PEGASUS
COMMUNICATIONS

THE SYSTEMS THINKER™

BUILDING SHARED UNDERSTANDING

VOLUME 11
NUMBER 1
FEBRUARY
2000

Celebrating 10 Years!

LARGE-SCALE PROJECTS AS COMPLEX SYSTEMS: MANAGING "SCOPE CREEP"

BY ANDREA SHAPIRO AND CAROL LORENZ

C. Northcote Parkinson's now famous adage, "Work expands so as to fill the time available for its completion," may be overly optimistic. Unfortunately, work tends to expand *far beyond* both the time and the money budgeted for its completion, particularly for complex projects. For example, estimates published last year in *Communications of the ACM* (Association for Computing Machinery) indicate that U.S. companies spent approximately $59 billion in 1995 on cost overruns and an additional $80 billion on canceled information technology (IT) projects alone. These estimates do not include overruns in government or public-sector projects. A similar Coopers & Lybrand study in the United Kingdom indicates that 85 percent of IT projects are over budget, miss their schedule, or fail to meet customer expectations.

Although IT and software development projects may be the most visible areas in which work extends beyond its original parameters, process reengineering efforts, wide-ranging organizational change initiatives, and large-scale construction projects certainly are not exempt. Take, for example, a small town in North Carolina that is building a reservoir for its municipal water supply. The original estimate for the project five years ago was $5.4 million, with a two-year window for construction. Today, the estimated cost is $8 million. Construction has barely begun and is now projected to take three years.

Given these troubling statistics, it seems fair to say that large projects rarely, if ever, stay within their original specifications and meet their forecast targets of time and dollars. In fact, any program or project that is complex in nature, covers an extended period of time, requires a significant monetary investment, and has multiple components needing to be managed simultaneously is vulnerable, despite heroic traditional project management efforts. This tendency of such projects to expand beyond their initial boundaries and thereby to extend far beyond their forecasts is often called "scope creep." In this article, we examine this phenomenon by viewing large projects as complex systems. By understanding scope creep in this context, we can begin to identify how and why it occurs. Equipped with this knowledge, empowered people, teams, and organizations can more effectively plan for and mitigate the effects of scope creep by taking a more realistic and dynamic perspective on the project management process.

Scope Creep Dynamics

How does scope creep begin, and what causes it to build on itself over time, leading to costly overruns and delays? Organizations initiate projects to address a perceived need (see "The Dynamics of Scope Creep" on p. 3). For instance, the town mentioned above faced periodic water shortages during times of drought, so it decided to build a reservoir to meet its perceived water needs. In an ideal world, after some delay, the project would be completed within its original budget and on schedule, and the perceived need would be filled (B1).

The amount of money invested in a project limits the amount of work that can be accomplished and therefore generally defines the project's scope—at least initially. The larger the project, and the more people, departments, and agencies involved, the more complex the governance becomes. For instance, in an IT company, a large-scale product development effort requires input from a number of departments, each with its own management and its own priorities. Several subprojects of the larger effort could end up competing with

Continued on next page ➢

➢ Continued from previous page

each other for the skills of testers, database programmers, or technical writers, making the overall governance complex. The competition for limited resources intensifies as the number of subprojects expands, making management of the overall effort more challenging.

When the governance is complex, managers find it difficult to make decisions in a timely manner and to attend to the long-term consequences of each decision. These difficulties threaten the accuracy of schedule estimates. As deadlines draw closer and time pressure rises, people tend to "cut corners" to keep the project on track. In a software development project, for example, cutting corners often means inappropriate parallel development—perhaps developing applications that depend on lower-level modules *before* the lower-level modules are fully designed. Cutting corners in a construction project could mean accepting the lowest bid without qualifying the vendor, not allowing enough time to coordinate the work of the various subcontractors, or even neglecting to let cement cure properly. In the short term, cutting corners can appear to alleviate schedule pressures (B2), but it is actually an example of the classic "Fix That Fails" archetypal structure. In a "Fixes That Fail" scenario, an action that solves a problem over the short term actually exacerbates the same condition over the long run. Thus, over time, cutting corners generally results in rework, which inevitably creates even more schedule pressure (R3).

THE SYSTEMS THINKER™

Managing Editor: Janice Molloy
janicem@pegasuscom.com
Founding Publisher: Daniel H. Kim
Publisher: Ginny Wiley
Editors: Laurie Johnson, Kellie Wardman O'Reilly
Contributing Editor: David W. Packer
Production: Julia Kilcoyne
Circulation: Julie McCay Turner
juliet@pegasuscom.com

THE SYSTEMS THINKER™ explores both the theory and practice of systems thinking and related organizational development disciplines. Articles by leading thinkers and practitioners articulate the challenges and issues involved in creating organizations on the leading edge of innovation. We encourage dialogue about systemic issues and strive to provide a forum for debating such issues. Unsolicited articles, stories, and letters to the editor are welcome.

THE SYSTEMS THINKER™ (ISSN 1050-2726) is published ten times a year by Pegasus Communications, Inc. Signed articles represent the opinions of the authors and not necessarily those of the editors. The list price is $189.00 for one year. Back issues and volume discounts are also available.

Editorial and Business Address:
Pegasus Communications, Inc.
One Moody Street, Waltham, MA 02453-5339
Phone (781) 398-9700 • Fax (781) 894-7175

www.pegasuscom.com

Rework also increases cost. As the cost of a project goes up, project managers need to justify the additional expenditures to upper management or to their constituents. To do so, they often find themselves promising new and enhanced features to make the additional expense more palatable. For instance, an elected official might tell an irate group of taxpayers, "Of course a dam costing this much will include a recreation area." Or the constituents might decide that such a costly undertaking must include a recreation area and pressure officials to add one. The expectations of an IT project might rise when the sales department assures dissatisfied customers, "Oh, we have a new release in the works that will address your concern." Assurances and assumptions such as these raise customers' expectations and expand the project's scope (R4).

As the project's scope increases, the web of interactions and dependencies among tasks, subprojects, departments, and agencies grows even more intricate. This rising complexity can lead to delays. For example, on a construction project, if one subcontractor has to wait for a second one to finish his tasks, he may need to take on another job in the interim. He then may not have the personnel immediately available to work on the original project when the first subcontractor is finally ready for him. Delay creates even more deadline pressure and reinforces the "scope creep" dynamic.

Such interactions and dependencies also make it difficult to accurately estimate costs. Delays can further compound the problem, making cost estimates even less accurate. If actual costs are much higher than the original estimates, the pressure on officials and managers to justify costs increases, which can further influence the expectations from the project and cause even more scope creep (R5).

The complexity of the governance affects more than the timeliness of decisions. It also affects the ability of decision-makers to see the effects of their decisions—both on other aspects of the project and later in time. With less visibility to downstream effects, decisions that appear to give positive local results can have negative global consequences. Therefore, the quality of decisions can go down—increasing the likelihood of additional rework and creating additional complexity (R6).

Thus, the tendency of large projects to increase in scope finds its roots in two underlying assumptions. The first is that we can manage a project by simply managing its parts. This practice leads us to ignore the systemwide impact of apparently small, local decisions, which in turn can undermine initial time and cost estimates and increase the project's complexity. The second is that we assume that we can estimate schedules and costs accurately in advance of initiating work. But often, when a project costs more or takes longer than expected, we respond either by assuming that it must include additional features or by deliberately adding more to justify the increased cost or time—compounding scope creep.

Beyond Traditional Project Management

To date, most explanations of scope creep stem from a linear, sequential view of projects and how they are managed. For example, the Project Management Institute defines the five project management phases as initiating, planning, executing, controlling, and closing (www.pmi.org/standards/pmbok.htm). This model assumes that one phase is completed before the next one begins, and that we should return to previous phases only when problems occur.

Using this sequential framework, several authors have written that scope creep occurs when the planning phase is incomplete. For example, some suggest that if objectives and project "deliverables" are not fully defined up front or if work breakdowns are unclear, then the project will exceed its original cost and schedule projections. Others attribute scope creep to ill-defined resource requirements or insufficient funding. Another set of explanations focuses on the problems that occur during the controlling phase; for instance, poorly documented changes to the project specifications.

All of these explanations are probably correct to some degree. However, even if a team rigidly adheres to this project management framework and completes each step with near perfection, scope creep can still occur, especially in large, complex projects. Good, linear project management is necessary. However, as the frequency of scope creep shows, it is insufficient to prevent the problem.

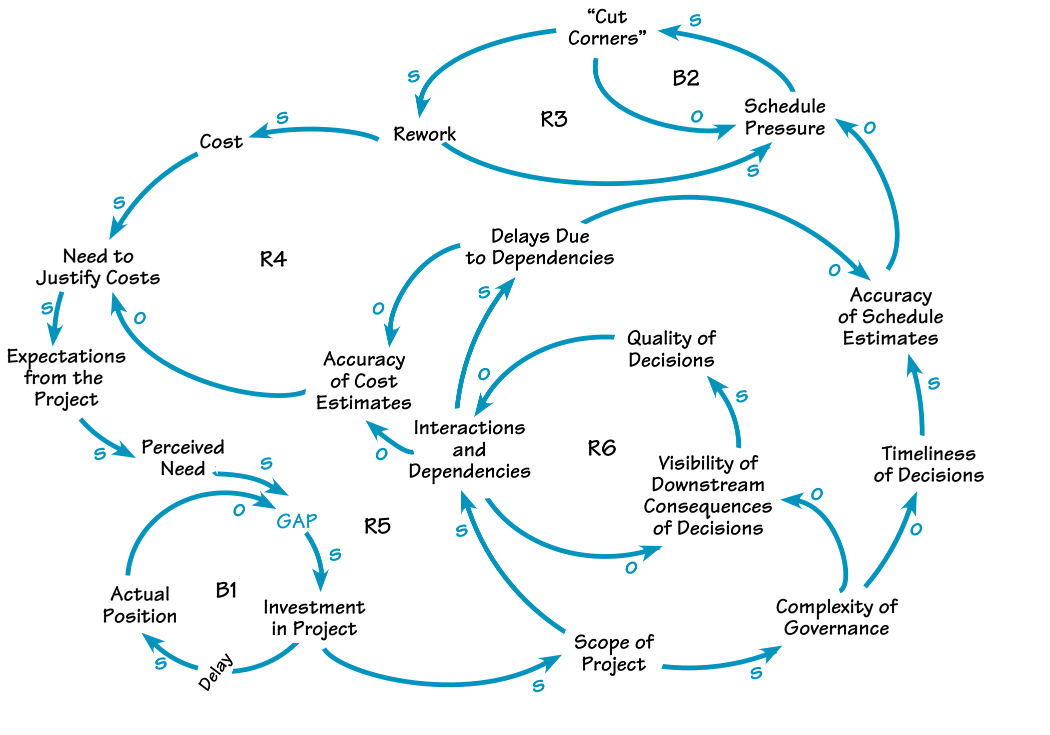
The downside of the traditional project management model is that it does not adequately account for the fact that a major project—like constructing a reservoir, developing a large, complex piece of software, or writing a major training curriculum—is actually a large system with multiple interacting parts. According to M.I.T. professor John Sterman, "Large-scale projects belong to the class of complex dynamic systems. Such systems (1) are highly complex, consisting of multiple interdependent components, (2) are highly dynamic, (3) involve multiple feedback processes, (4) involve non-linear relationships, and (5) involve both 'hard' and 'soft' data" ("System Dynamics Modeling for Project Management," section 3, p. 5, http://web.mit.edu/jsterman/www). In describing problems in these kinds of systems, author Meg Wheatley says, "They cannot be understood sufficiently either before or even while they are occurring; therefore, prediction and control are impossible" (*THE SYSTEMS THINKER,* V9N10). In other words, the interactions among the tasks, subprojects, and governance structures of a complex project make it all but impossible to fully define the project deliverables or grasp the resource requirements ahead of time.

For example, in software development, managers commonly divide projects into several modules—each of which is to be designed and developed independently and then combined into a completed product. They then break the work on each module into a series of tasks, which can be sequenced and placed on a timeline. It seems logical to think that if these tasks are accomplished in the prescribed way, their completion should constitute the total project. In practice, though, the degree of interdependency among the modules usually doesn't become fully evident until *after* the teams have invested much effort in design and development. At that point, the components often require more design, coding, and testing than originally budgeted to make the modules work together.

Equally important, as the interdependencies among the different modules come to light, the scope of the project changes—even if the project

Continued on next page ➢

THE DYNAMICS OF SCOPE CREEP



A large, complex project involves more than a series of tasks to be accomplished: It involves people in an organizational structure doing work, which includes decision-making, governance, and time and cost forecasts. Each of these aspects of the project has an impact on the others, adding dynamic complexity to the process and affecting people's expectations.

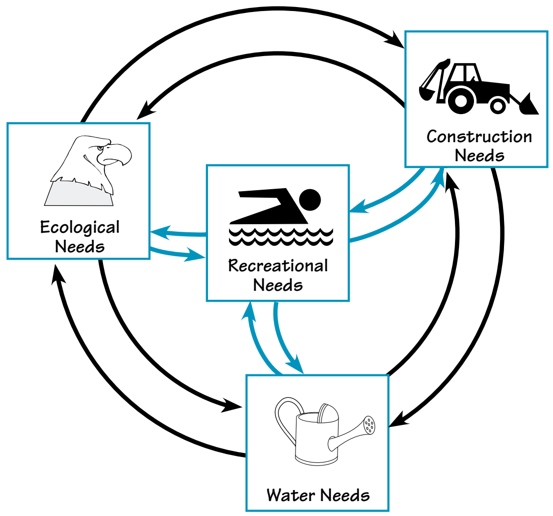
➢ Continued from previous page

plan does not—because the people involved get new ideas about what the product should offer. For example, a software project that began as a rewrite of code to fix a problem evolves when designers realize that addressing the problem makes it easier to implement a customer request that was difficult to implement in the old code. To satisfy the customer, they design the request into the rewrite. A single addition like this one typically does not pose a problem. However, when a rewrite is large and costly, managers may be pressured to add several of these customer requests to justify the project's expense. The standard approach to project management neglects to take into account such shifts in expectations. So, what began as a simple fix can unexpectedly become a major project to address customer demands.

Sterman identifies two kinds of complexity in projects: combinatorial and dynamic. *Combinatorial complexity* is created by the parallel and sequential activities that take place in a large, complex project—for example, in a large construction project, all of the permits needed before breaking ground. Traditional project management tools, such as PERT, CPM, and GANTT charts, are intended to help people handle these details.

On the other hand, *dynamic complexity* is created by the multiple feedback processes, time delays, and nonlinear causal relationships that exist in any large project. Because the interactions among multiple variables over time drive this kind of complexity, it can grow geometrically or even exponentially as additional elements enter the system (see "Increasing Dynamic Complexity"). For instance, the traditional view holds that adding a fourth task to three that are already scheduled would increase the workload proportionally—that is by 33 percent. However, this calculation does not take into account the fact that the new task interacts with each of the existing tasks, actually *doubling* the interactions between tasks and potentially increasing the amount of work by much more than one-third. For this reason, dynamic complexity contributes significantly to the phenomenon of scope creep and cannot be adequately addressed using standard, linear approaches.

INCREASING DYNAMIC COMPLEXITY



If a town chooses to add a recreational area to a dam project, it might seem logical to think that the scope of what has to be managed increases by 33 percent. Actually, by adding just one more component, the town has doubled the number of interactions among the different components—going from six to 12.

Mitigating Scope Creep

Clearly, we need to broaden our thinking about project management. Fortunately, there are several ways to do this:

Surface Dynamic Complexity. The first key to mitigating scope creep is to find ways to surface the feedback processes, time delays, and nonlinear relationships in a large, complex project. Here are a few suggestions:

- John Sterman has demonstrated the value of using system dynamics modeling in large-scale projects in conjunction with traditional tools. This approach can help to surface many of the interactions between tasks, subprojects, and governance structures that ultimately lead to scope creep and can make explicit many assumptions that often remain unspoken. With such information, decision-makers and project managers can then make more informed decisions about changing external conditions, proposed shifts in strategy, resource allocation, impacts of delays, and even the viability of the project over time.
- If using a system dynamics model isn't practical, drawing causal loop diagrams can help managers identify feedback loops and causal relationships separated by time and space. Discussion of the interactions and dependencies among various work streams can create a richer understanding of the complexities involved.
- With a basic knowledge of systems thinking and causal loop diagrams, decision-makers often recognize many "shortcuts" as "Fixes That Fail." But because it is difficult to spot patterns of behavior while implementing a project, managers should anticipate and document potential problem areas up front. One way to do so is to identify any systems archetypes operating in earlier projects, document them, and share them with the project team. The team then needs to continually evaluate whether similar patterns are emerging in the present project, and take appropriate actions to keep from falling into the same traps. Anticipating these situations, documenting them in advance, and appointing someone to flag them if they crop up can lead to smarter decisions once the action starts on a major initiative.
- The decision-making process in large projects is often extremely complex, cumbersome, and subject to pressures from the outside. These pressures can include political considerations, the organization's culture, ingrained management practices, and budgetary constraints. Because of the

complexity of the decision-making process, project managers often make decisions without regard for how they might affect the rest of the system. One way to simplify decision-making and to help identify unintended consequences is to use a decision matrix (see ”Sample Decision Matrix” at www.pegasuscom.com/matrix.html). To do so, make a list of potential outcomes across the top of a page. Then brainstorm the possible solutions under consideration and list them down the side of the page. Discuss the different resulting combinations represented by each "cell" in the matrix before making any decisions. Use a ranking process to determine which cell or cells best meet the desired outcomes.

- Scenario planning offers another way to identify and explore the possible outcomes—both predictable and unlikely—of decisions being considered. For a simple application of scenario planning, see Peter Senge et al., *The Dance of Change: The Challenges to Sustaining Momentum in Learning Organizations* (Doubleday, 1999), pp. 187–190. During the scenario-planning process, project managers and teams might also identify some additional "Fixes That Fail" to avoid.

Rethink Approaches to Budgets and Schedules. We also need to rethink the common assumption that we can make accurate cost and schedule predictions up front in a large project. This expectation is unrealistic, because of the inherently changeable nature of such projects. Therefore, we need a "scalable" approach to budgets and schedules, one in which we regularly review and update time and money estimates. Such reviews should include a timetable for decisions regarding moving the project forward, scaling it back, or perhaps even canceling it. Over time, as we learn more about the work involved, the estimates should become more accurate.

No organization can launch a project without having some idea about the amount of money and time involved. Managers should thus provide a *range* of initial cost and schedule estimates, along with a margin of error and a review plan with "escape points" as described above. Such estimates and built-in reviews help prevent people from feeling trapped and betrayed when large projects go "over budget" and get delayed.

Use the Original Purpose as an Anchor. There is a tendency to use current expectations as a kind of anchor point when considering changes to the scope of a project. This propensity can be dangerous, because current expectations may have shifted significantly since the project's inception. In fact, the dynamics behind scope creep are similar to those of the "Drifting Goals" archetypal structure. In "Drifting Goals," two balancing loops interact to cause goals and standards to decline over time. In scope creep, a series of reinforcing processes constantly "raise the bar" and prevent the project from being finished within its original projections. To help avoid this process, go back to the *original* purpose of the project and use it as a reference point. Why was the project started in the first place? If you do decide to change its scope, what might be the ramifications?

Follow Good Project Management Practices. Finally, continue to use good project management practices. Many traditional tools remain essential to tracking the progress of any project. However, remember that in a complex system like a large project, the whole is not equal to the sum of its parts; the whole includes the sum of its parts *as well as* the interactions among all of the components.

Beyond the Old Paradigm

Albert Einstein is often quoted as saying, "The significant problems we face cannot be solved at the same level of thinking we were at when we created them." Scope creep is such a problem. To date, most attempts to control scope creep have used linear, additive logic to counteract what is a highly complex, dynamic phenomenon. This sequential way of thinking places the blame for scope creep on poor project management practices. But rather than rushing out to try new project management technologies or blaming people for failing to implement older ones, what we really need is a new way to understand large projects as complex systems. In this article, we have offered approaches that are not visible from the traditional linear approach. We believe that these ideas can serve as learning opportunities and as seeds for yet other solutions. Learning to see large projects as complex systems may not be a revolution, but we think it is a worthy first step toward a possible and necessary evolution in the way organizations operate. ▪

Andrea Shapiro, Ph.D., is an internal organizational learning consultant for Nortel Networks. Her work emphasizes learning, communication, and systems thinking (ashapiro@nortelnetworks.com). Carol Lorenz, Ph.D., of Carol Lorenz and Associates, is an independent contractor who worked at Nortel for a number of years. Her work emphasizes learning, organization effectiveness, and systems thinking (lorenzc@mindspring.com).

NEXT STEPS

- Circulate the article among the members of your project team and discuss how the concepts may apply to your work. Ask, "What ideas in the article most caught your interest?" "Were there any parts of the article that rang especially true for us?" "How can thinking about our project as a complex system help us to make better long-term decisions?"
- Make a list of shortcuts that you might take (or have taken) on your project in order to keep it on schedule. What might be some long-term, undesirable effects of these shortcuts?
- Create a causal loop diagram capturing the relationships among the multiple aspects of your project. Use this exercise to surface and challenge assumptions about how the elements of your project affect each other.
- Document the purpose of the project and what it is expected to accomplish, and ask whether expectations have expanded during regular reviews.
- Appoint a "historian" to mine data from earlier projects, identifying patterns of behavior or archetypes that led to problems. Educate the project team on these problem areas so that everyone can watch for similar patterns in the current project.