprising a process are vaguely defined, something will be left out. Activity set knowledge is affected by the factors shown in Figure 4 and described below.

a. Design Specifications and Requirements' Quality, Simplicity, and Stability (Section 3.2.2d): The activity set is more difficult to settle if the requirements are unclear or unstable. Work statement growth, "creeping elegance," and "might as wells" create new activities, making the original plan less relevant and increasing schedule uncertainty.

b. Product and Process Novelty: Rechtin [1991: 139] lists "the job is more complex than we thought" as a major reason why schedules change (because activities must be added). New activities can cause unintentional iteration in a process.

3.2.4. Activity Flexibility

Activity flexibility is the extent to which activities can be rearranged or resequenced in the process should the need arise. The ability to adjust the schedule with ease because of unanticipated circumstances greatly reduces the probability that these events will prevent meeting planned schedule milestones [Boehm, 1989]. Activity flexibility is affected by the factors shown in Figure 5 and described below.

a. Long Lead Time Activities: Long lead time activities must begin early in order to finish on time. Therefore, they constrain activity flexibility.

b. Organization Inertia: An organization's lack of agility in adapting to modified process configurations (activity sequencing) constrains activity flexibility.

c. Knowledge of Activity Content: Activity flexibility should stem from an understanding of each activity's role. Each activity should exist to provide certain information. Understanding what information is needed and when helps illuminate the intelligent options for activity resequencing.

d. Resource Flexibility: If an activity can be accomplished by any of a number of people with varied skill sets, that gives planners more flexibility. Brooks [1995] emphasizes how activity cost/schedule elasticity is an important project management factor.

3.2.5. Resource Availability

Information, personnel (with appropriate experience, expertise, and leadership), facilities, and funding are necessary to execute a product development process. Furthermore, these resources must be at the right place at the right time. Resource availability should be predictable and stable. Bottlenecks resulting from resource scarcity increase activity duration and variance and diminish the advantages of improved activity sequencing. Individual activities differ in their sensitivities to various kinds of resource shortfalls. Resource availability is affected by the factors shown in Figure 6 and described below:

a. Knowledge of Activity Content: Knowledge of activity content is essential to allocate the right resources at the right time. Product and process novelty diminishes activity content knowledge and complicates the resource allocation problem.

b. Available Budget: Processes and their activities require stable and predictable funding profiles to reduce schedule uncertainty. In both military and commercial worlds, the availability of a funding profile often dic-

[3] Petroski [1985, 1994] provides case studies of design failures.

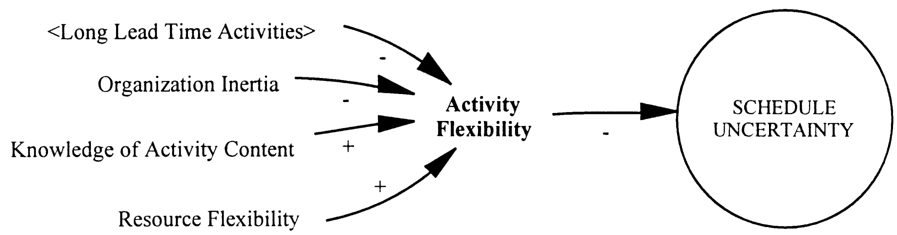


Figure 5 Factors affecting activity flexibility.

tates the time frame for product development [McNutt, 1998]. To minimize schedule risk, funding profiles could be made more flexible. Indeed, Rechtin [1991: 139] notes "unpredictable funding" as a major reason for schedule changes.

c. Quality of Communication, Coordination, and Integration (Section 3.2.2c): This factor affects the availability of information, an important resource for the activities in a process.

d. Project Priority: A project's priority relative to other projects influences resource allocation and availability. Low priority projects are constantly held up for lack of resources [Adler et al., 1995].

e. Resource Constraints: In the short term, resources may be limited in ways besides budget. There are maximum ramp up/down rates, limits on total resources available, shortages of material, development process capacity limits [MRT, 1997; Reinertsen, 1997], and holdups by labor issues.

3.2.6. Iteration Scope

The scope or amount of effort involved (number of activities involved and potential duration) in iterations is an important contributor to schedule variance. Several authors have noted the need to execute many short, fast iterations instead of a few long ones [Clark and Fujimoto, 1991; Eisenhardt and Tabrizi, 1995; Singh, et al., 1992]. Iteration scope is affected by the nature of the process configuration. If a process contains long feedback loops,[4] then it is more likely to experience large scope iteration.

Often, long feedback loops can be shortened or eliminated though improved activity sequencing within the process. In general, the greater the number of sequential activities required to iterate, the greater the scope and duration of the iteration. For example, manufacturing activities, which traditionally occur downstream in the product development process, tend to provide feedback that causes changes in upstream design activities. The scope of these iterations tends to be relatively large. However, if the activities are resequenced so that manufacturing can provide this feedback earlier, the scope of the iterative cycle is reduced.

3.2.7. Subprocess Schedule Uncertainty

Activities and subprocesses within a project are of uncertain duration. Variation in their duration is driven by the same factors causing schedule uncertainty in the overall development process. Thus, to the extent that lower level variations propagate uncertainty to a higher level process, determining schedule risk is a recursive, bottom-up, synthesizing process. Rolled up schedule uncertainty can include aspects such as: supplier ability to perform in terms of completing design work on schedule, on-time drawing releases and their quality levels, etc. The greater the size and complexity of the system development, the greater the number of levels to roll up and the greater the chance that something will cause variance. To make matters worse in these cases, negative outcomes are not canceled out by positive ones: Some activities will finish late, while activities will seldom if ever finish early. (When activities run ahead of schedule, they tend to relax and let work expand to fill available time [Goldratt, 1997].) Indeed, much of the potential to remove uncertainty from large processes lies in monitoring the status of constituent activities and reacting to their outcomes appropriately from a system perspective.

For individual activities, resource availability, resource stability, and sensitivity to resource fluctuations drive schedule duration and variance. Sometimes these factors are determined by a single person for an activity. Their dedication to the activity to the exclusion of distractions constitutes their availability and stability. If forced to be multitasking, the person will attend to the activity with lower productivity. Some activities are actually important decisions, which, if deferred or held up, can have a significant delaying effect on the process. (This is a driver of empowerment, or decision-making at the lowest level.)

Adler et al. [1995] determine that the variation in the duration of individual activities is a less significant source of schedule variance than other sources (such as

[4]That is, downstream activities providing information to sensitive, upstream activities; e.g., late, downstream manufacturing analysis of a design.

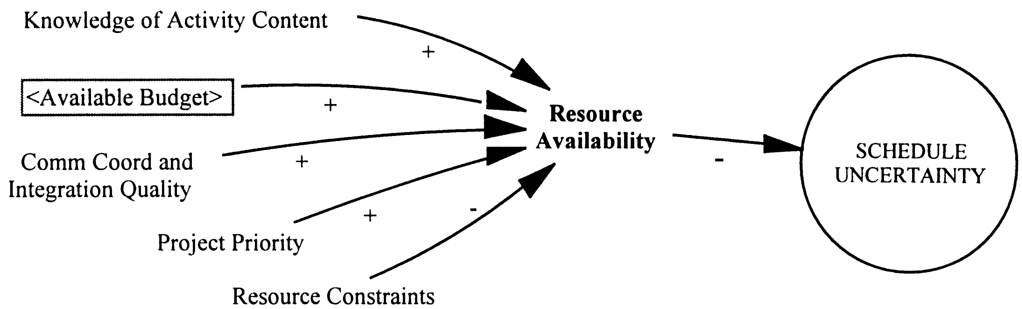


Figure 6 Factors affecting resource availability.

number of iterations). However, the impact of an activity or subprocess on the overall process is a function of the percentage of the process represented by that activity or subprocess. If an activity's duration is 90% of a process's duration, then the variance in the schedule of the activity can dominate. Decomposing processes into the smallest possible blocks of discrete activities (subprocesses) ameliorates this effect [Altus, Kroo, and Gage, 1995; Kusiak and Wang, 1993; Michelena and Papalambros, 1995].

Because of learning curve and setup time effects, activity durations can decrease when activities are repeated (number of intentional and unintentional iterations) [Browning and Eppinger, 1998]. However, if a large number of iterations are anticipated by development planners, a project must guard against a loss of productivity or gain per iteration resulting from a "we'll get to it next time" mentality.

3.2.8. Available Time

Available time is the planned duration of the project. The quality of this plan has a bearing on the feasibility of the schedule and therefore on the uncertainty surrounding it [Conrow and Shishido, 1997]. As available time increases, all else being equal, the uncertainty surrounding the ability of the project to produce a low performance risk product during that time increases.

3.2.9. Unknown Unknowns

The risk drivers addressed in this research represent "known unknowns." That is, they are factors known to cause schedule uncertainty in projects. This framework includes a notion of "unknown unknowns" as well, which, from the perspective of a regression analysis, would represent all schedule variance otherwise unaccounted for.

No framework or plan will be able to anticipate every possible factor affecting schedule uncertainty. Plus, as significant drivers of schedule uncertainty are addressed by improved management policies, new factors will become significant. Therefore, the framework at least acknowledges the influence of other, heretofore unaccounted for schedule uncertainty drivers. A risk management process requires an ongoing effort to update risk driver awareness based on lessons learned from a variety of projects and situations.

3.2.10. Uncertainty Reduction Actions

Actions by management to identify, assess, analyze, and mitigate schedule uncertainty play an important role in risk reduction. Schedule uncertainty is ultimately determined by the contributions of the factors mentioned above and the reduction of this uncertainty by specific actions. Many risk reduction actions become obvious based on the drivers contributing to schedule uncertainty: Efforts to temper the effect of risk drivers reduce the amount of risk to manage. In terms of general, project risk mitigation, Boehm [1989] suggests the following actions:

- Look for emerging win-lose or lose-lose conditions (these diminish morale and motivation to complete activities).
- Use checklists to ensure items are not overlooked.
- Set control limits and monitor the process.
- Continuously monitor cost, schedule, and performance.
- Provide advance training to reduce mistakes and increase productivity and coordination quality.
- Ensure resource availability.
- Invest in process improvement.
- Develop detailed interface understanding.
- Work to facilitate activity flexibility.

An important uncertainty reduction action is good planning. Although schedule planning is not discussed as a separate driver of schedule uncertainty, good planning—which impacts and is impacted by many of the factors mentioned in this section—is crucial to reducing schedule uncertainty. Planning may even illuminate more of the uncertainty than planners are comfortable with!

3.2.11. Putting It Together

Using Figure 1 and adding the factors discussed above, the complete causal framework for schedule uncertainty is shown in Figure 7. Variables in "<variable>" notation represent names used in more than one place. Most of these, such as "<Long Lead Time Activities>," are used in two or more places to reduce the clutter caused by crossing arrows. Others, such as "<Performance Uncertainty>," are also enclosed in boxes. These represent factors directly involved in other risk categories to be discussed in a future paper.

3.3. Additional Notes on Schedule Uncertainty

Underlying the schedule uncertainty drivers are several project factors: project scale, project scope, and *product and process novelty*. Project scope and scale act as multipliers, increasing the magnitude of the effects of many of the factors. They pertain not only to the scope and scale of the product (system architecture) but also to the scope and scale of the project, in terms of number of development sites, number of deliverables, and other contractual characteristics. Product and process novelty are explicitly shown as drivers of *requirements quality*, *simplicity*, *and stability*; *activity set completeness*; *design mistakes*; and *knowledge of activity content*. They probably influence several other variables as well. Osborne [1993] hypothesizes that product development time is driven by product and process novelty. Many authors, including Smith and Reinertsen [1991], have noted how an extremely large and novel "megaproject" creates much more uncertainty than several smaller step projects.

Some additional confusion can be avoided by distinguishing schedule "velocity" from schedule "acceleration." The former refers to the planned rate at which the process will move towards its objectives, while the latter refers to efforts to change that rate after the process is underway, either speeding it up ("crashing") or slowing it down. Setting schedule velocity at a higher rate is the premise whereby much of the current literature on cycle time reduction bases its claims of reduced development costs. However, accelerating an ongoing project increases development costs drastically. Schedule acceleration also has the following impacts: It increases the likelihood of mistakes; it increases the degree of concurrency (the amount of activity overlap); it raises the likelihood of activity omission (increasing performance risk); and it diminishes communication, coordination, and integration quality [Graves, 1989]. On the positive side, schedule acceleration can have the effect of weeding out less important activities—although the method whereby this is done is less likely to be well thought out under the circumstances.

Planners typically form a schedule based on their experiences with similar projects and expectations about the unique aspects of the project at hand. This "expected duration" is essentially a guess—often an optimistic one—or else it is based on an available funding profile or an identified "window of opportunity" in the market. Yet, it is no surprise that schedules overrun or that tremendous cost must be expended to remain on schedule. Part of the problem is that many schedules for complex system development are dictated from the "top down." While work is decomposed into activities in this fashion, the schedule for the work should be validated from the "bottom up" to reduce schedule uncertainty.

3.4. Consequences of Schedule Uncertainty and Overruns

Uncertainty is only part of the risk equation. Schedule risk also depends on the consequences of various duration outcomes. Consequences of schedule overruns include:

- Possible breach of contractual arrangements
- Failure to hit crucial windows of market opportunity [Smith and Reinertsen, 1991: 209–211]
- Design and product obsolescence
- Failure to provide product performance in the area of delivery timing—impact depends on value of timing to customer
- Need to change the schedule rate (acceleration or deceleration)
- Possible extreme reactions—e.g., searching for substitute parts and vendors to meet replanned schedule requirements or creating new activities to evaluate the appropriateness of substitute, COTS parts [Shishko and Jorgensen, 1996: 716]—which are usually quite expensive and may kill a project
- Development cost increase, perhaps to the point of invalidating the business case upon which the project was justified

Other consequences or impacts are also possible. In addition, there are also consequences of uncertainty itself that exist regardless of the outcome:

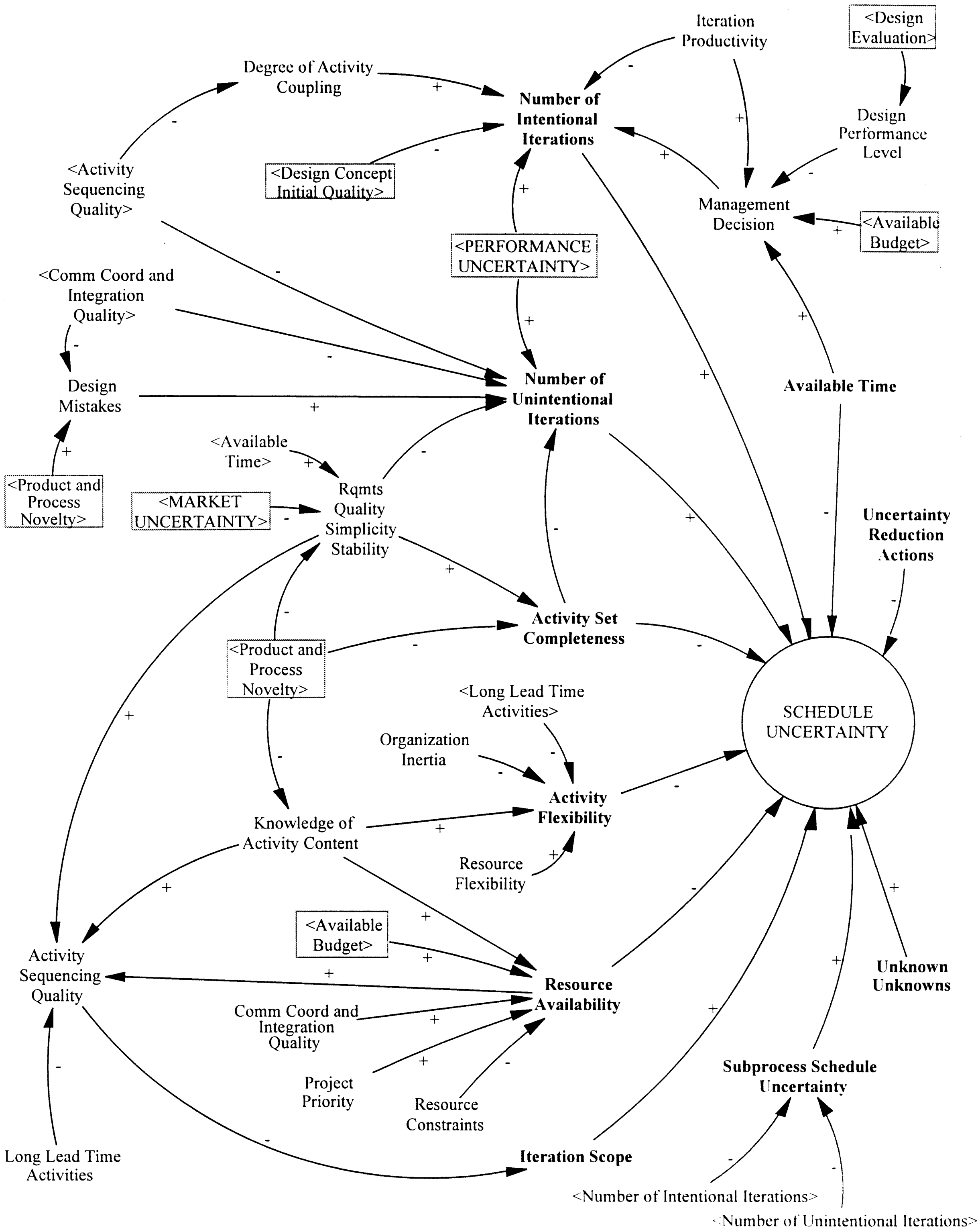


Figure 7 Causal framework of factors contributing to schedule uncertainty.

- Time is wasted as activities wait for each other
- Uncertainty requires flexibility, which means maintaining large resource reserves
- Project is unable to make firm deadline commitments
- Project viewed as "high risk" by upper management and customers
- Decision making is "fuzzy"; indecisiveness reigns because of the surrounding uncertainty; people are afraid their decisions may be invalidated later

Since complex system product development is inherently an uncertain process, many of these impacts are unavoidable. Nevertheless, the impacts can be diminished by project planning and control actions designed to reduce uncertainty.

Regarding delivery timing and the impact of a schedule overrun or extension: The firmness of the "need date" has a bearing on the consequences of unacceptable project duration outcomes. An effort should be made to explore the value to the customer or the market of having the system at a certain time. As with any performance feature, sometimes the customer requires help to realize the advantages in their own processes of faster or more certain product delivery. A need date is sometimes flexible or negotiable. Delivery timing might be traded off against other aspects of performance. In the commercial world, the deadline is based on the forecast window of profitability for the product. Usually, this means "as soon as possible," although there are notable exceptions (such as when the market/customer is not ready or when the product will prematurely cannibalize other product lines). For military deliverables, the need date is affected by the status of the threat against which it will be employed and political sensitivities. If the market will not pay for faster or more certain product development cycle times, then efforts along these lines may have to be justified for other reasons (if at all).

4. CONCLUSION

This paper has explored the drivers of schedule risk in product development projects. The approach synthesized fragments of existing theory and knowledge from several different disciplines into a new framework for understanding development project schedule risks. Effort was made to create a coherent definition of schedule risk, its causes, and their relationships. Risk contains components of uncertainty regarding outcomes and consequences of unwanted outcomes. Drivers of uncertainty and their relationships were highlighted using a causal framework. Many of the factors and relationships are based on empirical studies, while many others represent theories or hypotheses. The significance of the factors and relationships in a given project is largely a function of the organization, incentives, methods, competencies, and policies of the development firm(s).

Industrial practitioners can benefit from the framework by using it to identify risk areas and build lower risk project plans. The framework provides researchers with an enhanced ability to formulate focused research questions around these issues. While this paper seeks to motivate a set of problems and issues, much analytical and empirical work remains.

This exploratory study has several implications. First, uncertainty, in the form of instability or variance from the plan, drives risk. Identifying sources of uncertainty is useful in recommending new directions for risk management policies, methodologies, and tools. In particular, product development process models will benefit by realizing new variables to account for. This leads to a second implication: several important sources of uncertainty are inadequately addressed by current models and techniques. Some especially important links for future research and model building include the links between iterations and cost, schedule, and performance uncertainty, and the effects of cost and schedule uncertainty on performance uncertainty. Third, a systems view of the relationships between the drivers of uncertainty is helpful. Risk management policies should not create more problems or costs than they solve or prevent, and a knowledge of the highest leverage points in the process requires a systems view.

This study clarifies several rich areas for future research. Additional studies in a variety of industrial projects are needed to sharpen the framework and to enable the derivation of further conclusions. First, empirical studies are needed to verify pieces of the framework—the significance of the factors and the strength of the dependencies. Regression and ANOVA analysis could be used to determine the contributions of various drivers [e.g., Wohlin, Xie, and Ahlgren, 1995]. Second, both the empirical work and industrial practice would benefit from the determination of a real-time metric for each driver, which might provide some leading indicators of problematic outcomes. This would be a step towards a model to quantify project risks from a systems perspective. Third, quantitative frameworks could be used to explore possible tradeoffs between the various risk categories. This would motivate the development of improved decision support methods and models.

ACKNOWLEDGMENTS

Support for this research from the Lean Aerospace Initiative at the Massachusetts Institute of Technology is gratefully acknowledged. The author thanks the managers and system engineers who contributed their observations, experiences, and comments to this paper. Charles Boppe, David Grose, Ross McNutt, and Eric Rebentisch also provided helpful suggestions. This paper is based on material from an unpublished Ph.D. thesis [Browning, 1998b: Chapter 3] and a prior INCOSE conference paper [Browning, 1998c].

REFERENCES

P.S. Adler, A. Mandelbaum, V. Nguyen, and E. Schwerer, From project to process management: An empirically-based framework for analyzing product development time, Manage Sci 41 (1995), 458–484.

C. Alexander, Notes on the synthesis of form, Harvard University Press, Cambridge, MA, 1964.

S.S. Altus, I.M. Kroo, and P.J. Gage, A genetic algorithm for scheduling and decomposition of multidisciplinary design problems, Proc 21st ASME Des Automat Conf, Boston, September 1995.

C.Y. Baldwin and K.B. Clark, Design rules: The power of modularity, MIT Press, Cambridge, MA, 1999, in press.

T.E. Bell, Performance engineering: Doing it "later" on large projects, CMG Trans (1987), 75–81.

B.S. Blanchard, System engineering management, 2nd ed., Wiley, New York, 1997.

B.W. Boehm, Software risk management, IEEE Computer Society Press, Washington, DC, 1989.

F.P. Brooks, Jr., The mythical man-month: Essays on software engineering, anniversary edition, Addison-Wesley, Reading, MA, 1995.

T.R. Browning, Exploring integrative mechanisms with a view towards design for integration, Proc Fourth ISPE Int Conf Concurrent Eng: Res Appl, Rochester, MI, August 20–22, 1997, pp. 83–90.

T.R. Browning, Integrative mechanisms for multiteam integration: Findings from five case studies, Syst Eng 1 (1998a), 95–112.

T.R. Browning, Modeling and analyzing cost, schedule, and performance in complex system product development, Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA, 1998b.

T.R. Browning, Sources of schedule risk in complex system development, Proc Eighth Annu Int Symp INCOSE, Vancouver, July 26–30, 1998c, pp. 187–194.

T.R. Browning and S.D. Eppinger, A model for development project cost and schedule planning, Working Paper No. 4050, MIT Sloan School of Management, Cambridge, MA, November, 1998.

M. Carrascosa, S.D. Eppinger, and D.E. Whitney, Using the design structure matrix to estimate product development time, Proc ASME Des Eng Tech Conf (Des Automat Conf), Atlanta, GA, September 13–16, 1998.

T. Christiansen, The virtual design team: Using simulation of information processing to predict the performance of project teams, CIFE Working Paper No.15, Center for Integrated Facilities Engineering, Stanford University, June 11, 1992.

K.B. Clark and T. Fujimoto, Product development performance: Strategy, organization, and management in the world auto industry, Harvard Business School Press, Boston, 1991.

D. Clausing, Total quality development: A step-by-step guide to world-class concurrent engineering, ASME Press, New York, 1994.

E.H. Conrow, Managing risk in aerospace programs, Aerospace Am (1997), 36–39.

E.H. Conrow and P.S. Shishido, Implementing risk management on software intensive projects, IEEE Software (1997), 83–89.

M.A. Cusumano and R.W. Selby, Microsoft secrets: How the world's most powerful software company creates technology, shapes markets, and manages people, The Free Press, New York, 1995.

J.P. Draves, A risk management process for department of defense acquisition programs operating in an integrated product development environment, Master's Thesis (Eng.), California State University, Long Beach, 1993.

DSMC, Risk management guide for DoD acquisition, Defense Systems Management College Press, Ft. Belvoir, VA, 1998.

K.M. Eisenhardt and B.N. Tabrizi, Accelerating adaptive processes: Product innovation in the global computer industry, Admin Sci Quart 40 (1995), 84–110.

S.D. Eppinger, A planning method for integration of large-scale engineering systems, Proc Int Conf Eng Des ICED—97, Tampere, August 19–21, 1997.

S.D. Eppinger, D.E. Whitney, R.P. Smith, and D.A. Gebala, A model-based method for organizing tasks in product development, Res Eng Des 6 (1994), 1–13.

S.D. Eppinger, M.V. Nukala, and D.E. Whitney, Generalised models of design iteration using signal flow graphs, Res Eng Des 9 (1997), 112–123.

E.M. Goldratt, Critical chain, The North River Press, Great Barrington, MA, 1997.

S.B. Graves, The time-cost tradeoff in research and development: A review, Eng Costs Prod Econ 16 (1989), 1–9.

R.K. Gulati and S.D. Eppinger, The coupling of product architecture and organizational structure decisions, Working Paper No.151, MIT International Center for Research on the Management of Technology, Cambridge, MA, May 1996.

A.K. Gupta and D.L. Wilemon, Accelerating the development of technology-based new products, Calif Manage Rev 32 (1990), 24–44.

E.M. Hall, Managing risk: Methods for software systems development, Addison-Wesley, Reading, MA, 1998.

G.M. Hoedemaker, J.D. Blackburn, and L.N.V. Wassenhove, Limits to concurrency, INSEAD, Fontainebleau, France, working paper, January 1995.

V. Krishnan, S.D. Eppinger, and D.E. Whitney, A model-based framework to overlap product development activities, Manage Sci 43 (1997), 437–451.

A. Kusiak and J. Wang, Decomposition of the design process, J Mech Des 115 (1993), 687–695.

C.H. Loch and C. Terwiesch, Communication and uncertainty in concurrent engineering, Manage Sci 44 (1998), 1032–1048.

C.D. McDaniel, A linear systems framework for analyzing the automotive appearance design process, Master's Thesis (Mgmt./EE), MIT, Cambridge, MA, 1996.

R.T. McNutt, Reducing DoD product development time: The role of the schedule development process, Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA, 1998.

S. Mello, Product and service definition, Product Development Consulting, Boston, 1997.

N.F. Michelena and P.Y. Papalambros, Optimal model-based decomposition of powertrain system design, Proc ASME Adv Des Automat, 1995.

M.D. Morelli, S.D. Eppinger, and R.K. Gulati, Predicting technical communication in product development organizations, IEEE Trans Eng Manage 42 (1995), 215–222.

Management Roundtable, Lexington, MA, An interview with Don Reinertsen (two parts), in Product Development Best Practices Report, Vol. 4, 1997.

S.M. Osborne, Product development cycle time characterization through modeling of process iteration, Master's Thesis (Mgmt./Eng.), MIT, Cambridge, MA, 1993.

H. Petroski, To engineer is human: The role of failure in successful design, St. Martin's Press, New York, 1985.

H. Petroski, Design paradigms: Case histories of error and judgment in engineering, Cambridge University Press, New York, 1994.

N. Ramachandran, N.A. Langrana, L.I. Steinberg, and V.R. Jamalabad, Initial design strategies for iterative design, Res Eng Des 4 (1992), 159–169.

E. Rechtin, Systems architecting: Creating & building complex systems, Prentice Hall, Englewood Cliffs, NJ, 1991.

D.G. Reinertsen, Managing the design factory: A product developer's toolkit, The Free Press, New York, 1997.

R. Sanchez and J.T. Mahoney, Modularity, flexibility, and knowledge management in product and organization design, IEEE Eng Manage Rev (1997), 50–61.

SEI, Continuous risk management guidebook, SEI, Pittsburgh, 1996.

R. Shishko and E.J. Jorgensen, Evaluation of risk management strategies for a low-cost, high-risk project, Proc Sixth Annu Int Symp INCOSE, Boston, July 7–11, 1996, pp. 711–718.

K.J. Singh, et al., DICE approach for reducing product development cycle, Proc Worldwide Passenger Car Conf Expo, Dearborn, MI, September 28–October 1, 1992, pp. 141–150.

P.G. Smith and D.G. Reinertsen, Developing products in half the time, Van Nostrand Reinhold, New York, 1991.

R.P. Smith and S.D. Eppinger, Identifying controlling features of engineering design iteration, Manage Sci 43 (1997a), 276–293.

R.P. Smith and S.D. Eppinger, A predictive model of sequential iteration in engineering design, Manage Sci 43 (1997b), 1104–1120.

R.P. Smith and S.D. Eppinger, Deciding between sequential and parallel tasks in engineering design, Concurrent Eng Res Appl 6 (1998), 15–25.

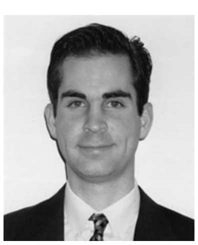
D.V. Steward, Systems analysis and management: Structure, strategy, and design, PBI, New York, 1981.

A. Ward, J.K. Liker, J.J. Cristiano, and D.K. Sobek II, The second Toyota paradox: How delaying decisions can make better cars faster, Sloan Manage Rev 36 (1995), 43–61.

D.E. Whitney, Designing the design process, Res Eng Des 2 (1990), 3–13.

C. Wohlin, M. Xie, and M. Ahlgren, Reducing time to market through optimization with respect to soft factors, Proc IEEE Eng Manage Conf, 1995, pp. 116–121.

A. Yassine, K. Chelst, and D. Falkenburg, A decision analytic framework for evaluating concurrent engineering, IEEE Trans Eng Manage (forthcoming).



Tyson R. 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 Los Alamos National Laboratory. He holds a Ph.D. in Technology, Management, and Policy (interdisciplinary technology management and systems engineering), an S.M. in Aeronautics and Astronautics, an S.M. in Technology and Policy from MIT, and a B.S. in Engineering Physics from Abilene Christian University.