Faculty and Researcher Publications

2006

Estimating Total Program Cost of a Long-Term, High-Technology, High-Risk Project with Task Durations and Costs That May Increase Over Time

Estimating Total Program Cost of a Long-Term, High-Technology, High-Risk Project with Task Durations and Costs That May Increase Over Time

Dr. Gerald G. Brown

Naval Postgraduate School
gbrown@nps.edu

Maj Roger T. Grose

Australian Army

Dr. Robert A. Koyak

Naval Postgraduate School
rakoyak@nps.navy.mil

APPLICATION AREAS:
Analysis of alternatives and cost analysis

OR METHODOLOGIES:
Linear programming, simulation

ABSTRACT

We plan a long-term project schedule for which the total budget depends upon the year the project finishes. Each task in the project can begin only when all its predecessor tasks have been completed, and each task has a range of feasible durations with a month-by-month cost profile for each duration. A task start can be delayed, but once started for some chosen duration, a task cannot be interrupted. Each task suffers some risk of delay and changed cost. Ignoring budget constraints, we use Monte Carlo simulation of the duration of each task in the project to infer the probability distribution of the project completion time. We then optimize a deterministic project schedule following budget guidance. Finally, we successively reschedule as the project progresses, simulating annual review of active tasks, and possibly delaying each active task's duration and changing its monthly costs for its forecast duration. We do not require an independence assumption, so we can accommodate learning effects from completed tasks. U.S. Army Future Combat Systems (FCS) is our motivating application. FCS is a complex of information technologies, sensors, and command systems expected to require more than a decade and $16 billion to develop. The U.S. General Accounting Office finds FCS at significant risk of cost and schedule growth, and suggests two alternatives to a baseline Army plan. We analyze these three alternate project plans for FCS to discover which one can most likely be completed soonest and cheapest.

> *"Now, I'll manage better this time."* ***Alice in Wonderland***

INTRODUCTION

U.S. Army Future Combat Systems (FCS) is a complex of information technologies, sensors, and command systems constituting a project with scores of tasks expected to require more than a decade and $16 billion (2004 U.S. dollars) just in system development and demonstration costs. In fiscal year (FY) 2005, FCS is expected to consume more than half of the U.S. Army's budget for all system development and demonstration, and perhaps $94 billion to acquire 14 of the 18 systems needed for FCS initial operational capability by the year 2010 (Brady, 2003; Francis, 2004). The U.S. General Accounting Office (GAO) (2003) finds FCS vulnerable to significant cost and schedule growth, and suggests alternate project designs to mitigate risk.

Francis (2004) outlines the accomplishments that must be coordinated in order for FCS to succeed, which we paraphrase:

- A specialized C4ISR (Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance) network must be developed for FCS;
- Fourteen major weapon systems and platforms must be designed and integrated simultaneously with other systems, subject to physical limitations;
- At least 53 technologies that are considered critical to achieving required performance capabilities must be matured and integrated;
- At least 157 Army and joint-forces systems must also be adapted to interoperate with FCS, which will require the development of nearly a hundred new network interfaces; and
- An estimated 34 million lines of software code will be required to operate FCS. This is nearly five times the software required for the Joint Strike Fighter, which had the largest software requirement of any Department of Defense acquisition prior to FCS.

FCS is so complex, a number of normal procedural reviews and hurdles have been relaxed, enabling an independent initial operational test and evaluation using an incomplete prototype scheduled for 2008 (Welch, 2003).

We seek a "project design" for such a long-term, high-risk, complex system. We anticipate that higher-risk tasks will exhibit more uncertainty and thus may take longer than planned and cost more. We are willing to state probability distributions predicting the cost and duration of each task, but we view an independence assumption between task outcomes as foolhardy: In complex, high-technology projects, trouble breeds company.

We seek a "robust project schedule" that offers the least schedule risk. We want to plan to complete our project at some given budget, by some given time, with some given probability. We can rearrange some of the planned project *partial orders* among tasks—i.e., what predecessor tasks have to be completed before any given task can start—and these rearrangements might influence schedule robustness. Our problem is: which rearrangement offers the most robust project schedule?

SCHEDULE OPTIONS AND SCHEDULE RISK

We refer to *schedule risk* as the costs of schedule overruns evaluated by their likelihoods. Planners might be presented with a set of options for scheduling the range of tasks that comprise a large acquisition project. These options must abide by a common set of temporal and fiscal constraints. They should also reflect the inherent uncertainty of the completion time of a developmental task. A rational planner assesses the schedule risk of each option and selects the option that affordably poses the least risk.

Significant "knowledge demonstration" (i.e., showing you can actually build components that integrate in the system design) often occurs late in development and early in production of a major defense acquisition program. The highest schedule risk comes when developed components must be integrated into a system of systems. Welch (2003) observes that the unusual complexity of FCS exposes it to higher schedule integration risk than normally expected of a major program. In particular, FCS is susceptible to "late cycle churn" to fix problems discovered late in development. Francis (2004) identifies the following factors that dispose FCS to late cycle churn, which again we paraphrase:

- Technology development is expected to continue through to the production decision;
- Technology development will still be ongoing at the design readiness review, putting at risk the stability of ongoing system integration;
- Production is planned to start while technology development and system integration are continuing and the first prototypes are being delivered;
- The final production decision will be made before some technologies reach their required maturation and before an integrated system demonstration has been conducted;
- Production delivery will start before the Army has completed the first full demonstration of FCS as an integrated system; and
- The full-rate production decision will be made while testing and demonstration are continuing.

The FCS program executive office has prepared a baseline project plan (i.e., a schedule with funding) for the system development and demonstration phase that governs current acquisition policy. Several alternate project plans have been proposed by the General Accounting Office (2003) to mitigate FCS schedule risks. We examine the baseline plan and two of the GAO alternatives here.

1. "FCS baseline" plan

The baseline plan develops all major subsystems concurrently, rather than developing one first to set the development context for follow-on systems. The FCS program executive office acknowledges that this plan is ambitious, and that the program was not ready for system development and demonstration when it was approved (Francis, 2004).

2. "GAO risk first" plan

This plan modifies the baseline to address risky technologies up front, requiring that the technology readiness level (a gauge of completion) be "at least 6" to pass intermediate review and "at least 7" to qualify for production (Wynne, 2003). Many key technologies are below the 6 threshold, and the FCS program executive office has already developed risk-mitigation strategies for each. This GAO suggestion first matures technologies that are below the technical readiness-6 threshold, and then proceeds as scheduled in the baseline plan. The advantage is that test and integration tasks occur later in the schedule, with theoretically reduced schedule risk compared to the baseline plan.

3. "GAO C4ISR first" plan

This plan modifies the baseline to develop C4ISR tasks before all others. The C4ISR components are believed to pose the greatest schedule risks to FCS development due to their scope and complexity. They are expected to require about 16 million lines of software code (of the 34 million total estimated), of which more than half will be new code (Welch, 2003). This huge undertaking is vulnerable to cost and schedule overruns. By investing early in these components, subsequent C4ISR test and integration tasks should pose less risk than in the baseline.

Key distinctions between these alternate plans are that the partial orders among tasks may change between plans and any task common to all of the plans may be allocated different risk levels in each. For instance, "system integration and testing" is high-risk in the "baseline plan" because immature technologies must be concurrently developed and integrated, but this same task has lower risk in the "GAO risk first" plan.

The three alternate plans displayed in the Appendix use nominal (i.e., unclassified, nonproprietary) FCS task data provided by the Cost Analysis Improvement Group, Program Analysis and Evaluation (PAE), Office of the Secretary of Defense.

TIME FIDELITY

Monthly time fidelity suffices for purposes of long-term planning and budgeting, although it is also customary to offer annual budget accounting for such plans and perhaps to conduct annual reviews of task progress. Indeed, annual task reviews are the most substantive control points in such projects, given that they are tied to annual budget authorizations. Accordingly, we plan all activities and events in months, but make annual task-state reviews with possible consequences on task duration and time.

EVALUATE EACH "PROJECT DESIGN" FOUR WAYS

We were asked to analyze the FCS baseline plan and the two GAO alternatives. The following is essentially a series of project reports as we went back to our PAE sponsor with intermediate results, seeking guidance for the next steps to try. We do not recount a lot of ideas that did not work. Overall, we spent 8 person-weeks with PAE, and 24 person-weeks finding out what works, and what doesn't. Remember: *the goal here is discovering new, effective ways to improve cost estimation for this huge, complex project, not manage it.*

First, we just find the deterministic project duration (i.e., the "shortest longest path length" in time, or simply the "critical path length."). This is easy, and exercises our newly-completed scenario data sets. We are still debugging and scrubbing data.

Then, we ignore costs and budgets, but assert probability distributions for task durations and apply Monte Carlo simulation to evaluate the critical path induced from each sampled project instance. The statistics we gather, and experience we gain, helps us understand the behavior of each project design, especially the partial orders among tasks.

Next, we provide a list of total project durations in years and a total program budget for achieving each of these durations. We specify the year-by-year spending goal of any selected project duration. Each task can be started only when all its predecessors in the project design have been completed. Each task can be started for any of a range of durations in months, and each of these durations has a monthly cost profile. Once a task is started, it cannot be interrupted. However, a task start can be delayed for lack of available budget(s) sufficient to support its chosen, uninterrupted duration once started. We optimize this deterministic, cost-constrained project schedule to minimize total project duration.

We note that "costs" need not be strictly expressed in constant-dollar allocations, but can include policy penalties rewarding desirable outcomes (i.e., finishing earlier), or penalizing bad ones (i.e., finishing very late). But, although completion time is a concern, the over-arching constraint will be total obligation authority (i.e., money) committed to the program.

Finally, we nest our cost-constrained project schedule optimization within an annual state review simulation of each *active task* (i.e., task in progress at time of review). At each annual review, each active task may be delayed depending on a probability distribution that depends on the risk of that task, *or on any prior experience with any other task.* So, year-by-year, we conduct an annual state review of all the active tasks, then reoptimize the remaining planning horizon. This takes a lot of computation, but the insights are worth the effort.

RELATED RESEARCH

Malcolm, Roseboom, Clark, and Fazar (1959) introduce Program Evaluation and Review Technique and Critical Path Method (PERT-CPM) developed for the Polaris fleet ballistic missile program, and Kelly (1961, Kelly 1963) provides a mathematical foundation. Wiest (1964) highlights two key shortcomings in CPM at its nascent stage: it only considers constant task durations and does not recognize resource constraints.

More recent concepts of CPM allow for greater flexibility in these areas, for example by allowing tasks to be scheduled in either "regular time" (with nominal costs) or in "crash time" (with higher costs), and by allowing cost constraints. Even with these innovations the concept of a "task" remains unitary in nature. At a fixed point in time of the project, tasks that are underway are not subject to decisions that affect their remaining times until completion.

If each task duration is random, and some deterministic equivalent time is used in CPM, estimates of project duration are generally optimistic as Fulkerson (1962) demonstrates using discrete random task durations. A task not on a critical path using mean durations may be on the critical path with positive probability when its duration is treated as a random variable. Dodin (1984) reports upper and lower bounds on project duration when task durations are independent random variables, and uses the Central Limit Theorem to justify treating the project duration as approximately normally distributed. While this assumption offers tractability, the longest random-length path is neither normally distributed in theory, nor in practice (as can be verified by simple Monte Carlo simulation), and this assumption can give misleading results.

Resource constraints are admitted by Bowman (1958), who introduces linear programming for CPM, and Senju and Toyoda (1968) and Pritsker, Watters, and Wolfe (1969) state integer-linear programs representing discrete decisions. Demeulemeester and Herroelen (2002) present formulations of resource-constrained project scheduling problems and review solution methods.

Using linear and integer linear programs to represent stochastic models has a long history. Babbar, Tintner, and Heady (1955), Tintner (1955, Tintner 1960), and Sengupta, Tintner, and Morrison (1963) show how to embed optimization within Monte Carlo simulation. Task duration may be treated as a random variable with a distribution not completely known (Herroelen, Reyck, and Demeulemeester, 1998). Factors influencing these random variables include resource availability, scheduling of deliveries, modification of due dates, and changes in project scope that might imply the cancellation or addition of future tasks (Herroelen and Leus, 2004).

Generally, the increased realism of stochastic PERT-CPM modeling comes at the price of increased analytic abstraction and computational cost. Deterministic equivalent objectives, such as the *expected* project critical path length or expected costs that include penalties for violating constraints (Gutjahr, Stauss, and Wagner, 2000), may be easy enough to state and solve, but the risk of such solutions is much more difficult to gauge, even given generous independence assumptions.

If task duration is random and not independent of other task durations, the distribution of the total project duration is difficult to characterize (Yang, Geunes, and O'Brien, 2001). An independence assumption is often made to render tractable analysis, but this assumption is not realistic. An optimal deterministic schedule typically has insufficient slack to remain optimal (or even feasible) in an uncertain setting, and thus lacks robustness (Herroelen, 2004). A trivial example with two identical, parallel tasks, each with random duration, reveals this property.

In addition, managers want the flexibility to change their scheduling decisions as the project evolves. *Full dynamic scheduling* offers decision points at task completions (Igelmund and Radermacher, 1983).

We need resource (essentially budget) constraints and we cannot ignore uncertainty. Of all these historical contributions, we admire Tintner's works most for their originality, elegance, and simplicity, and we follow his advice: for the stochastic modeling, use Monte Carlo identity simulation, and then use optimization for each random realization.

Finally, we do assume, as does PERT-CPM, that each task is separable and distinct from all others. In our case, these tasks are subcontracts, so this is true in law as well as in fact: If you want to re-define tasks, you must re-negotiate contracts.

FIND SHORTEST PROJECT COMPLETION DATE FOR EACH ALTERNATE PLAN WITH NO BUDGET CONSTRAINT

To check our alternate project plans to see if we get schedules that make sense, we ignore budget constraints and just solve a deterministic CPM problem.

Given a project network with fixed task durations, we wrote a Java (Sun Microsystems, 2005) procedure for an unconstrained reaching algorithm to search the project tasks over their adjacencies in partial order to find the completion time of the project. The completion time is the length of a longest path from project start to finish. This is one of the simplest network algorithms (e.g., see topological sorting and reaching in Ahuja et al., 1993, pp. 107–108), with worst-case runtime linear in the number of partial orders.

From a project start in January 2003, this primitive deterministic analysis yields an earliest project completion date of October 2012, for the "FCS baseline plan." The Army wants to field its first unit in September 2012, so this is reassuring.

Given that we can solve each of these deterministic problems in less than a millisecond, we suggested solving thousands of these problems in a Monte Carlo simulation to assess stochastic elements of each project alternative. PAE agreed.

MONTE-CARLO SIMULATE TASK DURATIONS FOR EACH ALTERNATE PLAN WITH NO BUDGET CONSTRAINT

The three-parameter Weibull distribution is often used to model the duration of developmental tasks for cost estimation and planning. Law and Kelton (2000, p. 376) explain the reasoning for the use of this distribution. The Weibull reliability function:

$$R(x;\, \alpha,\, \beta,\, \gamma) = e^{-\left(\frac{x-\gamma}{\alpha}\right)^{\beta}},\, x \geq \gamma$$

is completely characterized by its three nonnegative parameters. An absolute minimum task duration is given by γ. For $\beta > 1$, the Weibull density has a mode strictly greater than γ, and this mode appeals managerially as the task duration of maximum likelihood. The Weibull also features more and larger deviations from the mode in the positive direction.

Miller (2003) offers a convenient procedure for specifying the parameters of a three-parameter Weibull distribution from intuitive properties of task duration. We need a value for the duration mode, x_M, (for this, we just use the longest admissible task duration) and a categorization of the risk level as high, medium, or low. Miller suggests high risk for unprecedented tasks, medium for development and some new integration tasks, and low for routine, repetitive, or well-understood tasks. Each risk level is associated with fixed values of two attributes of the task duration that together with with the mode x_M are sufficient to determine all three parameters of the Weibull. Attribute $R_M = x_M/\gamma$ is the ratio of the mode to the minimum duration and $P_M = P(X > x_M)$ is the probability that the duration exceeds the mode. Miller suggests for (risk, R_M, P_M) the values (high, 1.25, 0.8), (medium, 1.20, 0.7), or (low, 1.15, 0.6). PAE concurs.

The three-parameter Weibull distribution can be defined using either triplet (α, β, γ) or (R_M, x_M, P_M). Table 1 shows the mapping between these equivalent descriptions. For example, a medium-risk task with most-likely duration $x_M = 36$ months is endowed with $R_M = 1.20$ and $P_M = 0.7$, and the associated Weibull parameters are:

$$\alpha = \frac{36(1 - 1/1.20)}{[-\ln(0.7)]^{1+\ln(0.7)}} = 11.65,$$

$$\beta = \frac{1}{1 + \ln(0.7)} = 1.554, \text{ and } \gamma = \frac{36}{1.20} = 30.0.$$

In consultation with PAE, we truncate our Weibull at its 90th percentile to avoid unrealistically-long project durations. The maximum allowable duration truncation point is calculated $d_{Max} = \gamma + \alpha[-\ln(0.1)]^{1/\beta}$. Such a Weibull is trivial to generate from a unit-uniform variate U via $X = \gamma + \alpha[-\ln(1 - .9U)]^{1/\beta}$.

We compare the three FCS project plans (baseline, GAO risk first, and GAO C4ISR first) ignoring cost constraints. For each simulated iteration, new task durations are sampled from their Weibull probability distributions and the resulting project completion time is recorded. The simulation is repeated for 60,000 iterations (i.e., we commit about a minute of computing time to each case). We thus induce the random distribution of project completion time for each project plan. Results from these simulations appear in Figure 1.

Table 1. Association between attributes and parameters of the three-parameter Weibull distribution show how to map from attributes to Weibull parameters or vice versa.

Attributes	Parameters
$R_M = \frac{x_M}{\gamma}$	$\alpha = \frac{x_M(1 - 1/R_M)}{[-\ln(P_M)]^{1+\ln(P_M)}}$
$x_M = \gamma + \alpha\left(1 - \frac{1}{\beta}\right)^{1/\beta}$	$\beta = \frac{1}{1 + \ln(P_M)}$
$P_M = e^{-(1-1/\beta)}$	$\gamma = \frac{x_M}{R_M}$

OPTIMIZE A BUDGET-CONSTRAINED DETERMINISTIC SCHEDULE

Our real-world project has a budget and costs that may be influenced by the rate at which we work to finish tasks. We adopt monthly planning fidelity. For each task, we introduce a set of discretionary task durations where each duration has its own month-by-month cost profile for completing the task. Our total project budget depends on the finish year we choose, where each candidate finish year induces a completely independent set of year-by-year budget guidelines. These generalizations suggest an optimization model to identify the least expensive feasible project completion time. We discretize the starting times for tasks and task durations to months, and to use the following integer linear program to suggest a project schedule:

Index Use [$\sim$cardinality]

$y \in Y$	Fiscal year (alias *yh*, *yf*) [$\sim$20]
$i \in I$	Task (alias *j*) [$\sim$200]
$\ell \in I$	Distinguished, last task in project
$(i, j) \in A$	Pairwise partial order: task *i* must be completed before task *j* starts
$m \in M$	Planning month [$\sim$240]
$m \in M(y)$	Month in fiscal year *y*
$s = s_i \in S_i \subseteq M$	Start month for task *i*
$d = d_i \in D_i$	Task *i* duration in months
$1 \leq p_i \leq d_i$	Months since start of ongoing task *i*

Given Data [units]

$\underline{budget}_{y,yf}$, $\overline{budget}_{y,yf}$	Lower and upper cost range during fiscal year *y* if program finishes in fiscal year *yf* [cost]
$cost_{idp}$	Cost of ongoing task *i* with duration *d*

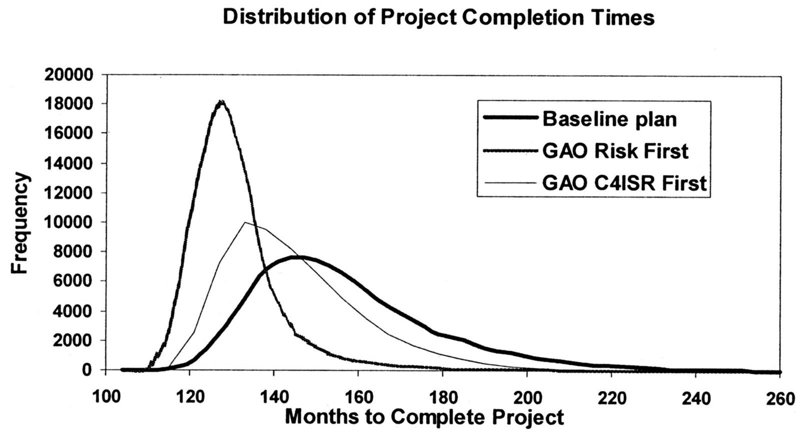

Figure 1. Sixty-thousand samples of each alternate project plan are depicted. There are no cost constraints and each task duration is generated independently from a Weibull distribution reflecting its risk in that plan. "GAO risk first" is the most desirable plan with the highest probability of an early completion time, while the baseline plan has the lowest probability of successful completion at any given time.

	during elapsed month p [cost]
pen_under, *pen_over*	Cost per unit of cumulative budget range violation [months/cost]

Decision Variables [units]

X_{isd}	= 1 if task i is started in month s with duration d, 0 otherwise [binary].
Q_{yf}	= 1 if finish year of program is year yf, 0 otherwise [binary].
$UNDER_y, SLACK_y, OVER_y$	When we compare expenditures through fiscal year y with desired lower and upper ranges on total budgets, these variables respectively measure lower-range violation, unspent funds below upper-range, or upper-range violation [months/cost].

Formulation

$$\underset{\substack{X,Q\\ UNDER,SLACK,OVER}}{MIN} \sum_{s\in S_\ell, d\in D_\ell \wedge s+d-1\le \|M\|} (s+d-1)\,X_{\ell sd} + \sum_{y\in Y} (pen\_underUNDER_y + pen\_overOVER_y) \quad \text{(F1)}$$

$$s.t. \sum_{s\in S_i, d\in D_i \wedge s+d-1\le \|M\|} X_{isd} \ge 1 \quad \forall\, i \in I \quad \text{(F2)}$$

$$X_{\ell sd} \le Q_{yf} \quad \forall\, yf \in Y, s \in S_\ell, d \in D_\ell \wedge s+d-1 \in M(yf) \quad \text{(F3)}$$

$$\sum_{yf\in Y} Q_{yf} \le 1 \quad \text{(F4)}$$

$$\sum_{\substack{yh \le y, m \in M(yh), i \in I, s \in S_i, d \in D_i \\ \wedge m-s+1 \ge 1 \wedge m-s+1 \le d \wedge s+d-1 \le \|M\|}} (cost_{id(m-s+1)} X_{isd}) + UNDER_y + SLACK_y - OVER_y = \sum_{\substack{yh \le y, \\ yf \in Y \wedge yf \ge y}} (\overline{budget}_{yh,yf}) Q_{yf} \quad \forall\, y \in Y \quad \text{(F5)}$$

$$SLACK_y \le \sum_{\substack{yh \le y, \\ yf \in Y \wedge yf \ge y}} (\overline{budget}_{yh,yf} - \underline{budget}_{yh,yf}) Q_{yf} \quad \forall\, y \in Y \quad \text{(F6)}$$

$$\sum_{s \in S_i, d \in D_i \wedge s+d-1 < s_j} X_{isd} \ge X_{js_jd_j} \quad \forall\, (i, j) \in A,\ s_j \in S_j,\ d_j \in D_j \wedge s_j + d_j - 1 \le \|M\| \wedge s_j > \underset{s \in S_i d \in D_i}{MIN} (s + d - 1) \quad \text{(F7)}$$

$$X_{isd} \in \{0, 1\} \quad \forall\, i \in I,\ s \in S_i,\ d \in D_i \quad \text{(F8)}$$

$$Q_{yf} \in \{0, 1\} \quad \forall\, yf \in Y \quad \text{(F9)}$$

$$UNDER_y \ge 0, SLACK_y \ge 0,\ OVER_y \ge 0 \quad \forall\, y \in Y \quad \text{(F10)}$$

Verbal Description

The overarching goal is to decide how long the project should take to complete. The objective function ($F1$) expresses total planned project duration in months, plus an elastic penalty term for any violation of cumulative budget ranges over the planning horizon. Each partition constraint ($F2$) requires that exactly one start month and duration be selected for each task. Each constraint ($F3$) permits the last project task to be completed in a fiscal year only if that fiscal year has been selected for project completion. Constraint ($F4$) requires that exactly one project completion year be selected. Each constraint ($F5$) accumulates expenditures from the first fiscal year through a current fiscal year and determines whether the cumulative budget ranges have been satisfied, or violated. (This cumulant form is amenable to both a linear programming solver and to managerial interpretation: Brown et al., 1997.) Each constraint ($F6$) limits cumulative slack budget by the hard constraints on yearly program budget determined by finish year. Each constraint ($F7$) ensures, for a pair of tasks adjacent in precedence, that the predecessor task must be completed before the successor task can start. Variable domains are defined by ($F8$–$F10$). ($F8$) can restrict admissible start months for each task and the admissible durations of each task.

TASK DURATIONS AND COSTS

For a task started in month s for duration d months, we can assert any month-by-month cost distribution we want, even including costs for months preceding task start or following task completion (as military research and development often requires: Brown et al., 2004). Here (following explicit guidance from PAE), we simplify: no matter when a task might start for a d-month duration, we allocate its $Task\_Cost_d$ over each month of this duration with a Rayleigh distribution truncated at its 97-th percentile, so that its cost in month p would be:

$$Month\_Cost_p = [Task\_Cost_d / 0.97][\exp(\{(p - 1)^2 \ln(0.03)\}/d^2) - \exp(\{p^2 \ln(0.03)\}/d^2)].$$

EACH POSSIBLE COMPLETION YEAR HAS ITS OWN BUDGET

The key policy question is (always) "how much are we willing to spend and when are we willing to spend it to finish our project (e.g., by the end of any given future fiscal year)?" Are we willing to spend more for a quicker completion? Are there competing projects that restrict our planned spending pattern? For planning purposes, sooner or later we have to at least estimate upper and lower limits on the overall planned project budget for each financial year of each planned project duration. Here, for any candidate project completion year and budget, we also use a Rayleigh distribution to distribute this budget year-by-year.

A complex, long-term military project rarely meets all its planned budget targets. Sometimes allocated funds are available before they can be used and sometimes costs exceed projections. Accordingly, we accumulate any year-by-year over-expenditure or under-expenditure, but penalize any such cumulative violation year-by-year until the surplus or deficit is repaired. The idea (Brown et al., 2004) is to allow some reasonable flexibility in program management, while showing good faith adhering to overall project budget guidance.

FCS ANNUAL BUDGETS

An FCS project budget estimate has been developed with help from PAE. Separate estimates must be prepared for each feasible project duration, ranging from FY2010 to FY2016. Table 2 shows the minimum, planned and maximum annual budgets for a FY2011 completion that has been Rayleigh-allocated over the planning years.

Preparing budgets for each completion year, we try to follow the best guidance available. For example, a GAO review of FCS (Francis, 2004) concludes that a one-year delay in FCS would increase costs by $4 billion to $5 billion (during the system development and demonstration, and production phases). Relative to the total projected cost of FCS, this represents a 0.5% cost overrun per year of delay. Conversely, Lee (1997) estimates for projects in general that accelerating the pace of work and decreasing a project duration by one year would require an increased budget of 0.2%. Of course, delays in any accelerated plan subject it to cost overruns as well.

SUPERIMPOSE MONTE CARLO SIMULATION OF ANNUAL TASK REVIEWS (WITH POSSIBLE TASK DELAYS AND COST CHANGES) ON SCHEDULE OPTIMIZATION

We nest our cost-constrained project schedule optimization within a simulated annual "project review" of each then-active task. Each reviewed active task may be delayed depending on a probability distribution that depends on the risk of that task, *or on any prior experience with any other task.* The cost of each reviewed task may also change, as can the forecast cost or duration of any future task. Each annual project review is followed by a re-optimization of the remaining future planning horizon. Year-by-year, we conduct an annual project review, re-optimize, and so forth. Figure 2 illustrates how this simulation might progress.

In our simple example, each active task reviewed is delayed with (risk, probability, and delay) of (high, 0.5, 140%); (medium, 0.3, 120%); or (low, 0.2, 110%), and a delayed task's costs

Table 2. For a project completion in FY2011, a nominal total FCS system development and demonstration budget of 20.04 billion 2004 dollars has been Rayleigh-allocated by fiscal year. These planned annual budgets are goals, but the minimum (20%) and maximum (105%) budget ranges are hard constraints. The sums of annual expenditures from FY2003 through any given year are constrained by these cumulative hard constraints. Within these hard cumulative limits, any cumulative expenditure under- or over-plan is penalized and carried forward to the next year, where it will be penalized again if not mitigated.

Year	Minimum Budget ($ Million)	Planned Budget ($ Million)	Maximum Budget ($ Million)
FY2003	$ 175	$ 875	$ 919
FY2004	482	2,410	2,530
FY2005	676	3,382	3,551
FY2006	732	3,658	3,841
FY2007	667	3,335	3,502
FY2008	530	2,652	2,785
FY2009	374	1,871	1,965
FY2010	237	1,183	1,242
FY2011	135	674	708
TOTAL	$4,008	$20,040	$21,042

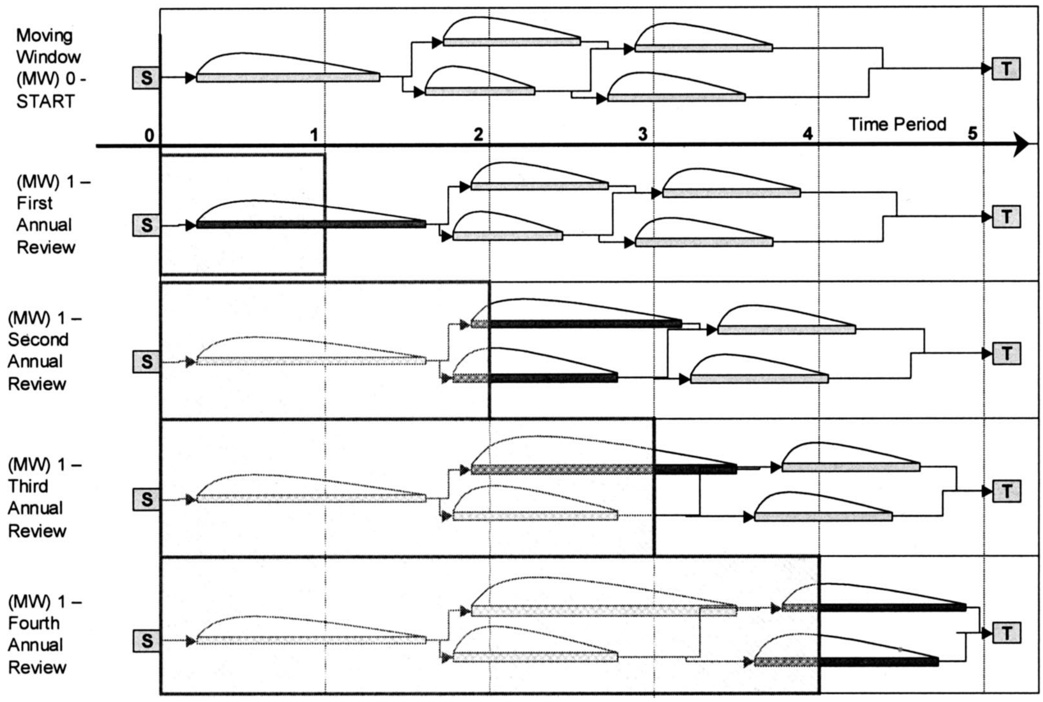

Figure 2. Each annual review (depicted top-to-bottom separating the shaded and un-shaded portions of each timeline row) may delay any currently-active task (i.e., any highlighted task spanning shaded and un-shaded timelines), or change its cost. After each annual review, the remaining schedule is re-optimized with monthly fidelity, subject to annual budget goals induced by the best project duration still achievable. The optimization must complete currently-active tasks as specified by the latest annual review, but can choose any admissible start month for any future task and choose any future admissible task duration it pleases, as long as the associated costs of the chosen duration are bearable. Directed arcs between partially-ordered task pairs and nominal Rayleigh-distributed task budgets are shown to illustrate how the optimization must schedule tasks such that total expenditures follow annual budget guidance.

increase with (risk, probability, change) of (high, 0.5, 150%), (medium 0.3, 130%), or (low, 0.2, 110%). As a practical matter, we permit a task to be delayed to, at most, twice its original duration (longer than this and the task would likely be cancelled, and the project redesigned).

If an annual review extends the remaining optimized project duration, the total project budget changes accordingly (here, it is increased in proportion to the length of the extended duration, though any adjustment is admissible).

This amalgam of annual budget review simulation and optimization of the remaining planning horizon offers a face-valid emulation of actual practice, and our Monte Carlo annual simulation can easily be replaced with a human umpire if more expert control and judgment appeal. We have tested dependent models for inflating costs and task durations, and two key lessons emerge: even mildly inter-task dependent delays cause havoc, and any project overseer would intervene long before these results played out. Although we could model *decreases* in task duration and/or cost, this prospect has never come up with PAE, nor have we ever observed such a signal event in our careers.

IMPLEMENTING THE OPTIMIZATION MODELS

The alternate project plans have been set up in Microsoft Project (2004). We want to use the graphical user interface offered by Project, as well as its integration with the MS Office Suite. Our optimization (with optional Monte Carlo annual reviews) is been implemented in the algebraic modeling language GAMS (Brook et al., 1998).

Each scenario is presented to GAMS as two input scripts, one for tasks, and the other for budgets. The former script has a descriptor for each task specifying each candidate start month, duration in months, and cost. Nobody will use an optimization model they can't control, so this script (via the Project interface) lets a planner completely control alternatives, including "start this task in this month for this duration at this cost."

As you would expect, the GAMS script imports a scenario from Project, solves it, and returns the solution for display and analysis. But, the majority of our GAMS script is devoted to diagnosis and exigent report writing, to better monitor the behavior of our experimental models.

For instance, early experience with our model revealed that although we offer precise controls for task start times and durations, nobody used these: by default, each task can start any time for any admissible duration. As a consequence, an enormous number of alternate task start variables was generated. Solvers mechanically detect and remove redundant model features. However, such "presolve" features do not tell you what they have removed, *or why.* And, presolve will not identify all redundancies: each reduction involves no more than removing one redundant variable with an equation substitution. You can't be sure you have removed the redundancies you worry about unless you filter them out yourself.

So, in addition to the index domain filtering that clutters the summations in our formulation (but makes our intent clear), we formulated an auxiliary, trivial optimization (not displayed) to find the admissible start times and durations for each task.

We work on our formulation and model generator until presolve finds as little as possible left to remove. After such filtering, a typical scenario consists of about 53 thousand constraints, and 19 thousand variables, almost all binary. We would expect such an integer linear program to solve on a laptop in minutes.

We used CPLEX 9.0 (ILOG, 2004). Default CPLEX stalled, and could not find an initial feasible integer solution. We provided an admissible integer starting point from our trivial presolve. CPLEX bogged down in problem preprocessing and integer cut generation. Eventually, to get CPLEX to work, we had do disable most of its default options for cut generation and root node heuristics.

Our solve times are still longer than we expected. If we fix project duration and budget, the resulting optimization model is easier to solve (and we can automate this fixing in GAMS for each project duration we fancy). However, even this simplifying restriction leaves us with a daunting scheduling problem: to choose a start time and duration for each task that satisfies every partial order between tasks, maximally complies with the cumulative budget guidance, and also finishes on time. Typically, it takes us 3 GHz-hours to resolve to a 10% integrality gap.

These integer linear programs may be hard to solve, but they convey remarkable insight we have not gained by any other means. ILP models depend on well-defined assumptions and offer fidelity that closely mimics real-world planning, and they also convey an objective assessment of solution quality that, for instance, lets us confidently compare alternate scenarios.

For instance, the objective assessment of solution quality we get from the integer linear programs is invaluable when comparing two competing alternatives: given assumptions stated clearly, and data defined commensurately, no matter how complex the project, if the optimized solutions exhibit integrality gaps that do not intersect, we can confidently declare a winner.

RESULTS AND CONCLUSION

A Rayleigh-distributed project budget just does not fit the needs of the constituent FCS tasks as the project proceeds for any alternate project plan. Accordingly, we state the budget as a cumulative goal from project start in FY2003 through each year, with any cumulative under- or over-expenditure carried forward to later years, charging a penalty for any deviation from cumulative budget until that violation is rectified. Without this flexibility, we must extend the project finish year and leave Rayleigh-allocated funds for intermediate years unused. For long-term planning, this makes no sense at all. Figure 3 shows how the deterministic, optimized plans use the Rayleigh-distributed project budget, and displays the same expenditures in cumulative terms.

Starting the "baseline plan" in January 2003, we find a (cumulative) cost-constrained schedule that finishes just as quickly as primitive CPM with no cost constraints at all: October 2012. Our nominal task costs and total budget restrictions *just suffice* without delaying the project: this is another reassuring discovery. And, our suggested cumulative expenditure history follows long-term guidance closely.

When we simulate annual reviews, with task delays, the optimized project plans take a lot longer to complete (see Figure 4). Monte Carlo delays of task durations as the project proceeds extend achievable project completion, so the projected budget (discovered year-by-year as the project progresses and these delays arise) is characterized by transitions to successively longer finish-year budgets (see Figure 5).

For the baseline plan, just introducing random Monte Carlo task durations increases the median project duration by about 10%. If budget constraints are imposed in addition to random task delays, estimated project duration rises by about 39%. For FCS, a 39% delay corresponds to approximately four years, where a one-year delay has been estimated by the GAO to add between \$4 billion and \$5 billion to the total acquisition cost.

In the absence of budget constraints, mitigating the technologies below the required maturity level prior to other tasks (GAO risk first) leads to project completion faster than the baseline plan. When budget constraints are added, this plan maintains its advantage although it is subject to delays similar to the baseline plan.

Table 3 assembles FCS project duration estimates for each alternate plan and from each of our models. With no budget constraint, it's best to mitigate high-risk technology first. With project budget constraints, both the baseline and risk-first plans are attractive, but with annual review simulation, the GAO C4ISR first plan turns out to be least vulnerable to delay. Given the high risk of the FCS program, we prefer the behavior of GAO C4ISR first.

FCS is a long, complex, technically risky, expensive, and *important* project. But, FCS is not unique in these respects: there are (always) other defense projects that are comparable (Brown et al., 2004). Based on our planning experience with such projects, we recommend a high-level assessment such as that presented here to forecast as early as possible and as well as possible where the fragilities

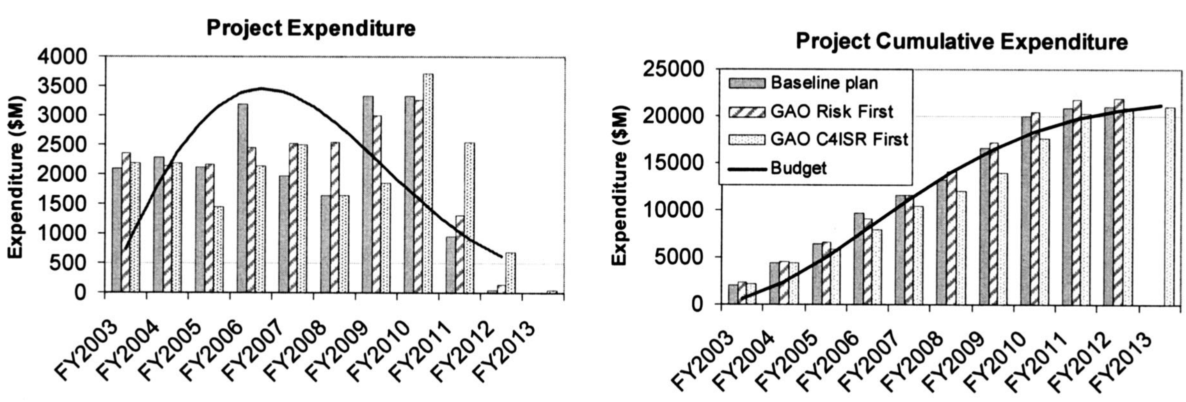

Figure 3. A Rayleigh-distributed budget for a FY2012 project finish is shown in annual and cumulative terms along with deterministic, optimized expenditures for each alternate project plan. The budget is stated as a cumulative goal from project start in FY2003 through each year, with any cumulative under- or over-expenditure carried forward to later years, charging a penalty for any deviation from cumulative budget goal until that violation is rectified. Note the banking of unused budget (e.g., in FY2007) in anticipation of borrowing it back (e.g., in FY2010). Without this flexibility, we must extend the project years beyond FY2012 and leave allocated funds for intermediate years unused. For long-term planning, this makes no sense at all.

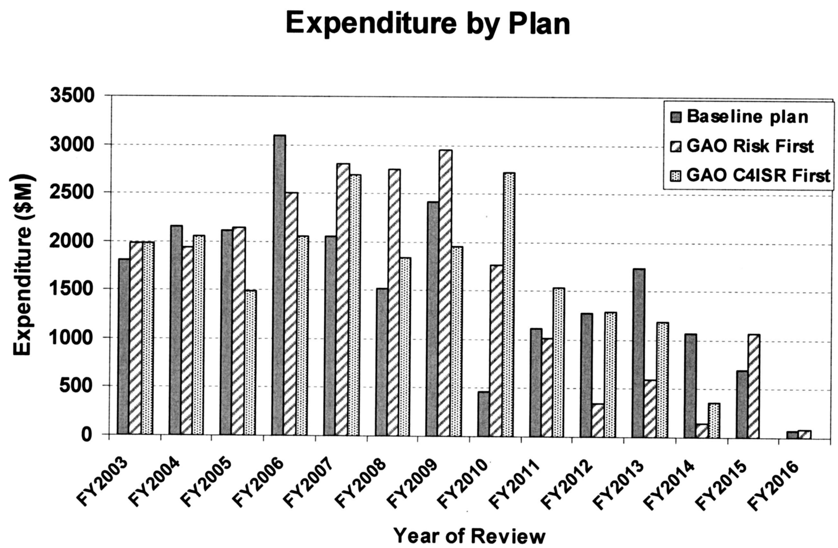

Figure 4. Annual expenditures are shown for optimized FCS project schedules with a simulated annual review of each then-active task that, depending on task risk, may randomly induce a delay and a cost increase. As expected, project completion is delayed for all project plans, and costs rise (by about four years and $600 million, respectively). The idea is to animate how these task delays arise over time and how they cascade and influence other competing or succeeding tasks. Note that GAO C4ISR finishes two years before the other plans.

and vulnerabilities are in an overall project plan, and to prescribe work-arounds sooner, rather than later.

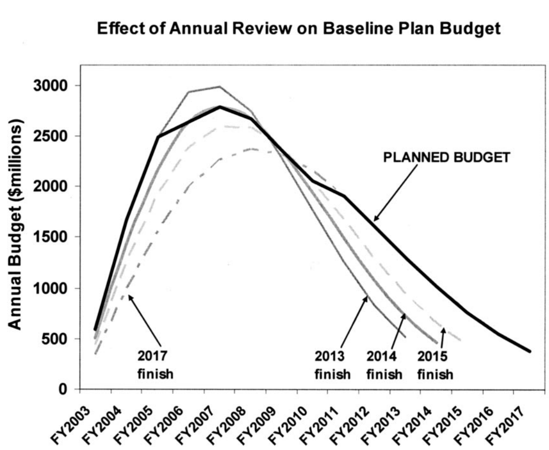

Figure 5. For the baseline plan, as the project progresses year-by-year and is subjected to annual reviews that delay then-active tasks, the remaining project tasks are reoptimized and the project takes longer to complete. This shifts the best achievable project budget to a later year. The display shows when the optimization must jump to a larger and longer budget.

We emphasize key, distinguishing, real-world advantages offered here. *These component models are easy to discuss, illustrate, brief, and understand:* Project scheduling and primitive Monte Carlo simulation are ubiquitous. We co-opt the graphical user interface of a project management system and its data base, and embed the optimizer and Monte Carlo reviews, thereby producing a visually-appealing planning product at a low per-seat cost. The embedding of deterministic optimization within time-phased simulation decouples the two in a way that requires few simplifying assumptions and that invites very basic, intuitive analysis to evaluate results. We closely mimic real-world behavior:

- each optimization decision offers to start a task for some duration on some cost schedule, and this corresponds directly to contract terms we must commit; and
- simulated annual review of each then-active task state can depend on any prior learning, but, more importantly, this dependence can be described in simple, intuitive terms of the facts already in hand for the review.

Table 3. Deterministic CPM gives a lower bound for each plan duration. Monte Carlo CPM, here using for each task an independent Weibull task time based only on that task's risk, shows the delaying influence of task time variability on the median project duration for 60,000 samples of each schedule plan. Deterministic optimized plans honor project budget goals and show the delaying influence of doing so. Optimized plans with Monte-Carlo annual reviews show the combined delaying effects of task time variability and budget goals. For reference, a start in January 2003 for 118 months yields a finish in October 2012.

Schedule Plan	Estimated FCS Program Durations in Months			
	No Budget Constraint		Project Budget Constraint by Fiscal Year Completed, and Allocated Yearly	
	Deterministic CPM	Monte Carlo CPM	Deterministic Optimized	Optimized with Monte Carlo Annual Reviews
Baseline	118	150	118	164
GAO Risk First	116	126	116	162
GAO C4ISR First	129	139	130	145

EPILOG

Time will tell whether our work proves prescient for FCS. We have delivered presentations to the Cost Analysis Improvement Group, Program Analysis and Evaluation, Office of the Secretary of Defense, and thank them, especially Mr. Walter Cooper, for their continued encouragement and support. Grose (2004) exhibits additional underlying detail. Since this writing, a number of revisions to the FCS program and its nominal schedule have already arisen, and these are reported in the open press, where we direct interested readers.

APPENDIX: THREE ALTERNATE PLANS FOR FCS SYSTEMS DEVELOPMENT AND DEMONSTRATION

Summary description tasks are in **bold text** (Microsoft, 2004). Zero-duration tasks are milestones.

Baseline Plan:

ID	Task Name	Estimated Cost ($M)	Duration (Weeks)	Successors
1	Notional Start	0.00	0	24,13,3
2	**Major Events**			
3	Milestone B Complete	0.00	0	4,67,37,29,25,14
4	SFR (System Functional Review)	0.00	0	5,16,26
5	SoS PDR Complete	0.00	0	6,17
6	SoS CDR Complete	0.00	0	7
7	Facilitation	0.00	0	8,95
8	LL IPR Waiver	0.00	0	9,97
9	IPD (Milestone C)	0.00	0	10,77
10	IOC	0.00	0	11,32
11	UA	0.00	0	101
12	**SoS Definition and Design**			
13	Systems Engineering	571.42	104	5
14	Systems Design	1,428.57	260	10
15	**Prototype Systems Build and Test**			
16	1st Variant PDC (Preliminary Design Complete)	0.00	0	17
17	Last Variant PDC (Preliminary Design Complete)	0.00	0	18,20,44
18	Long Lead Prototype	800.00	52	19,21
19	Prototype Integration and Assembly	1,200.00	78	22
20	First Variant CDC (Critical Design Complete)	0.00	0	69,21
21	Last Variant CDC (Critical Design Complete)	0.00	0	22,6
22	Final Prototype	0.00	0	97,8
23	**C4ISR Software and Platform**			
24	SW Build 1	507.93	104	27,44
25	SW Build 2	634.92	130	27,34,69,31,46,52,59
26	SW Build 3	825.39	169	28,52,59
27	SW Build 4	571.42	117	9,63,59
28	SW Build 5	507.93	104	83,89,64
29	SIL Delivery 1 (System Integration Lab)	253.96	52	68,33,30
30	SIL Delivery 2	253.96	52	69,31,27,52
31	SIL Delivery 3	253.96	52	32,28
32	SW Update	190.47	39	11,80
33	Software PDR Complete	0.00	0	34,5
34	Software CDR Complete	0.00	0	6
35	**Integrated Test Program**			
36	**IPS1 (Integration Phase SDD 1)**			
37	SoSIL Development	280.99	51	38,39,30
38	Integration	71.62	13	41,5,40
39	Sims Delivered	0.00	0	40
40	IT and UT	71.62	13	42
41	TRR	0.00	0	42
42	Analysis	71.62	13	45,44
43	**IPS2**			
44	Integration	280.99	51	47,6,46
45	Early Emulators Delivered	0.00	0	46
46	IT/UT	71.62	13	48
47	TRR	0.00	0	48
48	Analysis	71.62	13	50,51,28
49	**IPS3**			
50	Integration	209.36	38	53,52

APPENDIX: Continued

ID	Task Name	Estimated Cost ($M)	Duration (Weeks)	Successors
	Baseline Plan:			
51	Initial DP Prime Items Delivered	0.00	0	52
52	IT and UT	71.62	13	54,55
53	TRR	0.00	0	54,55
54	Analysis	104.68	19	58,8
55	User Trial	11.01	2	57
56	**IPS4**			
57	Integration	187.32	34	60,59
58	Initial System Deliveries	0.00	0	59
59	IT and UT	71.62	13	61,63,72
60	TRR	0.00	0	61
61	Analysis	71.62	13	9
62	**IPS5**			
63	Integration	209.37	38	64
64	IMT	71.63	13	65
65	Analysis	71.63	13	77,100
66	**SoS Testing and Integration**			
67	Phase 1: Integration and Test SDD (Simulation)	183.75	78	70,5
68	Phase 2: HW and SW	214.37	91	6,95
69	Phase 3: Prototype	214.37	91	72,57,8
70	Integration and Qualification and Live Fire Tests	489.99	208	73,9,76
71	**Test Events and Milestones**			
72	LUT 1	4.71	2	73
73	LUT 2	4.71	2	77,79,74,98,99
74	IOT (Initial Operational Test) Phase 1	47.11	20	10,75
75	IOT Phase 2	44.76	19	80
76	Integration and Test Production	214.37	91	10,80
77	FUSL	244.99	104	80,11
78	Training and Fielding	244.99	104	80
79	IOTE 1	61.25	26	80
80	IOTE 2	30.62	13	11
81	**Combat Systems Testing**			
82	**Phase 1: LRIP Prime Items**			
83	Integration	634.15	39	85,89,100
84	LRIP PI for SoSIL	0.00	0	85
85	LRIP PI for TFT Delivered	0.00	0	86
86	Testing	211.38	13	87,90
87	Analysis	211.38	13	92,74,79
88	**Phase 2: LRIP Late LRIP PI**			
89	Integration	520.33	32	91
90	LRIP PI for SoSIL	0.00	0	91
91	LRIP PI for TFT Delivered	0.00	0	92
92	Testing	211.38	13	93
93	Analysis	211.38	13	11,10,32
94	**Production**			
95	Facilitation (Pre-LL Production)	682.93	52	100,84,96
96	Facilitation (LL Production)	1,195.12	91	100,84
97	Long Lead Lot 1	682.93	52	98,99,100,9,83,84,76
98	Lot 1	1,024.39	78	79,78
99	Lot 2	1,707.32	130	11,80
100	Lot 3	1,707.32	130	11,80
101	Notional End Task	0	0	

APPENDIX: GAO "Risk First": Mitigate High Risk Technologies First

ID	Task Name	Estimated Cost ($M)	Duration (Weeks)	Successors
1	Notional Start	0.00	0	57,13,3
2	**Major Events**			
3	Milestone B Complete	0.00	0	4,100,70,62,58,14
4	SFR (System Functional Review)	0.00	0	5,49,59
5	SoS PDR Complete	0.00	0	6,50
6	SoS CDR Complete	0.00	0	7
7	Facilitation	0.00	0	8,128
8	LL IPR Waiver	0.00	0	9,130
9	IPD (Milestone C)	0.00	0	10,110
10	IOC	0.00	0	11,65
11	UA	0.00	0	134
12	**SoS Definition and Design**			
13	Systems Engineering	571.43	104	5
14	Systems Design	1,428.57	260	10
15	**Prototype Systems Build and Test**			
16	**TRL Mitigation (Technology Readiness Level)**			
17	**KPP 1: Joint Interoperability**			
18	Interface and Information Exchange	113.24	65	4
19	**KPP 2: Networked Battle Command**			
20	Security Systems and Algorithms	249.13	143	6
21	Quality of Service Algorithms	67.94	39	3
22	Wideband Waveforms	181.18	104	5
23	Multispectral Sensors and Seekers	90.59	52	3
24	Combat Identification	22.65	13	3
25	Sensor and Data Fusion and Data Compression	67.94	39	3
26	**KPP 3: Networked Lethality**			
27	Dynamic Sensor-Shooter Pairing and Fire Control	90.59	52	3
28	LOS and BLOS and NLOS Precision Munitions Guidance	271.78	156	6
29	Aided Target Recognition	67.94	39	3
30	Auto Target Recognition	181.18	104	5
31	Recoil Management and Lightweight Components	90.59	52	3
32	Distributed Collaboration of Manned and Unmanned Vehicles	226.48	130	5
33	Rapid Battle Damage Assessment	67.94	39	3
34	**KPP 4: Transportability**			
35	High Power Density and Fuel Efficient Propulsion	90.59	52	3
36	**KPP 5: Sustainability and Reliability**			
37	Embedded Predictive Logistic Sensors and Algorithms	90.59	52	3
38	Water Generation and Purification	90.59	52	3
39	**KPP 6: Training**			
40	Computer Generated Forces	22.65	13	3
41	Tactical Engagement Simulation	45.30	26	3
42	**KPP 7: Survivability**			
43	Active Protection System	22.65	13	3
44	Signature Management	90.59	52	3
45	Lightweight hull and Vehicle Armour	10.45	6	3
46	Power Distribution and Control	10.45	6	3
47	Advanced Countermine Technology	226.48	130	5
48	High Density Packaged Power	10.45	6	3
49	1st Variant PDC (Preliminary Design Complete)	0.00	0	50
50	Last Variant PDC (Preliminary Design Complete)	0.00	0	51,53,77

APPENDIX: Continued

ID	Task Name	Estimated Cost ($M)	Duration (Weeks)	Successors
107	IOT (Initial Operational Test) Phase 1	47.11	20	10,108
108	IOT Phase 2	44.76	19	113
109	Integration and Test Production	214.37	91	10,113
110	FUSL	244.99	104	113,11
111	Training and Fielding	244.99	104	113
112	IOTE 1	61.25	26	113
113	IOTE 2	30.62	13	11
114	**Combat Systems Testing**			
115	**Phase 1: LRIP Prime Items**			
116	Integration	634.15	39	118,122
117	LRIP PI for SoSIL	0.00	0	118
118	LRIP PI for TFT Delivered	0.00	0	119
119	Testing	211.38	13	120,123
120	Analysis	211.38	13	125,107,112
121	**Phase 2: LRIP Late LRIP PI**			
122	Integration	520.33	32	124
123	LRIP PI for SoSIL	0.00	0	124
124	LRIP PI for TFT Delivered	0.00	0	125
125	Testing	211.38	13	126
126	Analysis	211.38	13	11,10
127	**Production**			
128	Facilitation (Pre-LL Production)	833.33	65	129
129	Facilitation (LL Production)	1,166.67	91	133,117
130	Long Lead Lot 1	666.67	52	131,132,133,9,116,117
131	Lot 1	1,000.00	78	112,111
132	Lot 2	1,666.67	130	11,113
133	Lot 3	1,666.67	130	11,113
134	Notional End Task	0.00	0	

APPENDIX: GAO "C4ISR First": Develop C4ISR Infrastructure First

ID	Task Name	Estimated Cost ($M)	Duration (Weeks)	Successors
1	Notional Start	0	0	24,13,3
2	**Major Events**			
3	Milestone B Complete	0	0	4,67,37,29,25,14
4	SFR (System Functional Review)	0	0	5,26
5	SoS PDR Complete	0	0	6,16
6	SoS CDR Complete	0	0	7,17
7	Facilitation	0	0	8,95
8	LL IPR Waiver	0	0	9,97,21
9	IPD (Milestone C)	0	0	10,77
10	IOC	0	0	11,32
11	UA	0	0	101
12	**SoS Definition and Design**			
13	Systems Engineering	571.43	104	5
14	Systems Design	1428.57	260	10
15	**Prototype Systems Build and Test**			
16	1st Variant PDC (Preliminary Design Complete)	0	0	17
17	Last Variant PDC (Preliminary Design Complete)	0	0	18
18	Long Lead Prototype	800	52	19,20,21
19	Prototype Integration and Assembly	1200	78	22
20	First Variant CDC (Critical Design Complete)	0	0	95
21	Last Variant CDC (Critical Design Complete)	0	0	22
22	Final Prototype	0	0	57,69,97,96
23	**C4ISR Software and Platform**			
24	SW Build 1	507.94	104	27,44
25	SW Build 2	634.92	130	27,34,69,31,46,52,59
26	SW Build 3	825.4	169	28,52,59
27	SW Build 4	571.43	117	9,63,59
28	SW Build 5	507.94	104	83,89,64
29	SIL Delivery 1 (System Integration Lab)	253.97	52	68,33,30
30	SIL Delivery 2	253.97	52	69,31,27,52
31	SIL Delivery 3	253.97	52	32,28
32	SW Update	190.48	39	11,80
33	Software PDR Complete	0	0	34,5
34	Software CDR Complete	0	0	6
35	**Integrated Test Program**			
36	**IPS1 (Integration Phase SDD 1)**			
37	SoSIL Development	280.99	51	38,39,30
38	Integration	71.63	13	41,5,40
39	Sims Delivered	0	0	40
40	IT and UT	71.63	13	42
41	TRR	0	0	42
42	Analysis	71.63	13	45,44
43	**IPS2**			
44	Integration	280.99	51	47,6,46
45	Early Emulators Delivered	0	0	46
46	IT and UT	71.63	13	48
47	TRR	0	0	48
48	Analysis	71.63	13	50,51,28
49	**IPS3**			
50	Integration	209.37	38	53,52
51	Initial DP Prime Items Delivered	0	0	52
52	IT and UT	71.63	13	54,55
53	TRR	0	0	54,55
54	Analysis	104.68	19	58,8

APPENDIX: Continued

ID	Task Name	Estimated Cost ($M)	Duration (Weeks)	Successors
55	User Trial	11.02	2	57
56	**IPS4**			
57	Integration	187.33	34	60,59
58	Initial System Deliveries	0	0	59
59	IT and UT	71.63	13	61,63,72
60	TRR	0	0	61
61	Analysis	71.63	13	9
62	**IPS5**			
63	Integration	209.37	38	64
64	IMT	71.63	13	65
65	Analysis	71.63	13	77,100
66	**SoS Testing and Integration**			
67	Phase 1: Integration and Test SDD (Simulation)	183.75	78	70,5
68	Phase 2: HW and SW	214.37	91	6,95,57
69	Phase 3: Prototype	214.37	91	72
70	Integration and Qualification and Live Fire Tests	489.99	208	73,9,76
71	**Test Events and Milestones**			
72	LUT 1	4.71	2	73
73	LUT 2	4.71	2	77,79,74,98,99,76
74	IOT (Initial Operational Test) Phase 1	47.11	20	10,75
75	IOT Phase 2	44.76	19	80
76	Integration and Test Production	214.37	91	10,80
77	FUSL	244.99	104	80,11
78	Training and Fielding	244.99	104	80
79	IOTE 1	61.25	26	80
80	IOTE 2	30.62	13	11
81	**Combat Systems Testing**			
82	**Phase 1: LRIP Prime Items**			
83	Integration	634.15	39	85,89,100
84	LRIP PI for SoSIL	0	0	85
85	LRIP PI for TFT Delivered	0	0	86
86	Testing	211.38	13	87,90
87	Analysis	211.38	13	92,74,79
88	**Phase 2: LRIP Late LRIP PI**			
89	Integration	520.33	32	91
90	LRIP PI for SoSIL	0	0	91
91	LRIP PI for TFT Delivered	0	0	92
92	Testing	211.38	13	93
93	Analysis	211.38	13	11,10,32
94	**Production**			
95	Facilitation (Pre-LL Production)	682.93	52	100,84,96
96	Facilitation (LL Production)	1195.12	91	100,84
97	Long Lead Lot 1	682.93	52	98,99,100,9,83,84,76
98	Lot 1	1024.39	78	79,78
99	Lot 2	1707.32	130	11,80
100	Lot 3	1707.32	130	11,80
101	Notional End Task	0	0	

REFERENCES

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1993). *Network Flows: Theory, Algorithms and Applications,* Prentice Hall, Englewood Cliffs, NJ.

Babbar, M.M., Tintner, G., and Heady, E.O. (1955). "Programming with Consideration of Variations in Input Coefficients," *Journal of Farm Economics,* May, **37,** No. 2, pp. 333–341.

Bowman, E.H. (1958). "The Schedule Sequencing Problem," *Operations Research,* **7,** pp. 621–624.

Brady, E. (2003). *Future Combat Systems,* Modeling and Simulation Conference, Project Executive Office Ground Combat Systems, U.S. Army, 30 January 2003. Accessed 14 January 2004 via the World Wide Web at *http://www.boeing.com/defense-space/ic/fcs/bia/modsim_030130.html.*

Brook, A., Kendrick, D., Meeraus, A., and Raman, R. (1998). *GAMS: A User's Guide,* The Scientific Press, Washington, D.C.

Brown, G.G., Dell, R.F. and Newman, A.M (2004). "Optimizing Military Capital Planning," *INTERFACES,* **34,** pp. 415–425.

Brown, G.G., Dell, R.F., and Wood, R.K. (1997). "Optimization and Persistence," *INTERFACES,* **27,** pp. 15–37.

Demeulemeester, E., and Herroelen, W. (2002). *Project Scheduling: A Research Handbook,* Volume 49 of the International Series in Operations Research and Management Science, Kluwer Academic Publishing, Boston.

Dodin, B. (1984). "Bounding the Project Completion Time Distribution in PERT Networks," *Operations Research,* **33,** pp. 862–881.

Francis, P.L. (2004). "Defense Acquisitions: The Army's Future Combat Systems' Features, Risks and Alternatives," GAO-04-635T, Testimony before the Subcommittee on Tactical Air and Land Forces, Committee on Armed Services, U.S. House of Representatives, April.

Fulkerson, D.R. (1962). "Expected Critical Path Lengths in PERT Networks," *Operations Research,* **10,** pp. 808–817.

Grose, R.T. (2004). "Cost-constrained Project Scheduling with Task Durations and Costs That May Increase Over Time: Demonstrated with the U.S. Army Future Combat Systems," MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, March.

Gutjahr, W.J., Stauss, C., and Wagner, E. (2000). "A stochastic branch and bound approach to activity crashing in project management," *INFORMS Journal on Computing,* **12,** No. 2, pp. 125–135.

Herroelen, W., and Leus, R. (2004). "Robust and Reactive Scheduling: A Review and Classification of Procedures," *International Journal of Production Research,* **42,** No. 8 (15 April), pp. 1599–1620.

Herroelen, W., Reyck, B.D., and Demeulemeester, E. (1998). "Resource-Constrained Project Scheduling: A Survey of Recent Developments," *Computers & Operations Research,* **25,** No. 4, pp. 279–302.

Igelmund, G., and Radermacher, F.J. (1983). "Pre-selective strategies for the optimization of stochastic project networks under resource constraints," *Networks,* **13,** pp. 1–28.

ILOG (2004). ILOG Optimization Suite White Paper. Accessed 13 June 2004 via the World Wide Web at *http://www.ilog.com/products/optimization/papers.cfm.*

Kelly, J.E. (1961). "Critical Path Planning and Scheduling: Mathematical Basis," *Operations Research,* **9,** pp. 296–320.

Kelly, J.E. (1963). "The Critical Path Method: Resource Planning and Scheduling," in Muth, J.F., and Thompson, G.L. (Eds.), *Industrial Scheduling,* Prentice Hall, Englewood Cliffs.

Law, A.M. and Kelton, W.D. (2000). *Simulation Modeling and Analysis,* third ed., McGraw-Hill, New York.

Lee, D. (1997). *The Cost Analyst's Companion,* Logistics Management Institute, McLean, VA.

Malcolm, D.G., Roseboom, J.H., Clark, C.E., and Fazar, W. (1959). "Application of a Technique for Research and Development Program Evaluation," *Operations Research,* **7,** pp. 646–669.

Microsoft Corporation (2004). "Microsoft Office." Accessed 10 February 2004 via the World Wide Web at *http://office.microsoft.com/home/office.aspx?assetid=FX01085795.*

Miller, S. (2003). "Simulation Based Schedule Analysis," In *Proceedings of the Department of Defense Cost Analysis Symposium,* Washington, D.C. Accessed 12 December 2003 via the World Wide Web at *http://www.ra.pae.osd.mil/adodcas/index2.htm.*

Pritsker, A.B., Watters, L.J., and Wolfe, P.M. (1969). "Multi-Project Scheduling with Limited Resources, A Zero-One Programming Approach," *Management Science,* **16,** pp. 93–108.

Sengupta, J.K., Tintner, G., and Morrison, B. (1963). "Stochastic Linear Programming with Applications to Economic Models," *Economica,* **30,** pp. 262–275.

Senju, S., and Toyoda, Y. (1968). "An Approach to Linear Programming with 0–1 Variables," *Management Science,* **15,** No. 4, pp. 196–207.

Sun Microsystems (2005). "Java technology." Accessed 28 May 2005 via the World Wide Web at *http://java.sun.com.*

Tintner, G. (1955). "Stochastic Linear Programming with Applications to Agricultural Economics," In Antosiewicz, H.A., *Second Symposium on Linear Programming,* National Bureau of Standards, Washington, D.C., 27–29 January.

Tintner, G. (1960). "A Note on Stochastic Linear Programming," *Econometrica,* **28,** No. 2, pp. 490–495.

U.S. General Accounting Office (2003). *"Issues Facing the Army's Future Combat Systems Program,"* GAO-03-1010R, Washington, D.C., August.

Wiest, J.D. (1964). Some Properties of Schedules for Large Projects with Limited Resources, *Operations Research,* **12,** pp. 395–418.

Welch, L.D. (2003). *Objective Force and Future Combat Systems Independent Assessment Panel Report,* Institute for Defense Analysis, Alexandria, VA.

Wynne, M.W. (2003). *Operation of the Defense Acquisition System,* DoD Instruction 5000.2, Office of the Principal Deputy Under Secretary of Defense for Acquisition, Technology and Logistics, U.S. Department of Defense, 12 May.

Yang, B., Geunes, J., and O'Brien, W.J. (2001). *Resource-Constrained Project Scheduling: Past Work and New Directions,* Research Report 2001–6, Department of Industrial and Systems Engineering, University of Florida.