Strategic management of complex projects: a case study using system dynamics

James M. Lyneis,[a]* Kenneth G. Cooper[a] and Sharon A. Els[a]

James M. Lyneis works in the Business Dynamics Practice of PA Consulting Group and is a Senior Lecturer at MIT. Prior to joining PA in 1978, he was an Assistant Professor at MIT's Sloan School of Management. Dr Lyneis has a Ph.D. in Business Administration from the University of Michigan and undergraduate degrees in Electrical Engineering and Industrial Management from MIT.

Kenneth G. Cooper is a member of the Management Group of PA Consulting and leads the Business Dynamics Practice within PA. His management consulting career spans 25 years, in which he has directed over 100 consulting engagements, among them analyses of 60 major commercial and defense development projects. His clients include AT&T, Aetna, Arizona Public Service, Hughes Aircraft, IBM, Litton, MasterCard, McDonnell-Douglas, Northrop, Rockwell, and several law firms. Mr Cooper received his bachelor's and

Abstract

System dynamics models have been used extensively over the last 20 years on complex development projects and have proven their value in contributing to significantly improved project performance. System dynamics models facilitate the strategic management of projects, including planning the project (setting the initial schedule and budget, the organization structure, process model, etc.), determining measurement and reward systems, evaluating risks, and learning from past projects. The use of system dynamics for strategic project management is illustrated with a case study of the Peace Shield Air Defense System. On this project, the model was used to support the project bid, to identify and manage risks, and to assess the benefit of several process and organization changes which were implemented on the project. Upon completion, the project results were systematically compared to an earlier project to assess the management lessons—what worked and what did not, and what was the benefit. These lessons were systematized in a management learning system. Copyright © 2001 John Wiley & Sons, Ltd.

Syst. Dyn. Rev. **17**, 237–260, (2001)

Introduction

The challenge—project performance problems

Research, design, and development projects are the lifeblood of many organizations. This is true not only for "project-based" organizations such as large aerospace or civil construction companies, but also for companies that depend on a flow of new products and services to remain competitive, such as in the automotive, electronics, and software industries. Delivering new products and services on time and in budget increasingly determines success or failure.

Yet most large, complex development projects experience substantial cost and schedule overruns. A review by Morris and Hough (1987; p. 7) of some 3500 projects revealed that "overruns are the norm, being typically between 40 and 200 percent." A survey by Roberts (1992) of corporate R&D projects found that less than half met their time-to-market and budget objectives. Recent advances in project management techniques do not seem to have improved the situation significantly. A survey by the Defense Systems Management College at Fort Belvoir, Virginia, indicated that the average cost overrun for engineering and manufacturing development of a major system was 45 percent, and the

* Corresponding author.
[a] Business Dynamics Practice, PA Consulting Group, One Memorial Drive, Cambridge MA 02142, U.S.A.
Business Dynamics Practice, PA Consulting Group (formerly Pugh-Roberts Associates)

System Dynamics Review Vol. 17, No. 3, (Fall 2001): 237–260
DOI: 10.1002/sdr.213
Copyright © 2001 John Wiley & Sons, Ltd.

Received August 2000
Accepted April 2001

master's degrees from MIT and Boston University, respectively.

Sharon A. Els is a Managing Consultant in PA's Business Dynamics Practice. Since joining PA in 1989, Ms Els has managed and participated in many consulting assignments for clients in the financial services, civil construction and software development industries. She received an S.B. degree in Civil Engineering and an S.M. in Management Science from MIT.

schedule overrun was 63 percent (Kausal 1996). In a sample of ten projects by Reichelt and Lyneis (1999), the average budget overrun was 86 percent (66 percent when the cost of added work scope is removed), and schedule overrun 55 percent. Project problems occur in the non-profit world as well. A World Bank (1992) survey of its projects found that only 70 percent of recent projects had been rated "satisfactory", with only one-third substantially achieving institution development goals and with delays in completion averaging 50%.

Problems of cost and schedule overrun on projects have persisted for decades, in spite of numerous advances in the field of "project management". In the 1950s, the static network modeling approaches PERT and the critical path method (CPM) were developed. These have continued to evolve with the addition of probabilistic parameter estimates and integration with resource loading assessments. Alternative approaches to software development such as the waterfall and spiral methods have been adopted. Finally, teaming, concurrent engineering, and the recognition and emphasis on "soft" and people factors have emerged as methods of enhancing project performance. The annual market for project management software has been estimated to exceed $1 billion (Shtub *et al.* 1994). The Project Management Institute, dedicated to the improvement of project management, boasts a membership of 39,000 worldwide.

Why do projects continue to perform poorly in spite of these advances and the substantial effort on tools and techniques? We believe that a major reason for continued schedule and budget performance problems is that while projects are fundamentally complex dynamic systems, most project management concepts and tools either (1) view a project statically or (2) take a partial, narrow view in order to allow managers to cope mentally with the complexity (for example, analyzing design functions individually or having separate focuses on soft and hard factors, when all are simultaneously important). Traditional tools and mental models are inadequate for dealing with the dynamic complexity of projects. In addition, these tools foster the perception that each project is unique, which makes systematic learning across projects difficult. As a result, managers continue to make mistakes. While some learn from experience, these lessons are not effectively passed to subsequent generations of managers.

Typical project dynamics

Figure 1 illustrates the nature of project dynamics. In the ideal project, staffing follows the plan and increases to a peak, then falls off as the work is completed. In reality, however, project staffing is often slower to build up than planned (delays in getting approvals, difficulties finding staff) and frequently exceeds planned levels for an extended period (the overrun). Often, there is a second "peak". On most projects, managers also assume, either explicitly or implicitly, that productivity will remain constant over the duration of the project. In

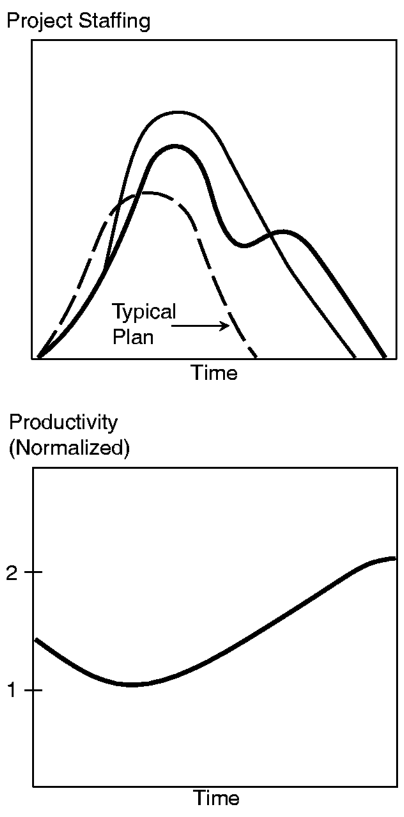


Fig. 1. Typical project dynamics

reality, productivity typically falls from the beginning through the middle of the project, before rising at the end. Productivity often varies by a factor of two over the course of a project. Examples of such budget and schedule overrun behavior from real projects are shown in Figures 2 and 3.

System dynamics applied to project management

Given the dynamic nature of project problems, it is not surprising that system dynamics has been applied to understanding and improving the behavior of complex projects. In fact, without doubt project management applications have been the most extensive and successful use of system dynamics. In terms of numbers of applications, consulting revenues, and value to clients, system dynamics project modeling far and away exceeds all others.

Perhaps the best known of the project applications has been in the resolution of cost-overrun and schedule disputes. Beginning with the groundbreaking work for Ingalls Shipbuilding in the late 1970s (Cooper 1980; Sterman 2000),

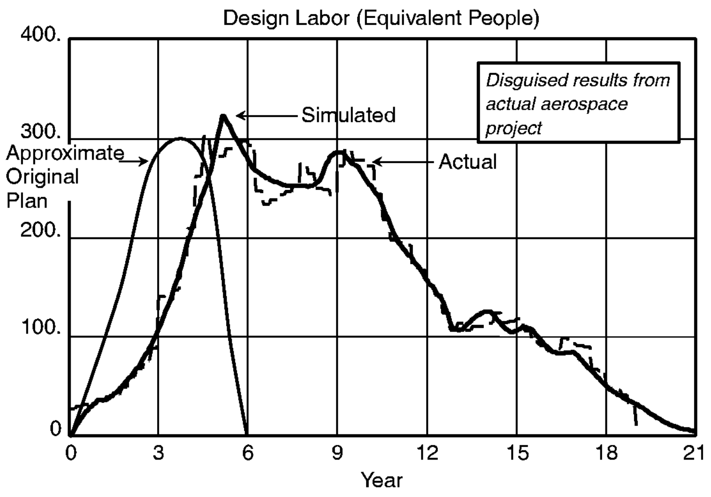


Fig. 2. A typical aerospace project with a long staffing "tail"

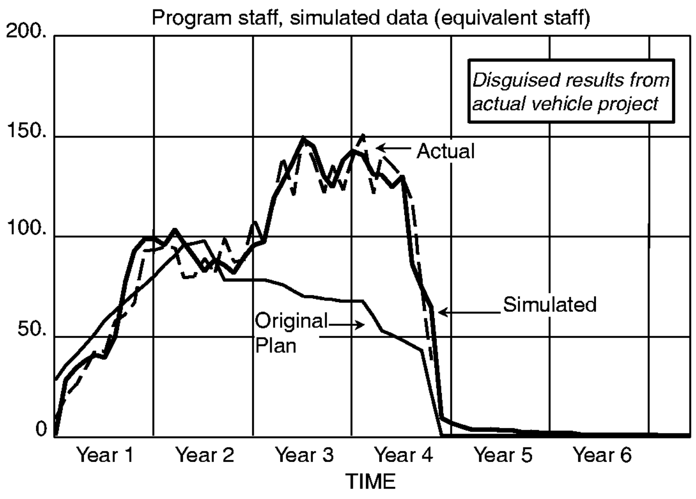


Fig. 3. A typical product development project with significant overstaffing in later stages

Pugh-Roberts/PA Consulting has applied system dynamics in more than 30 contract disputes. The approximate value of these disputes is in excess of $4 billion, with an average recovery of 75 percent using system dynamics versus 40 percent with traditional approaches. Much more extensive but less well known or documented, however, is the application of system dynamics to "strategic" project management (including the avoidance of cost and schedule overrun and resultant disputes). Pugh-Roberts/PA Consulting alone has applied system dynamics modeling in a "proactive" way to more than 75 different design, construction, and development projects. These include projects in aerospace (missiles, aircraft), civil construction (nuclear power plants, Channel Tunnel), shipbuilding (both military and commercial), software (telecom switching systems, air traffic control, air defense systems), and product development (vehicle development). These projects have generated consulting fees in excess of $25 million, while conservatively saving clients more than $5 billion on improved project budget performance. Savings from earlier completion and higher delivered quality further increase the benefit to clients. In addition to Pugh-Roberts/PA Consulting, many others have applied system dynamics to improving project performance, including Abdel-Hamid and Madnick (1991), Homer *et al.* (1993), Ford and Sterman (1998), and Rodrigues and Williams (1998).

While some of the work by Pugh-Roberts/PA Consulting is touched on in Cooper and Mullen (1993) and Reichelt and Lyneis (1999), the full breadth of system dynamics use in project management has not been described in detail. This article therefore discusses the role of system dynamics models in the proactive, strategic/tactical management of design, construction, and development projects, and illustrates with a case study of the Peace Shield Air Defense System.

Strategic project management

The types of decisions made on projects are often categorized as being strategic, tactical, or operational. The use of system dynamics most naturally falls into the strategic/tactical end of the spectrum. But what is strategic project management? In one view, strategic project management involves determining the fit of specific projects in achieving the strategy of the company (what products and enhancements, when what technology and so on?). However, these questions are more a part of strategic *company* management than strategic *project* management. In our view, strategic *project* management covers decisions that are taken up front in designing the project, and then the guidance provided to operational decisions that considers the longer-term impact of these decisions on downstream performance of the project. Specifically, strategic project management involves:

- *Designing the project*. Strategic design entails setting the initial schedule and budget, selecting a process model (e.g., waterfall vs. spiral) and organization structure (e.g., functional vs. integrated team), establishing appropriate buffers, and determining overlap/concurrency between phases of work, so as to give the best chance of successfully meeting the project's company–strategic objectives. While the scope, schedule and budget may be beyond the direct control of project management, the consequences of alternatives can be communicated to company management and negotiated.

- *Determining what indicators to measure, monitor, and exert pressure on.* What is measured and rewarded drives behavior. If adherence to schedule is all that is monitored, then people are likely to make efforts to meet the schedule regardless of its impact on quality or even cost (particularly if the costs are hard to see, such as long hours and overtime). If too many conflicting factors are monitored (e.g., cost, schedule, and quality), then either staff become demoralized and ignore them all or they shift emphasis from one to another in response to the current crisis. Designing a reward system in advance can help assure achievement of the project's strategic objectives.
- *Risk management.* Specification or scope changes, design difficulties, risks such as delays in getting staff from other projects, labor shortages, late designs or material deliveries from other programs or vendors, and other similar problems often impact a project. While decisions must ultimately be taken as actual conditions evolve, managers can more quickly and effectively handle change if they have determined in advance which risks pose the greatest threat to the project, what should be monitored to provide early warning of each risk, and the best responses to such potential changes. In some cases, actions developed for specific risks may improve performance under normal circumstances and should be built into the project from the beginning (for example, integrated product design and "teaming").
- *Incorporating learning from past projects.* Based on benchmarking and other analyses of past projects, how can we better design and then manage this and future projects? This requires determining what really happened in terms of cost, schedule, and rework on prior projects; what risks actually occurred; and what management initiatives worked and what did not.
- *Making mid-course corrections.* Mid-project changes in project schedules, staffing, etc. in response to actual progress and external conditions will generally be required. These tactical responses need to consider the indirect and long-term impacts of the corrections in order to avoid potentially disastrous downward spirals (see discussion of model).

In supporting strategic project management along these dimensions, system dynamics has been used by Pugh-Roberts/PA Consulting (and others) in the following specific ways on actual projects.

Pre-project

BID OR PLAN ANALYSIS The model is used to establish and/or test the feasibility of schedule and budget given scope and other strategic requirements. Ideally, a model of an ancestor program is first used to determine the characteristics of a typical project in the organization, including normal productivity, rework, management practices, etc. The model is then adapted to the scope and anticipated external conditions of the proposed project, and used to assess cost/schedule tradeoffs for the proposed project. If a bid has already been submitted, the model is used to assess the assumptions required to make the bid, e.g., productivity, rework, external conditions, and actions to bring the project in as close as possible.

Competitor analysis Publicly available information is used in conjunction with the simulation model to estimate what the program might cost a competitor, and therefore provide a range of possible competitive bids.

Risk analysis The model is used to determine the impact of possible changes in external conditions on the performance of the project versus the bid or plan. A simulation that reflects the project plan provides a baseline against which alternatives are measured. The direct impacts of possible changes in external conditions (specification or scope changes, design difficulties, risks such as labor shortages, late vendor design or material deliveries, etc.) are input to the model (alone and in combination) and simulation results compared to the baseline. Risks of high probability and/or those producing a significant overrun for the project warrant careful monitoring and/or mitigating actions.

Mitigation analysis The model is used to determine changes in program schedules, interim milestones, resourcing, etc., which minimize the consequences of risks.

During the project

Risk management The model is used to determine the impact of project risks that actually materialize. First, the baseline simulation is compared to a simulation in which the direct impacts of the risks are included. Then, the model is used to determine changes in the program that minimize the consequences of that specific risk (for example, changes in schedule duration, interim milestones, phase overlap, additional staff, new processes or methods).

Change management Change management is a subset of risk management, but where the changes (usually scope increases or design changes) are often at the request of an external customer and therefore generally involve adjustments to the contracted cost and schedule. The model is used to determine the likely full cost and schedule implications of specification and scope changes by comparing two simulations: the baseline simulation (or current simulation of project) and a simulation in which the direct impacts of the changes are included. The latter simulation determines the indirect, secondary and tertiary effects of the direct impact on the project. These results are used as the basis for negotiating reasonable compensation and/or for designing actions to mitigate the cost/schedule impact of the change.

EVALUATION OF PROCESS CHANGES The model is used to assess the total impact of process or organizational changes, such as computerized design, new tools, integrated product design, and teaming. Implementation of change is often disruptive and involves short-term costs before long-term benefits are realized. Without an analysis of these dynamics, change programs are often abandoned before they have a chance to succeed. See Sterman *et al.* (1997) for an example of this.

Post-project

BENCHMARKING AND EVALUATION OF BEST PRACTICES Without a simulation model, it is very difficult to compare the performance of projects in a meaningful way. How much of the difference results from different products? From different external conditions? From different management practices? With a model, it becomes possible to answer these questions. First, models are developed and calibrated to the different projects. Then, differences in external conditions and scope/complexity are removed. What remains can be attributed to differences in management actions. The improvements from specific actions are further assessed by changing the actions in the simulation and comparing the performance of the simulated projects.

TRAINING AND DEVELOPMENT The training and development of future project managers can be enhanced through the use of simulation models of projects. First, models are used as "flight simulators" to allow practice and learning. Second, the lessons about what works are inferred from past projects, as described above, and communicated to the next generation of managers.

Many of these model uses were performed for the Peace Shield Program described in the remainder of the article.

The Peace Shield program and model

Project background

The Peace Shield Weapon System was a program undertaken by Hughes Aircraft Company for the U.S. Air Force on behalf of the Kingdom of Saudi Arabia. Hughes, now a part of Raytheon Corporation, won a competitive bid for the program after the Air Force terminated another contractor for default. The estimated value of the contract was more than $1 billion with final delivery scheduled for 3 January 1996. This schedule required a 54-month design, development, and testing effort. While many people in the Air Force considered this schedule to be impossible, with some estimates as high as 116 months, Hughes contracted for the 54 months with a $50 million bonus should Hughes achieve a three-months-early delivery.

The Peace Shield program involved both hardware and software. It "required delivery of a nationwide ground-air defense and command, control, and communications system to the Saudi Air Force. Key elements included 17 radar installations, a central command operations center, five sector command and operations centers, nationwide communications links, interfaces with all agencies having a role in national defense, and communications centers to contact and control civil and military aircraft." (Kausal 1996).

The Peace Shield program was not the first program of this type for Hughes. During the 1980s, the company had developed similar systems for NATO (Northern European Command and Control System) and Egypt. Pugh-Roberts/PA Consulting used a system dynamics model to advise Hughes during several periods on the Peace Shield project. The next section describes the Peace Shield model and the final section how the model was used in support of the program.

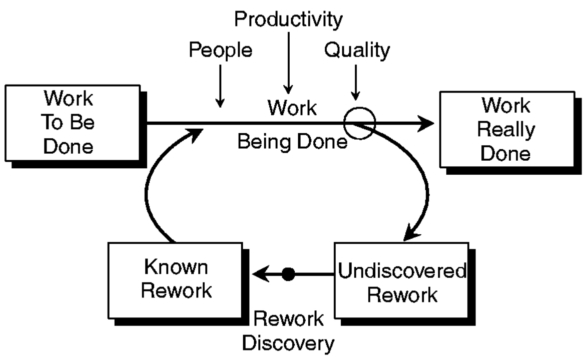
Model structure

THE REWORK CYCLE There are three important structures underlying the dynamics of a project:

- the work accomplishment structure, now referred to as the "rework cycle";
- feedback effects on productivity and work quality;
- knock-on effects from upstream phases to downstream phases.

The rework cycle, illustrated in Figure 4, was first developed by Pugh-Roberts/PA Consulting on a delay and disruption claim model for the Ingalls

Fig. 4. The work accomplishment or rework cycle structure



Shipbuilding Division of Litton Industries (Cooper 1980), and refined over many subsequent applications (Cooper 1993a; b; c). Almost all dynamic-project models have a rework cycle in some form (Abdel-Hamid, 1991; Ford and Sterman, 1998; Homer *et al.* 1993).

As shown in Figure 4, the rework cycle consists of four stocks of work. At the start of a project or project stage, all work resides in the stock Work To Be Done. As the project progresses, changing levels of staff (People) working at varying rates of Productivity determine the pace of Work Being Done. Work Being Done initially depletes the stock of Work to be Done, and later the stock of Known Rework. Work is executed at varying, but usually less than perfect, Quality. Quality represents the fraction of the work being done at any point in time that will enter the stock Work Really Done and which will never need re-doing. The rest will subsequently need some rework and flows to the stock of Undiscovered Rework—work that contains as yet undetected errors. Errors are detected in the normal course of work and as the result of downstream efforts or testing; Rework Discovery may occur months or even years after the rework was created, during which time dependent work has incorporated these errors or technical derivations thereof. Once discovered, the stock Known Rework demands the application of resources beyond those needed for executing remaining Work to be Done. Rework is executed at the productivity and quality levels then prevailing (although the inherent level of effort required for rework may be more or less than that for initial work).

Some re-worked items will flow through the rework cycle one or more subsequent times. A benchmarking analysis by Cooper and Mullen (1993) indicated that the average for 14 commercial software projects was 1.5 cycles, and for seven defense software projects was three cycles. As a result of this cycling, rework can increase significantly in the middle and at the end of a project. Figure 5 illustrates this phenomenon for the automotive project—the application of resources to execute rework is the source of overrun on this and many projects.

Feedback effects on productivity and quality Numerous feedback effects, illustrated in Figure 6, surround the rework cycle. Some of these feedbacks are “negative” feedbacks used by management to control resourcing on a project. In Figure 6, for example, overtime is added and/or staff are brought on to a project (“hiring”) based on work believed to be remaining (expected hours at completion less hours expended to date) and scheduled time remaining to finish the work. At the beginning of the project, staff are needed to get the work done and so hiring increases. As staff increases, work gets done. Eventually, enough work is accomplished and staffing needs are less than current staff, such that reductions in staff assigned to the project occur.

Other feedback effects drive productivity and quality. As illustrated earlier in stylized fashion in Figure 1, productivity and quality change significantly

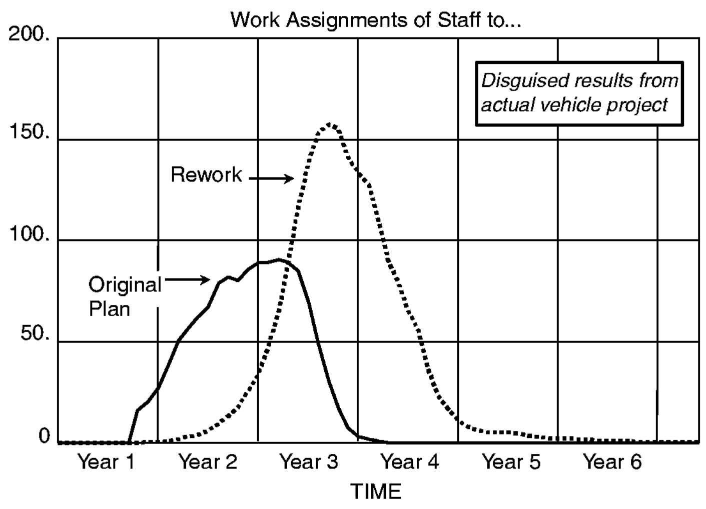


Fig. 5. Effort on rework is responsible for the extended "tail"

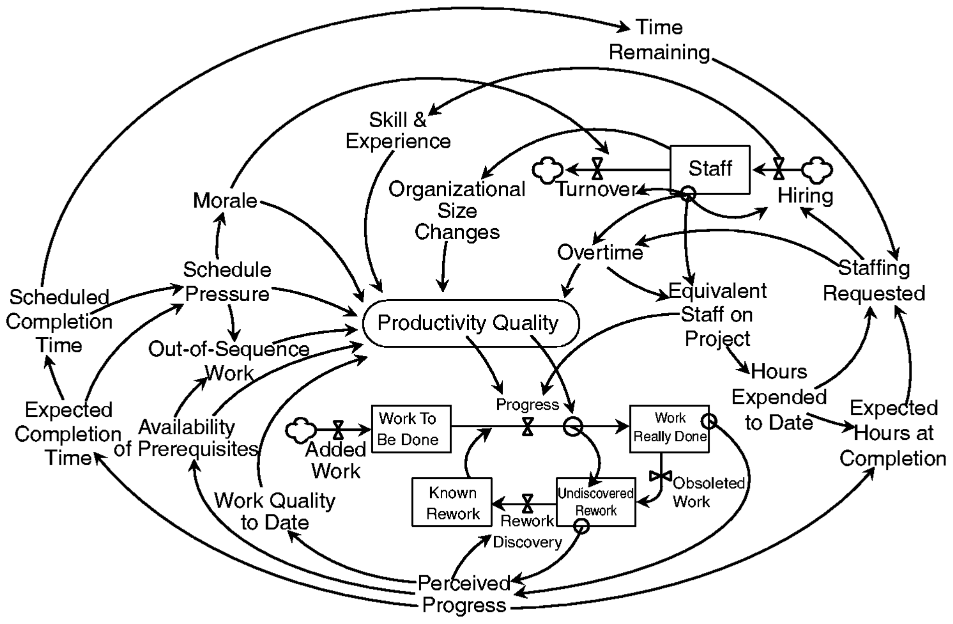


Fig. 6. Project dynamics

during the course of each phase of project work, often by factors of two or more. Figure 6 shows some of the structural reasons for the dynamic behavior of productivity and quality. Productivity and quality are affected by work quality to date, availability of prerequisites, out-of-sequence work, schedule pressure, morale, skill and experience, organizational size changes, and overtime. Each of these effects, in turn, is a part of a complex network of generally reinforcing feedback loops that early in the project drive productivity and quality down and later cause it to increase.

On a project, if things go according to "plan", staffing increases smoothly to its planned peak, and decreases smoothly to the scheduled completion. Often, however, a project does not go as planned. Sometimes, the plan is infeasible and there is an inconsistency between scope, budget, and schedule. In other cases, unanticipated problems or changes occur. Whatever the source, the unexpected often initiates a series of dynamics that can create substantial cost and schedule overrun. For example, as illustrated in Figure 6, assume that a mid-project design change:

- adds scope and therefore work to be done;
- obsoletes work already done;
- causes work to be done out-of-sequence (as a result of adding or obsoleting work, upstream work products are often no longer complete or correct).

Out-of-sequence work causes a reduction in productivity and/or quality (errors are made as a result of incorrect or incomplete upstream products).

As a result of the design change (or because of an inconsistent plan), the project is likely to fall behind schedule. In response, the project may bring on more resources. However, while additional resources have positive effects on work accomplished, they also initiate negative effects on productivity and quality. Bringing on additional staff reduces average experience level. Less experienced people make more errors and work more slowly than more experienced people. Bringing on additional staff also creates changes in the size of the organization, which in turn reduces productivity and quality while communication and reporting channels are re-established and while additional space and equipment are arranged. Finally, while overtime may augment the effective staff on the project, sustained overtime can lead to fatigue, which reduces productivity and quality.

Because of these "secondary" effects on productivity and quality, the project will make less progress than expected and contain more errors—the availability and quality of upstream work has deteriorated. As a result, the productivity and quality of downstream work suffers. The project falls further behind schedule, so more resources are added, thus continuing the downward spiral.

In addition to adding resources, a natural reaction to insufficient progress is to exert "schedule pressure" on the staff. This often results in more physical output, but also more errors ("haste makes waste") and more out-of-sequence work. Schedule pressure can also lead to lower morale, which also reduces productivity and quality and increases staff turnover.

We could continue adding feedback effects on productivity and quality. In addition to the effects noted above, typical influences on a project included in a model are:

- baseline design quality (clarity of initial specifications and pre-requisite design);
- availability and quality of procured design and/or products;
- availability of customer-furnished information and/or equipment;
- adequacy of supervision;
- space constraints;
- management concern for quality;
- managerial continuity and experience.

The many productivity and quality feedback effects in combination with inadequate plans, typical startup problems, and/or mid-project changes can cause productivity and quality to be low and/or deteriorate in the early and middle stages of a project, before increasing at the end. Minimizing the degradation of productivity and quality on a project is the key to minimizing cost and schedule delay. This requires understanding the dynamic drivers of project performance.

PHASE-ON-PHASE KNOCK-ON A rework cycle and its associated productivity and quality effects form a "building block". Building blocks can be used to represent an entire project or replicated to represent different phases of a project, in which case multiple rework cycles in parallel and series might be included. At its most aggregate level, such building blocks might represent design, build and test. Alternatively, building blocks might separately represent different design stages (e.g., conceptual vs. detail) and/or design functions (structural, electrical, power, etc.). In software, building blocks might represent specifications, detailed design, code and unit test, integration, and test.

When multiple phases are present, important additional productivity and quality effects result from interactions between the phases, specifically:

- availability of design products from upstream phases;
- quality of design products from upstream phases.

Problems early in a project can therefore "knock-on" to create later problems in downstream phases. In addition, downstream progress affects upstream work by fostering the discovery of upstream rework.

The building blocks in the Peace Shield model are illustrated in Figure 7. Because hardware was not anticipated to be a problem, the model represented only the software development effort in six phases: SRS (specifications and requirements), Detailed Design, Code and Unit Test, Integration and Type II test, Type I test, and "KOSA" (Kingdom of Saudi Arabia) final assembly and test.

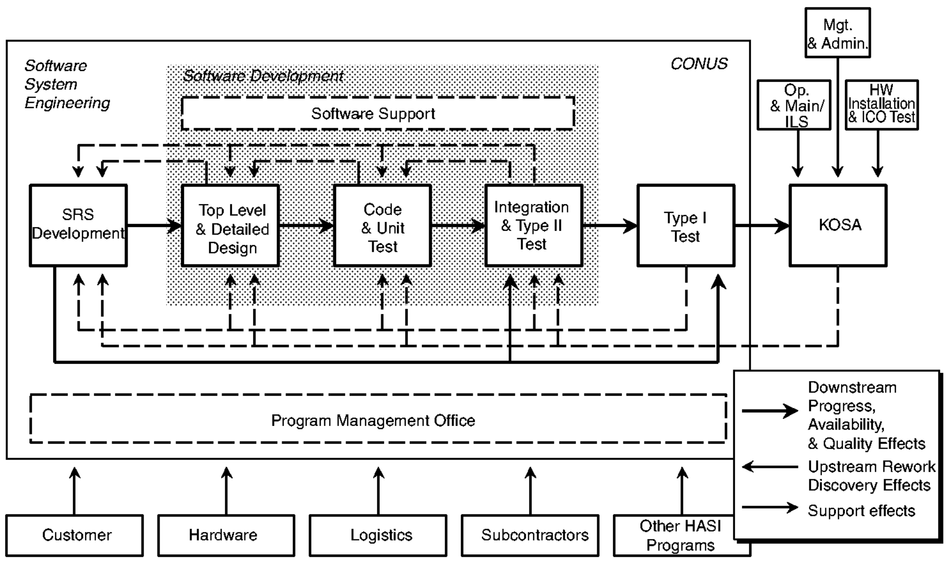

Software System Engineering
Software Development
CONUS
Software Support
SRS Development
Top Level & Detailed Design
Code & Unit Test
Integration & Type II Test
Type I Test
KOSA
Mgt. & Admin.
Op. & Main/ ILS
HW Installation & ICO Test
Program Management Office
Downstream Progress, Availability, & Quality Effects
Upstream Rework Discovery Effects
Support effects
Customer
Hardware
Logistics
Subcontractors
Other HASI Programs

Fig. 7. Phases in the Peace Shield program

Use of the model on the Peace Shield program

Bid support and risk assessment

In May 1991, Pugh-Roberts/PA Consulting assisted Hughes in developing its bid for the project and in assessing risks of alternative assumptions underlying that bid. Because a model of a prior, similar program had been developed, we were able to quickly adapt that model to represent the Peace Shield program. This process entailed:

- changing the work scope, technical complexity, milestones, and schedules from the prior program to estimates for Peace Shield;
- eliminating from the prior model any external events which impacted that program (design or scope changes, labor constraints, supplier problems, etc.);
- adding factors specific to Peace Shield (the most important of these was the use of software code intended to be used verbatim or "lifted" from an earlier program).

These adjustments were made based on interviews and discussions with Hughes managers.

The adjusted simulation provided a "Base" against which risks could be assessed. We then tested the likely cost and schedule impact of critical assumptions regarding the degree of liftability (fraction of code that does not require change and can be used as is), availability of experienced staff, vendor delays, and so on. As an example, Figure 8 summarizes the results from the "liftability" experiments. The Peace Shield program was using some code from a prior project. The Base case assumption was that 80 percent of that lifted code would be usable without change and that 20 percent would require rework. Figure 8 indicates that development cost increases significantly if more of the code requires rework.

While other assumptions impacted the program, liftability was determined to be one of the most critical and uncertain areas. Given this sensitivity, analyses were done to find effective mitigation efforts. The most effective mitigation was found to be concentrating early program efforts on flushing out as-yet-undiscovered rework in the lifted work, rather than steeply ramping up staff and effort to accomplish new work, as would normally be done. While slowing *measured* progress early in the project, the emphasis on rework discovery improves the baseline off which the new work will be done, with attendant beneficial effects on downstream productivity, quality, and progress.

In addition to assisting in Hughes' bid and project design, the model was adapted to determine a likely range for competitor bids. The other bidder was a team without any experience in programs of this type. Therefore, it was important to consider how this inexperience might lead them to view the project. Based on publicly available information and the estimates of experienced Hughes managers, the model was set up to represent the situation at the competitor, including likely productivity and work quality. Then, two scenarios for the competitor's bid were developed:

- the "naïve" competitor scenario, in which they significantly underestimate the scope and customer changes likely to occur as the program progressed;
- the "realistic" competitor scenario, which reflected the scope and customer changes which Hughes thought the program would actually operate.

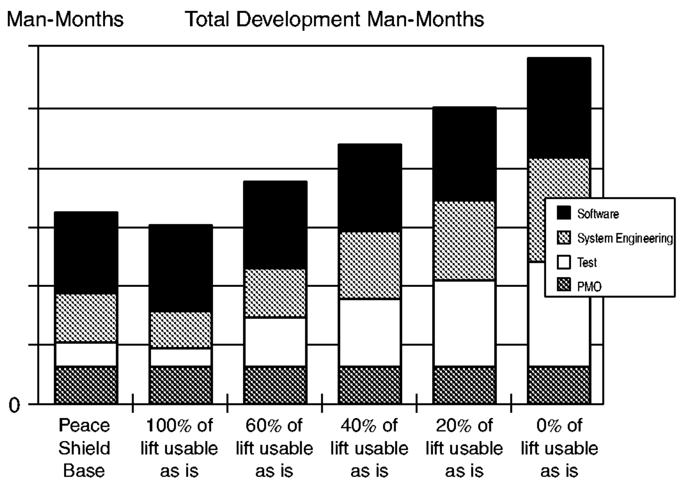


Fig. 8. Peace Shield sensitivity to "liftability" from prior program

The first scenario was likely to produce a bid by the competitor that Hughes would find difficult to meet profitably without aggressive actions at the start of the program. However, the actual cost of the program under the more likely second scenario would be nearly double that of the first scenario. Hughes challenge was to submit a winning but profitable bid and program design, under these circumstances.

Based on these analyses, our recommendations to Hughes in preparing their bid were:

1. Submit a bid that is likely to somewhat higher than the Scenario 1 competitor bid above, but which can profitably be met by Hughes if the program is managed aggressively (see point 4 below);
2. Carefully define in the bid and contract Hughes' understanding of the work scope, such that any future changes are clearly additional work that can be compensated for;
3. Stress Hughes' experience in this type of project and therefore the likely reliability of their bid;
4. Aggressively manage the program from day one to control costs: staff the program with the most experienced technical and managerial people; staff up in a controlled fashion, focusing first on discovering rework in the lifted code; and carefully manage customer expectations so as to minimize changes.

All of these were incorporated into the bid and the design of the project.

Ongoing project management

Pugh-Roberts/PA Consulting supported management of the Peace Shield project on a regular basis during the execution of the work. The model was updated as data about program performance and external conditions evolved. In addition to decisions regarding the early emphasis on rework discovery and the use of experienced staff, analyses supported a number of important decisions taken during the course of the project:

1. *Implementation of a "teaming" structure and other improved processes for the project.* Historically, each phase of a development effort was conducted exclusively by the staff specializing in that area, in a so-called "waterfall" approach. Approximately one year into the Peace Shield project, a new project manager took over and advocated a "teaming" style of project execution along with other process changes, including detailed progress reporting metrics (including monitoring of rework discovery) and customer involvement in design reviews. As it applied to Hughes, "teaming" meant having both upstream and downstream staff specialists involved in designing and reviewing the work in a given stage via participation in

project reviews and other meetings, and reviewing interim work products. The implementation of teaming and other process improvements was tested on the Peace Shield model. Teaming was assumed to have a significant short-term disruptive impact on productivity during implementation, followed by a less significant longer-term reduction in productivity (more people involved, reviews, disruption), but with resultant improvement in quality of work and reduction in rework discovery time. These "direct impacts" of teaming were estimated by Hughes staff and input to the model. The model then simulated the secondary and tertiary benefits that accrued later as a result of the teaming initiative. The resultant total impact was computed by the model to be significant. In spite of initial slowing of progress due to lower productivity, the project would finish three weeks earlier under the most pessimistic assumptions about benefits and 18 weeks earlier under the most optimistic. Teaming was therefore felt to be a relatively low risk means of improving the schedule and buffering against unforeseen future problems.

2. *Implementation of a different staffing strategy for software engineering and software coding.* As completion of the initial tasks on a phase of work winds down, the tendency is to release the staff on that phase to make them available for other projects or for downstream activities. However, we have continually found that a better strategy on complex projects is to slow the staff "roll-off" and assign the extra staff to flushing out undiscovered rework from the current phase of work, while delaying somewhat the start of the next phase. Analyses of software engineering roll-off and software coding roll-off indicated that these changes would reduce program staffing by approximately 20 percent. More importantly, the delayed roll-off of software coding would reduce the time required for downstream work phases and total project duration.

Results

The Peace Shield project finished in month 47, six months and 13 days ahead of schedule. It was viewed by all as highly successful. To quote Ms Darleen Druyun, Acting Assistant Secretary of the Air Force for Acquisition, "In my 26 years in acquisitions, this is the most successful program I've every been involved with, and the leadership of the U.S. Air Force agrees." (Kausal 1996).

Post-project benchmarking and policy assessment

The on-budget, ahead-of-schedule, highly complimented Peace Shield program stood in stark contrast to a past effort to develop a different command and control system in the same organization. The latter program exceeded its original cost and schedule plans by several times, and suffered a large contract dispute with the customer. Theories abounded as to what had produced such significantly improved performance on Peace Shield. Naturally, they were “different” systems, different customers, different program managers, different technologies and different contract terms. These and more all were cited as (partially correct) explanations of why such different performance was achieved. Hughes executives were not satisfied that all the lessons to be learned had been.

System dynamics models of both programs had been developed. The two models were identical in structure (that is, the causal factors used in the models), but with different initial conditions and external impacts. Overlaying the simulations from these two programs such that they start at the same time indicates the substantial difference in their aggregate performance (Figure 9). Despite this difference (and many more detailed differences) in performance, the two programs were accurately simulated by an identical model structure (Figures 10 and 11 show model fit to aggregate staffing).

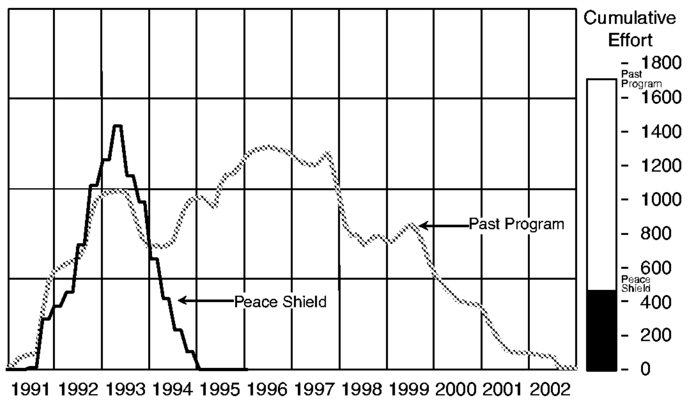


Fig. 9. Past program performance compared to Peace Shield

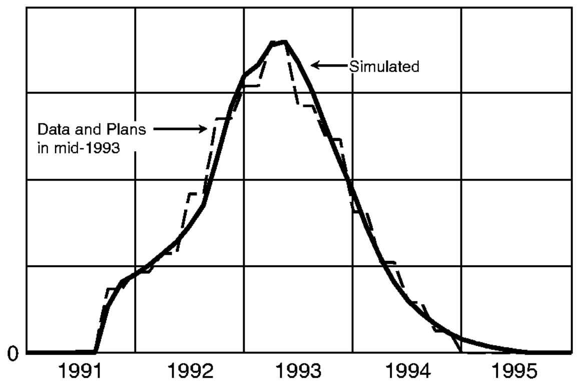


Fig. 10. Simulated Peace Shield: performance plotted against data

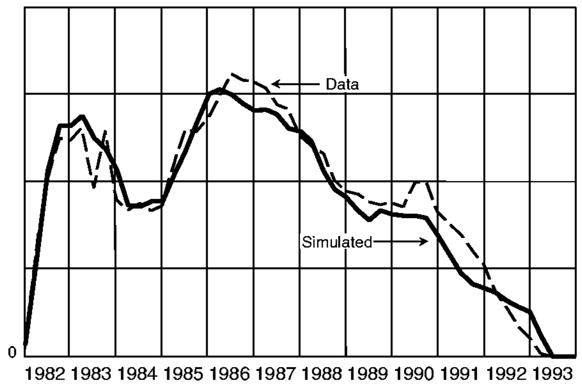


Fig. 11. Simulated past program: performance plotted against data

After achieving the two accurate simulations, the next series of analyses stripped away the differences in factors, one set at a time, in order to quantify the magnitude of performance differences caused by different conditions. Working with Hughes managers, the first step was to isolate the "external" differences—those in work scope, suppliers, labor markets. The removal of those different conditions from the past program model yielded the intermediate simulation shown in Figure 12.

After removal from the troubled program simulation of the differences in scope and external conditions, the simulation in Figure 12 represents how Peace Shield would have performed but for the changes in managerial practices and processes. While a large amount of performance difference clearly was attributable to external conditions, there is still a *halving* of cost and time achieved on Peace Shield remaining to be explained by managerial differences. We then systematically altered the remaining factor differences in the model that represented managerial changes discussed above. The incremental impact

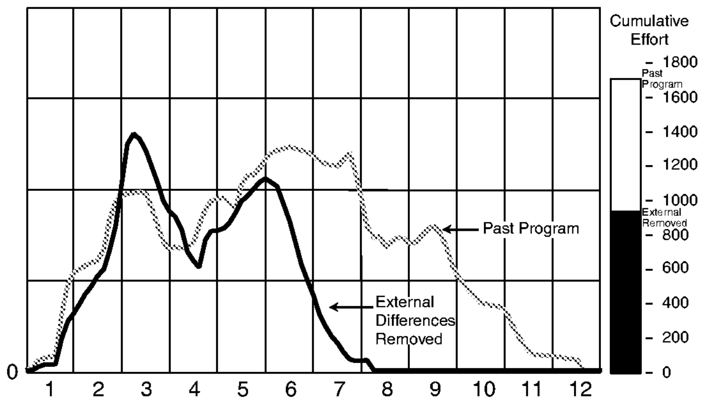


Fig. 12. Past program with external differences removed indicates how Peace Shield would have performed absent management policy changes

of these changes from the simulation in Figure 12, in the following order, is shown in Figure 15:

1. More aggressive Peace Shield schedules (these would have increased the costs of the program).
2. Different staffing strategy for roll-off of software design and coding, and start of next phases (as illustrated in Figure 13, this policy delayed the start of software coding until software design was 30 percent complete, versus 10 percent for the past program).
3. Use of more experienced engineers and supervisors.
4. Teaming and other process improvements, which generated significantly reduced rework discovery times as illustrated in Figure 14.

When these were made on top of removing the external differences, the model was transformed from that of a highly troubled program to that of a very successful one—and the performance improvements attributable to each aspect of the managerial changes were identified and quantified.

A summarized version of the results is given in Figure 15 and 16. Approximately 44 percent of the improvement in Peace Shield comes from better external conditions and 56 percent from the three categories of better management policies and processes discussed above (note that the first change, schedule acceleration, is not really a better management policy or process).[1] Figure 16 shows that enormous savings were and can be achieved by the implementation of what are essentially "free" changes—if only they are known and understood. That was the groundbreaking value of the preceding analysis: to clarify just how much improvement could be achieved by each of several policies and practices implemented on a new program. What remained to be achieved was to systematize the analytical and learning capability in a manner that would support new and ongoing programs, and help them achieve continuing performance gains through a corporate "learning system" that

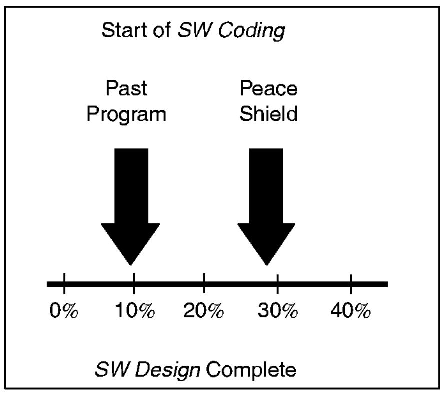


Fig. 13. More disciplined staffing pushed the start of downstream work (software coding) until the upstream phase (software design) was further along

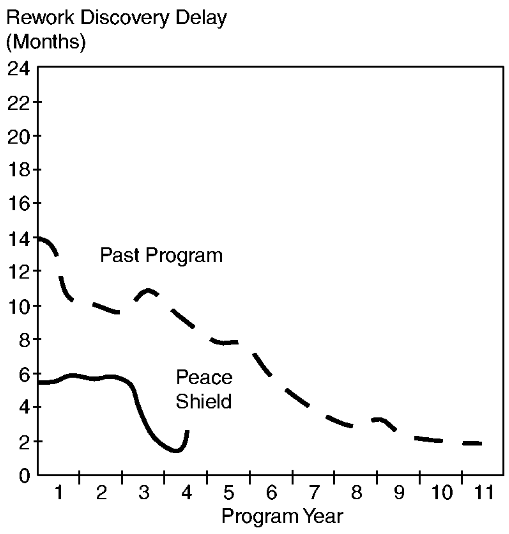


Fig. 14. Teaming and other process improvements reduced rework discovery time

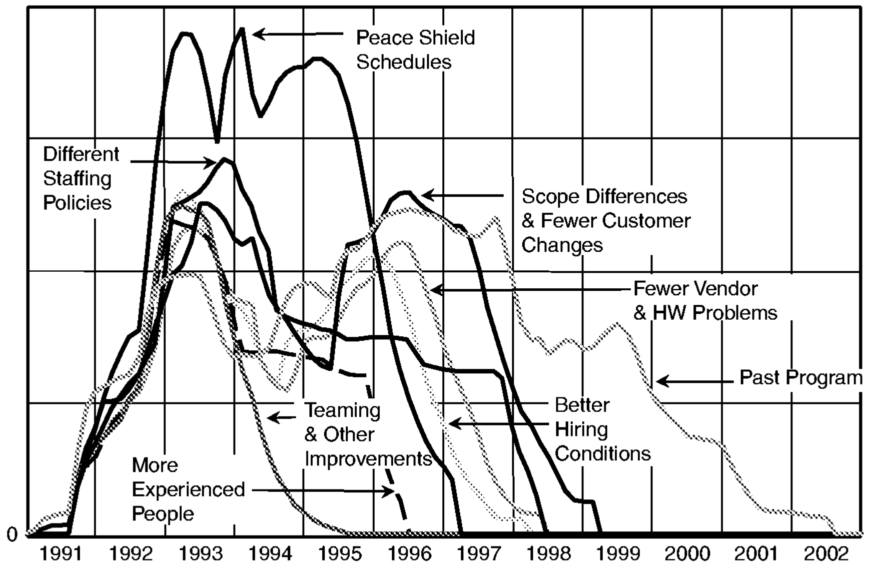


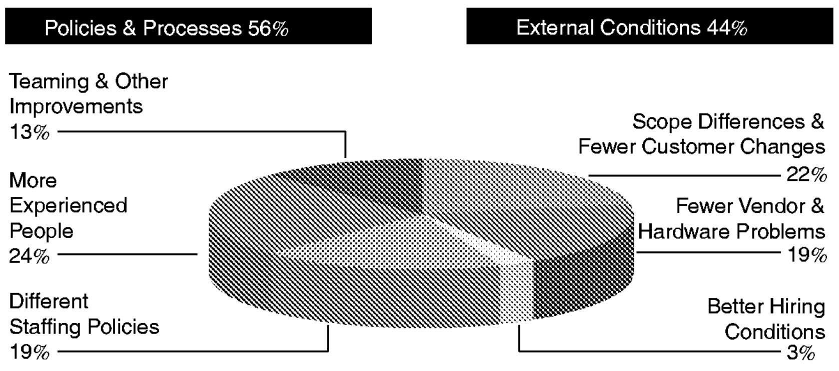
Fig. 15. Improvement in program performance from management policy changes

would yield effective management lesson improvement and transfer across programs.

Systemization of the learning from these analyses was accomplished via:

- Development of a system around the program model. The system included the basic model, a database of key parameters from several programs, a

Fig. 16. Where did the cost improvement come from?



series of questions that guided the user into setting up the model for a new program, software for "calibrating" the new program model to data and/or plan, scenario test capability, and simulation analysis capability.

- An intranet-based document describing best practices for managing programs, including key managers to consult for advice.
- Use of the system on two additional programs. On one of these programs, use of the model in response to customer requested changes resulted in documented savings of approximately $10 M, plus a schedule saving of a few months, and a very satisfied customer.

Chuck Sutherland, Peace Shield Program Manager, noted in the best-practices document:

> One of the tools that should be used on every program is a simulation model. We have used a model on several programs and it allows you to understand the full impact of changes. For example, a model will help determine what will happen if you add manpower, or do something else. Also, your model can help determine whether the cost and schedule you've predicted is consistent with the way your processes are performing on the program. It is a tool that should be implemented on all of our programs. It should be made available to all program managers so they can predict better and see the impact of risks and problems.

Conclusions

The strategic management of complex development projects, including designing project schedules and resources, determining measurement and reward systems, evaluating risks, and learning from past projects, is greatly facilitated by the use of system dynamics models. Complex development projects are highly non-linear feedback systems and have proven extremely difficult to manage successfully using traditional tools alone. System dynamics models have demonstrated their ability to improve significantly the quality and performance of management on complex projects.

The use of system dynamics is most effective when it is part of an ongoing learning system: prior models and practices provide a basis for designing and bidding future projects, for assessing risks, and for introducing new management methods. Insights and guidelines learned improve the intuition of managers and make development of future managers more effective.

Acknowledgements

In addition to the authors, our colleagues Craig Stephens, Rick Park, and Donna Mayo contributed to the Peace Shield effort.

Note

1. In any complex dynamic system, the impacts of a series of changes on system performance are not usually independent and therefore the total change is generally not the sum of the impacts of the individual changes computed separately. For example, the first change to occur on a project may not cause significant problems as it may be handled easily with existing buffers or with a small amount of overtime; if a few additional resources are required, the skill dilution may not be that significant. However, if more changes are added, the impact builds—buffers may become exhausted, overtime periods extended, skill dilution more significant. As secondary and tertiary feedback effects amplify the problem, additional changes have an even more significant impact. As a result, the impact of change generally builds nonlinearly with the cumulative magnitude and duration of the changes. This non-linearity complicates the attribution of impact to any specific change because the order of removal or addition of changes in a simulation affects the computed incremental impact of the change. This problem is often encountered in determining the costs of changes imposed by "owners" on contractors and builders in contract disputes. In these cases, however, the solution is also straightforward—the changes are removed in reverse chronological order, working backwards from what did happen to what would have happened absent the changes. In the Peace Shield, analysis, however, where we are comparing two completely different projects, chronological order is not relevant. In this analysis, we removed the external factors first, thereby potentially attributing to them a greater than warranted impact (and vice versa for the management factors). The beneficial impact of the management factors would increase if they were applied to the troubled program (i.e., with significant external impacts).

References

Abdel-Hamid T, Madnick SE. 1991. *Software Project Dynamics: An Integrated Approach*. Prentice Hall: Englewood Cliffs, NJ.

Cooper KG. 1980. Naval ship production: a claim settled and a framework built. *Interfaces* **10**(6): 20–36.

Cooper KG, Mullen TW. 1993. Swords & plowshares: the rework cycles of defense and commercial software development projects. *American Programmer* **6**(5): 41–51.

Cooper KG. 1993a. The rework cycle: why are project mismanaged. *PM Network Magazine* February 1993; 5–7.

Cooper KG. 1993b. The rework cycle: how It really works ... and reworks .... *PM Network Magazine* February 1993; 25–28.

Cooper KG. 1993c. The rework cycle: benchmarks for the project manager. *Project Management Journal* **24**(1): 17–21.

Ford DN, Sterman JD. 1998. Dynamic modeling of product development processes. *System Dynamics Review* **14**(1): 31–68.

Homer J, Sterman J, Greenwood B, Perkola M. 1993. Delivery time reduction in pulp and paper mill construction projects: a dynamic analysis of alternatives, *International System Dynamics Conference*.

Kausal BAIV. 1996. *Program Manager*. March–April: 22–24.

Morris P, Hough G. 1987. *The Anatomy of Major Projects*. Wiley: New York, Chichester.

Reichelt KS, Lyneis JM. 1999. The dynamics of project performance: benchmarking the drivers of cost and schedule overrun. *European Management Journal* **17**(2): 000–000.

Richardson GP, Pugh AL. 1981. *Introduction to System Dynamics Modeling with DYNAMO*. Productivity Press: Cambridge, MA.

Roberts EB. 1992. Strategic Management of Technology: Global Benchmarking, 10 December [Results of a survey sponsored by the Massachusetts Institute of Technology, Cambridge, Mass and PA Consulting Group, London, England].

Rodrigues AG, Williams TM. 1998. System dynamics in project management: assessing the impacts of client behaviour on project performance. *Journal of the Operational Research Society* **49**(1).

Shtub A, Bard JF, Globerson S. 1994. *Project Management: Engineering, Technology, and Implementation*. Prentice-Hall: Englewood Cliffs, NJ.

Sterman JD, Repenning NP, Kofman F. 1997. Unanticipated side effects of successful quality programs: exploring a paradox of organizational improvement. *Management Science* **43**(4).

Sterman JD. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin McGraw-Hill: New York.

The World Bank Operations Evaluation Department (OED). 1992. *Evaluation Results*. The World Bank: Washington, DC.