PROJECT MANAGEMENT

SOFTWARE DEVELOPMENT MANAGEMENT

SOFTWARE PROJECT MANAGEMENT

This article provides an overview of project management principles and practices, with particular emphasis on software projects. Software has become an integral part of products ranging from automobiles to microwave ovens. For electrical and electronics engineers, software development is seldom an end in itself. Instead, software projects are often embedded in the context of a broader product development process. Yet, in many cases, problems in the area of software development can produce detrimental effects that ripple through the entire product development process, often delaying the process, causing it to go over budget, or resulting in a product that is of low quality. For these reasons, it is important to have some background and appreciation for project management in general, as well as the particular pitfalls often associated with software projects.

Nowhere is the effect of poorly managed software projects more evident than in the context of medical devices. In this and other contexts in which software is embedded, software defects can lead to products that not only fail to perform as intended, but also are downright lethal. In the mid-1980s, for example, a software bug caused a medical device to deliver huge overdoses of radiation to six cancer patients, ultimately killing three of them. Since 1986, more than 450 reports have been filed with the U.S. Food and Drug Administration concerning software defects in medical devices (1). During the first six months of 1997 alone, the FDA issued 20 product recalls because of software problems that made the devices unsafe. The problem of poor software quality extends, of course, beyond medical devices. Anthes (1) provides several examples of other product categories in which software defects were responsible for triggering accidents:

- Transmission software was implicated in a 1991 accident in which a bus plunged off a California mountain, killing seven Girl Scouts.
- During the Gulf War in 1991, a software bug in the targeting software of a Patriot defense missile allowed an Iraqi Scud missile to hit the barracks of American servicemen, killing 29 Americans.
- In 1996, General Motors recalled nearly 300,000 automobiles because of a software problem that could cause an engine fire.

In addition to the safety-related issues associated with software products that are improperly developed or tested, software projects themselves frequently fail and are notorious for going significantly over budget and falling behind schedule. In a survey conducted by the Standish Group, a Massachusetts-based consulting organization, it was estimated that in 2004 companies in the United States alone spent $38 billion on canceled software projects and an additional $17 billion in cost overruns for software projects that were eventually completed (2). In some cases, mismanagement of software projects means that they escalate out of control, continuing to absorb valuable resources without ever delivering benefits to the organizations that undertake them (3). Consider the following example:

- Between 2000 and 2006, the UK Child Support Agency (CSA), poured over £456M into a new system designed to replace an inadequate system that no longer met the needs of the agency. Repeatedly delayed (4, 5), "[t]he project . . . went down the pan, despite an incredible array of reviews that warned it was headed for a crash. As well as the executive programme board and an EDS guide, there were no less than 40 internal audits. Then there were the Gateway Reviews, which were introduced as a means of preventing programmes of IT and organisational change going awry; but are conducted in secret and their results protected from scrutiny. After taking all this advice, the CSA wasted £91m on external advice." (6) Two thirds of the money collected by the CSA is wasted on administration because of the computer system. There's a backlog of 333,000 cases representing 25% of all claims received and the typical case takes 34 weeks to clear. (6).

Unfortunately, the UK CSA case is not an isolated example. A recent survey sponsored by the Information Systems Audit and Control Association (ISACA) found that 30–40% of all software projects undergo some degree of project escalation (7).

PROJECTS AND PROJECT MANAGEMENT: THE CHALLENGE

Having illustrated some of the consequences that can result from improper management of software projects, we turn now to defining what is meant by the terms project and project management and why software projects in particular may be especially challenging to manage.

Defining the Terms: Project and Project Management

A project can be defined as an interrelated set of activities designed to accomplish certain desired objectives within a limited period of time. Understanding this definition is critical to successful project management. Central to the definition of a project is the notion of achieving certain desired objectives. Failure to pin down these objectives at the outset of the project is probably the number one cause of project failure. Another key aspect of the definition is the notion that projects should not go on forever (i.e., they should have a defined end point). Project management is a set of concepts, principles, and techniques by which projects can be defined, planned, and controlled in order to meet project objectives.

What Makes Projects Challenging

Projects are challenging for several reasons. First, they are of limited duration and often involve highly constrained timelines as well as constrained resource availability. Second, successful project execution often requires a complex sequencing of many different activities. Third, there is often some degree of uncertainty regarding the work required, the methods by which the work will be accomplished, and hence, the costs and scheduling of activities. Finally, the project manager must often lead a diverse group of individuals from different disciplines and functional areas, often without any formal permanent authority over the individuals on the team.

What Makes Software Projects Particularly Challenging

If projects are challenging by their very nature, it can be argued that software projects, which are the focus of this article, are particularly challenging. It is well known, for example, that software projects are notoriously difficult to control (8–10). There are at least three factors that explain why this is so. First, software is abstract and intangible. It cannot be seen or touched in the same way that one can see and touch a physical object. Second, software requirements are seldom known with great certainty at the outset of a project. Instead, the requirements often evolve and change throughout the development process. Third, though we sometimes speak of software engineering, the reality is that software development is still very much of an art as opposed to a mature engineering discipline.

The intangible, or invisible, nature of software has serious implications for software project management (11). First, it is difficult to manage something that one cannot see. This may be one reason why software projects are consistently undersized (12, 13). From the outset it seems, many software projects suffer from poor estimation. Second, the lack of visible milestones exacerbates the problem by making it difficult to obtain an accurate indication of project status along the way to completion. This difficulty is commonly referred to as the 90% syndrome. The 90% syndrome refers to the tendency for estimates of work completed on a software project to increase steadily until a plateau of 90% is reached. The problem, according to Brooks (8) is that software projects tend to be 90% complete for half of the total coding time. Abdel-Hamid (14) has conducted simulation studies suggesting that the 90% syndrome results from "the interaction of two factors: underestimation and imprecise measurement of project progress due to poor visibility."

Another defining characteristic of software projects is requirements volatility (10, 11). With software projects, we have grown accustomed to the idea that functional specifications will change and evolve during the course of development. In the context of construction projects, most of us would be shocked to learn of a user request that required undoing 50% of the existing structure, yet in the context of software engineering we have become desensitized to the implications of such requests. In fact, they seem normal. Almost certainly, projects that are subject to such volatility are more difficult to manage.

Finally, software development is still largely an art. In a comparison of software engineering with other engineering disciplines, Shaw (15) concludes that software engineering is a relatively new and immature discipline. Thus, software has traditionally been crafted by individuals who translate functional requirements into custom solutions using the tools and methods of an artisan rather than an engineer (16, 17).

A manufacturing analogy can be made to the era of craft production that preceded the industrial revolution. Two hundred years ago, all products were crafted by hand, starting with the most basic raw materials. A gunsmith, for example, would create all his own screws out of rod stock, carefully threading each individual screw uniquely to fit its location in the finished weapon (16). As one can imagine, this was a painfully slow and expensive way to make rifles. It was also hard to control the quality of the finished goods. In fact, no two rifles were exactly the same. It is precisely this paradigm applied to software development that leads to many of the problems mentioned earlier.

There are, however, signs that a new paradigm of software development based on so-called object technology has begun to emerge. The essence of object technology is the construction of new software out of standard, existing components, leading to fundamentally higher levels of productivity and quality. Some would argue that this new paradigm, and the impact it may hold for software development, is analogous in many ways to the changes in manufacturing practice that resulted in the industrial revolution.

In the long run, there is the prospect that software development will evolve along the same trajectory as other engineering disciplines, maturing from craft to professional engineering discipline (15). In the meantime, however, we are left with the challenge of managing software projects in an environment that is still very much in a state of flux.

Before discussing more about how to manage such projects, it would perhaps be useful to describe the alternatives that exist for acquiring software. This will provide some sense of the range of different types of software projects that one may encounter. While each approach produces a different kind of project, the basic concepts and techniques of project management that will be discussed later remain broadly applicable.

ALTERNATIVE MEANS OF ACQUIRING SOFTWARE

Software can be acquired in several different ways. Figure 1 illustrates the approaches that are commonly used. While the discussion that follows focuses on software that is acquired for organizational use, many of the same concepts are equally valid for software that is embedded in products.

The basic choice is whether to make or buy. This decision rests on a number of different factors including the availability of in-house software development resources, as well as the strategic value of the software to the organization and the degree to which it is expected to confer a competitive advantage in the marketplace. In general, if the software is not seen as strategic, there are often compelling

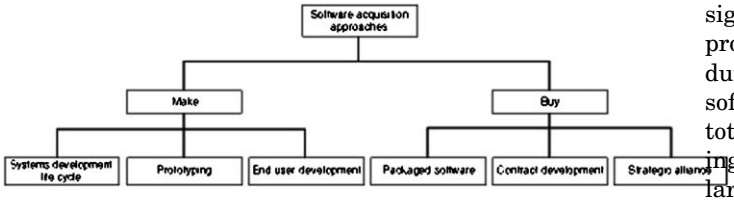

Figure 1. Alternatives for software acquisition. Managers can choose from a variety of approaches in acquiring software.

reasons to buy it rather than make it in-house.

Buying Software

If the application addresses a common problem that other organizations have experienced, there is a good likelihood that packaged software is available to meet the need. If this is the case, it is usually advisable to simply buy the software for reasons of both cost and quality. Before making a decision to buy packaged software, however, it is important to know the degree to which the software conforms to organizational needs. If conformance is low, the software may require modification which can be costly. The first decision, however, is whether to make or buy the software.

If packaged software is not readily available, one option is to contract with a software development firm. Another option is to enter into a strategic alliance with another firm that is interested in developing the same type of application. In any case, it is important to note that buying software still involves many of the same activities encountered in making software. Neither the requirements determination (i.e., definition) process nor the implementation (i.e., deployment) process go away when a decision is made to purchase software as opposed to making it in-house. Since these are processes where many software projects run into trouble (as opposed to the actual coding or development process), the risks associated with purchasing software should not be underestimated. Many of the same project management processes still apply.

Making Software

If a determination is made to make rather than buy the software, there are several choices available here as well. If the application is small, there is the option of end user development. For small- and medium-sized applications, prototyping is an increasingly popular approach and is especially useful when requirements cannot be well specified at the outset of the project. Prototyping is an iterative process that typically begins when a developer meets with a user to discuss the scope of the proposed system and the user's basic requirements. The developer then spends a few days putting together a prototype using tools that allow for rapid development of input/output screens and some system functionality. The developer then shows the prototype to the user and asks for feedback. This process continues until the user is satisfied with the system. In the ideal case, the prototype evolves in this manner and becomes a usable system. For large applications, a more formal approach known as the systems development life cycle (SDLC) lends additional structure to the project by breaking it down into a series of phases (e.g., analysis, design, coding, testing, and implementation). SDLC is an approach for managing software projects that was developed during the 1970s and is still used today for large, complex, software development projects. It should be noted that prototyping and SDLC are not mutually exclusive; prototyping often provides a means of fleshing out requirements on large, complex applications. A variety of other software development methodologies (many of which are proprietary) have been developed that incorporate elements of structure from SDLC with the notion of iteration from prototyping.

SDLC Phases

Many different authors have written about the SDLC and it seems that each author breaks the life cycle into a different number of phases and uses slightly different names to refer to each particular stage. For the purposes of this article, I will present a simplified mnemonic version of the SDLC with just 4 phases which I call the 4 D's: define, design, develop, and deploy. Most other versions of the SDLC can be easily mapped into the 4D framework.

Define. The define phase involves scoping out the requirements of the system. During this phase, a statement of project scope is developed and a preliminary feasibility assessment is performed, eventually leading to a go/no go decision on the project. Next, a more detailed requirements determination and analysis process is undertaken to document the required functionality of the system as well as the input screens and output reports that will be required. A variety of techniques are used to elicit such requirements from end users including interviews, group meetings (e.g., joint application design sessions), observation, and questionnaires.

Next, modeling techniques such as data flow diagrams, entity relationship diagrams, and object models exist to aid systems analysts with the complex task of representing system requirements. As noted earlier, building prototypes is also a good mechanism for fleshing out requirements. For small systems, an SDLC process may require more overhead than is necessary and the entire system can be developed using a prototyping approach.

Design. The design phase involves translating the requirements into a set of design documents that can be used to guide development. The design phase is often broken down into sub-phases that are referred to as logical and physical design. Logical design involves translating the user requirements into a conceptual design that includes a full functional description of the system and its data requirements, but is independent of the hardware on which the system will be built. Physical design involves the development of detailed specifications for programmers to follow and takes into account the physical hardware that will be used to implement the system.

Develop. The develop phase involves the actual coding, testing, and debugging of program modules. It also includes developing appropriate documentation for the system. The development phase also includes plans for conversion to the new system, writing user manuals, and provisions for

training users.

Deploy. The deploy phase involves implementing the new system in the organization. If the system is designed to replace an existing system, there are several so-called conversion strategies. These include running the two systems in parallel until the new system is shown to perform adequately, piloting the new system at selected sites with an eye toward learning things which can be applied to subsequent deployments, or phasing in particular modules of the system one at a time. These strategies are designed to reduce risk and are not mutually exclusive. A higher risk strategy is the so-called "cold-turkey" strategy which involves simultaneously turning off the old system and bringing up the new system. Aside from the technical problems of conversion, there are often more significant problems that arise from what has been labeled user resistance. For a variety of reasons, often political in nature, individuals will resist using new software systems. Thus, it is not uncommon for firms to completely develop software applications that are in fact never used. Such failures are termed use *failures*.

CASE Tools

A wide variety of computer-aided software engineering (CASE) tools have been developed to support the activities that make up the various SDLC phases. The tools that are currently available address both technical and managerial activities including analysis and design, code generation, testing, and certain aspects of project management including cost estimation and scheduling. Tools that support the front end of the life cycle are sometimes referred to as upper CASE tools while tools that support the back end of the life cycle are called lower CASE tools. While there is benefit to be derived from using individual tools that are specially designed to support specific activities, many believe that the real power of CASE tools can only be achieved through an integrated CASE (I-CASE) tool environment. An I-CASE environment covers the entire life cycle of activities by providing a integrated set of tools that can share and maintain software engineering information all under a common user interface.

While various research studies have reported CASE tools to have a positive impact on the quality of developed systems (and to a lesser extent on the productivity of the software development process), these tools have not been as widely adopted as one might expect (18). Kemerer (19), for example, reports that one year after introduction, 70% of CASE tools are never used. One explanation for this is that the adoption and use of CASE tools represents a form of organizational change (20). The learning curve required to use the technology can be significant (19), particularly for organizations operating at a low level of software engineering maturity.

KEY DIMENSIONS OF PROJECT PERFORMANCE

Regardless of the type of project or technologies involved, there are three key dimensions of project performance: time, cost, and quality. While all three dimensions are critical to delivering a successful project, most project management experts would agree that (for most projects) managing time is the key to successfully controlling all three dimensions. If you fail to manage time, you can be reasonably certain that the project will fail on cost and quality as well.

It is, of course, quite difficult to manage all three of these dimensions. In recognition of this, some would advocate that the relative importance of each dimension be considered at the outset of the project and that the project be managed accordingly. For example, in the case of the Apollo missions to the moon, in which the goal was landing a man on the moon and returning him back to Earth safely, quality was the primary dimension by which project performance was judged. A secondary dimension was to accomplish this goal by the end of the decade (time). The third dimension was cost. In other words, quality and time were managed more tightly than cost. This prioritization of project performance dimensions has important ramifications in terms of whether a project is ultimately viewed as a success or failure. In the example just given, though sending a manned spacecraft to the moon and back may have cost more than originally budgeted, the mission was judged by most to be a highly successful project.

In the area of software development projects, which is the primary focus of this article, the same tradeoffs exist among the three project performance dimensions discussed earlier. In the context of software development, the three dimensions are sometimes referred to as: scope, time frame, and resources. Few software development managers are able to consistently deliver on all three dimensions. Many software project managers have therefore learned to practice a technique that provides some flexibility on at least one of the three project performance dimensions. In the words of one software project manager:

> You let users or management define any two of those and the third one pops out. They can define what they want, how much they want to spend on it, and given the resources you got the time frame will pop out automatically. Or they can set a date and what they want in terms of scope and you'll be able to tell them "this is what it's going to cost to do it." Or they can tell you a date and the money they have to spend and you can tell them what functionality they have. The third one pops out. You can't let them define all three or they will try to. They will try to define all three every time. They only get to define two—that's the rule. They'll want you to commit early, they'll want you to commit to a price before you've done any of the requirements definition—that's standard. Then, once you commit to that price they give you two-thirds of it—this is my experience. They'll approve two-thirds of it with the expectation that you've already overrun to the original number anyway. So they'll approve two-thirds and then they'll try to define all the scope and give you a really aggressive due date. And the scope equals whatever they haven't thought of yet that they decide they want.

PROJECT MANAGEMENT: A STEP-BY-STEP APPROACH

This section provides an overview of the key elements of project management in the form of a step-by-step approach.

Develop Project Charter

Establishing a clear project charter is critical to successful project management, yet it is a step that is frequently overlooked. The project charter is a mission statement that clearly defines the project (21). In essence, the project charter says: "This is what we are doing, and this is why we are doing it." The project charter establishes the project scope, objective (time, cost, and quality), assumptions, and constraints.

One purpose of the project charter is to ensure clear and consistent understanding of project scope and objectives. The project charter can also be used to establish measurable goals. The process of developing a project charter often helps to achieve consensus and commitment among project participants and key stakeholders.

The project charter should be a clear and concise document with realistic objectives. It should contain a brief description of the project's scope, the primary objective(s) of the project, targeted completion date, any constraints or assumptions that may affect the project, and a listing of the key personnel who will be responsible for managing the project. In helping to scope the project, identifying what will not be included is as important as identifying what will be included. Deliverables should be specified as clearly as possible.

Failure to develop a project charter usually means that there is a lack of definition or consensus about the project's scope and objectives. This, in turn, invites conflict both within the project team and between the project team and the customer for whom the work is being done. Failure to define the project clearly and to build commitment is the single greatest cause of project failure (21).

Establish Work Breakdown Structure

With the exception of very small, simple, projects, it is desirable to establish a work breakdown structure. A work breakdown structure is a logical hierarchy of tasks involved in a project. Developing a work breakdown structure involves decomposing a project into smaller and smaller tasks or work packages, which can then be assigned to specific individuals. In essence, the work breakdown structure provides a way of figuring out what activities have to be accomplished and who is going to perform them.

The work breakdown structure is developed from the top down as a means of identifying specific activities. An activity is anything that requires time. Activities generally have identifiable beginning and end points. Most activities produce an identifiable deliverable. Each activity should have one individual assigned as having responsibility for that activity. There are many different approaches for establishing a work breakdown structure. For software projects, it often makes sense to break the project down into phases such as: requirements analysis, design, development, testing, and implementation. These phases can correspond to the phases specified under the system development life cycle (SDLC) approach to managing software projects. Developing a work breakdown structure for a software project would involve taking each phase of the SDLC and breaking it down into smaller components. Under the phase system development we might lay out the different modules of the system that is to be developed. For each module, we might then provide a more detailed classification of tasks or activities that must be performed in order to complete the module.

In this section, we have suggested the SDLC as one means of developing a work breakdown structure for software projects. From a project management standpoint, once a work breakdown structure has been developed (based on the SDLC or some other approach), the next step is to examine the activities that have been identified and to analyze sequencing relationships.

Analyze Sequencing Relationships

In almost all projects, there are sequencing relationships among activities that must be accounted for in both planning and managing the project. There are a variety of network-based techniques that have been developed for accomplishing this. These include: critical path method (CPM), program evaluation and review technique (PERT), generalized precedence programming, and project simulation. Here, we will focus on what has probably become the most popular of these: CPM. Developed in the late 1950s by Du Pont and Remington Rand for plant maintenance projects, CPM provides a basic framework for project planning and control using one duration estimate for each activity. Under CPM, project activities are represented as nodes on a network. Precedence arrows between nodes A and B, for example, would indicate that activity A must be completely done before activity B can be started. While other types of precedence relationships are possible, this type which is called "finish to start" is the most common. It is the ability to handle precedence information that makes CPM preferable to the commonly used Gantt Chart.

To develop such a network, it is helpful to review the activities on the work breakdown structure and ask: "Which activities must be completed before this activity can start?" The exercise of developing a network diagram may also help to identify additional activities. When it is completed, the network becomes a model of the project, revealing how activities will progress as the project moves toward completion. In addition, the network diagram provides the foundation for further project planning.

Estimate Normal Activity Durations

After the initial network has been drawn along with all precedence relations, it is time to estimate normal activity durations. An activity duration is the amount of time between the start and completion of the activity. The normal duration of an activity is the duration associated with the most efficient use of resources (i.e., the lowest cost). While it may be possible to speed up an activity, this generally involves adding additional resources and such approaches should not be used in the initial estimation process.

Activity estimation begins with a definition of activity scope and content. The next step is to determine what ap-

proach will be used to complete the activity and who will be assigned the work. Knowing who will be assigned to the activity is critical to producing a reliable estimate of the activity's duration. In software programming, for example, a ten-fold difference in productivity between average and highly proficient programmers is not uncommon.

Once the approach and staffing issues have been determined, one can estimate the hours required to complete the activity. Before calculating a duration (usually in days), we must consider the average availability of the staff members assigned to a given activity. Duration can be estimated by dividing the hours required to complete the activity by the available hours per day. Initial duration estimates can then be adjusted, if needed, to allow for contingencies that are known to effect projects (e.g., equipment down time, etc.). In establishing duration estimates, it is important to obtain input and commitment from the individuals who will actually be doing the work. The estimates should be as realistic as possible (i.e., do not pad or low ball estimates). As a guidepost, duration estimates should not exceed one month. If they do, this is a signal that the activity needs to be subdivided into smaller activities.

Cost estimation has proven to be particularly problematic in software projects where cost overruns of 100–200% are not at all uncommon. Contributing factors include a chronic tendency to underestimate software projects, uncertainty about the requirements at the outset of the project, scope creep or changes in requirements that occur once the project is underway, insufficient effort to develop an historical database of experience with previous projects, and constant changes in the technological environment (i.e., new hardware and software platforms) that affect software productivity in new and (sometimes) unpredictable ways.

In response to the challenges in this area, considerable research has been directed at gaining a better handle on software cost estimation [see, for example, (22)]. Two common approaches toward estimation involve basing estimates on expert judgment and reasoning by analogy with one or more completed projects. Another approach, and the subject of much research, is algorithmic cost estimation models which estimate cost as a function of variables which are believed to be major cost drivers (22). Some of the early work on algorithmic cost estimation models was based on economic production functions derived from data on previous software development projects. The most well known of these models is Boehm's (12) COCOMO (Constructive Cost Model). Another stream of early work in this area is based on the Rayleigh curve which allows for the modeling of manpower buildup that typically occurs on software projects. The Rayleigh curve, for example, is the basis for Putnum's (23) SLIM (Software Lifecycle Model).

Both of the early approaches just described can be criticized on the grounds that they require an estimate of the number of source lines of code (SLOC) in order to generate cost and duration estimates. More recently, the function point method (24) has gained prominence. Function points operate at a more abstract level than SLOC and involve counting the number of user functions and adjusting for processing complexity. One advantage of function points over SLOC is that they may be easier to estimate at the early stages of the life cycle. Several cost estimation models have been developed based on a function point type of approach. An example of one such model is Estimacs (25).

A number of studies have attempted to validate various cost estimation models using empirical data. Kemerer (26) actually compares COCOMO, SLIM, Estimacs, and function points, concluding that all of the models must be customized to the environment in which they are to be used and that additional research is needed to understand the variables that impact software productivity. More recently, researchers have begun to explore the use of knowledge-based systems for cost estimation and the use of system dynamics to model software development (27).

Perform Basic Network Calculations

Once duration estimates have been made, it is time to perform network calculations and to identify the so-called "critical path" through the network. These calculations can be done by hand or, as is common today, using packaged software designed for supporting project management. The critical path refers to the "series of activities whose combined duration is the longest of any path through the project network" (21). It is important to note that a project can have more than one critical path. To shorten the duration of the project, it is necessary to shorten the durations of all critical paths. Any delay that occurs along a critical path will delay project completion. Thus, the critical path method helps by focusing attention on the areas that must be managed successfully if the project is to be completed on schedule. It is important to recognize that the network must be validated and revised throughout the course of the project. For large, complex, projects, it is sometimes useful to break the project into phases and to develop a network diagram for each phase. The current phase of the project can thus be modeled in detail, whereas subsequent phases that involve greater uncertainty can be initially modeled at a less detailed level. Models of subsequent phases can then be refined as the project progresses.

Analyze Time–Cost Tradeoff

Once the basic network calculations have been performed, it is necessary to conduct what is called a time–cost tradeoff analysis. This involves analyzing the total project cost which consists of both direct and indirect costs. Direct costs are the costs associated with the performance of the individual activities that make up the project (e.g., direct labor, materials, etc.). Indirect costs are costs that are associated with the project but are not related to individual activities (e.g., project manager, utilities, opportunity costs, etc.). The objective of the time–cost trade-off analysis is to minimize total project costs or to meet a required completion date as cost efficiently as possible. Often times, it is possible to reduce overall costs by finishing the project early. While indirect costs are lowered when the project is completed earlier, direct costs are raised. The tradeoff is usually such that savings can be realized from some compression of the project schedule. Too much compression, however, will lead to an increase in total project costs.

Schedule compression can best be achieved by shortening specific activities. This is known as crashing. While

activities that are on the critical path are the most obvious candidates for crashing, other activities such as those that involve relatively long durations and can be shortened at relatively low cost should also be examined. After identifying activities to be crashed, the network must be recalculated and examined for possible changes in the critical path (i.e., the critical path may shift and/or more paths can become critical). After achieving a final project duration, look for opportunities to uncrash activities that no longer appear on the critical path (21). It should be noted that highly compressed projects pose a danger in that we reach a point where there is no margin for error. Such compression should be avoided.

Resource Planning and Budgeting

After conducting the time–cost trade-off analysis, the next step is to load resources to activities. In this context, a resource is defined as any entity that contributes to the accomplishment of project activities. Resource loading requires identification of the "types and quantities of resources required to perform each activity in a project" (21). Resources should be loaded to one activity at a time and should be identified by name or by type (e.g., system analyst). Once resources have been loaded, one must ensure that the resources required are actually available to execute the project according to schedule. If the resource requirements (i.e., work load) exceeds available resources, there are a number of options available. One option is to find a way to temporarily add resources to the project either by working overtime, using temporary personnel, or contracting out some of the work. Resource leveling can also be used to resolve imbalances. This is done by delaying the start or extending the duration of activities that are not on the critical path. After resources have been loaded to activities, a project budget and cash flow plan should be developed. It is important to note that resource planning, budgeting, and cash flow analysis must account for simultaneous projects that compete for the same resources as well as nonproject-related work load.

Project Control

During the project, it is important to constantly monitor for deviations from the project plan, so that corrective action can be taken if needed. In this sense the project plan serves as the basis for controlling the project. Progress on activities should be updated regularly to determine the impact on project completion date before making any changes to the project schedule. For relatively small deviations against plan, it is often possible to make adjustments to the project without changing the basic project plan. For large deviations against plan, it may become necessary to replan the remainder of the project, starting with a clean sheet of paper. There is, after all, no sense in adhering to an original project plan if it has lost all credibility.

There are several key principles for maintaining control over projects. First, all project-related activities must be viewed with an eye toward completion. Working hard is not what ultimately matters; finishing activities is what is important to keeping the project under control. Second, project team members must be committed to accomplishing specific tasks under a short time horizon. Tasks that are too broad and time horizons that are too long will make the project that much more difficult to control. Third, slack should be preserved where possible by completing tasks as early as possible under the project plan. This strategy provides a safety cushion against contingencies that may arise later in the project. Finally, communication is critical to project control. Regular project control meetings should be held (at least monthly) so that problems can be put on the table for discussion and corrective action can be taken.

PROJECT MANAGEMENT RESOURCES

There are numerous resources on project management including journals and magazines such as the International Journal of Project Management, the Project Management Journal, and PM Network. Many countries also have organizations that provide information on project management and opportunities to interact with other project management professionals. Table 1 provides a listing of many such organizations along withcontact information for each.

Project Management Software Packages

In addition to the resources listed in Table 1, there are a wide variety of project management software packages available ranging in price from approximately one hundred dollars to several thousand dollars (28). Table 3 presents a representative sampling of software packages that are currently on the market, along with contact information for obtaining more details from software vendors. The products in the table can be broken down into three categories—high end, mid-range, and low end—based on the price and features that they provide. Generally speaking, what distinguishes the high-end products is their ability to handle multiple projects simultaneously. These packages are able to identify conflicting demands for the same resources and allow the user to set priorities among projects that must draw from the same resource pool. High-end products typically cost $2000 or more. Primavera's P3 product is an example of a high-end product. Mid-range products cost approximately $200–$500 and are typically geared toward managing a single project with up to around 2000 tasks. Mid-range products include Microsoft Project, Micro-Planner Manager, and Primavera's SureTrak.

Low-end products often cost less than $100 and provide basic support for applying some of the tools of project management, such as simple Gantt and PERT charts. Low-end products include: Milestones Simplicity, Project Vision, and Quick Gantt.

MODELS AND STANDARDS RELATING TO SOFTWARE PROJECT MANAGEMENT

There are a variety of models and standards relating to software project management. Prominent among these are: the software Capability Maturity Model (CMM), the ISO 9000 standard, and the SPICE standard. In a broad way, each of these standards attempts to address a common set of problems in software development, namely software that

Table 1.

Country	Organization	Contact Information
Global	Project Management Institute	Address: Project Management Institute Four Campus Boulevard Newtown Square, PA 19073-3299 Phone: +610-356-4600 URL: http://www.pmi.org Email: customercare@pmi.org
	International Association for Project and Program Management	Address: Renaissance Square 426 Main Street # 360 Spotswood, NJ 08884 Tel: +732-421-2306 URL: http://wwww.iappm.org? Email: info@iappm.org
Americas	Project Management Institute	Address: Project Management Institute Four Campus Boulevard Newtown Square, PA 19073-3299 Phone: +610-356-4600 URL: http://www.pmi.org Email: customercare@pmi.org
	American Society for the Advancement of Project management (asapm)	Email: info@aspm.org URL: http://www.asapm.org/
	The American Project Management Forum	(610) 734-3330 ext. 1045 Email: admin@pmi.org
Argentina	Project Management Institute	Tel.: 5411-4328-1007 Email: presidente@pmi.org.ar URL: http://www.pmi.org.ar
Brazil	Project Management Institute—Bahia	Tel: 55-71-273-7530 Fax: 55-71-273-7502 Email: andre@wbsitda.com.br
	Project Management Institute-Distrito Federal	Tel.: 5561-447-9661 Fax: 5561-272-3471 Email: Rodrigo@romasystems.com.br URL: http://www.pmidf.org.br
	Project Management Institute-Espriito Santo	Email: pmies@mpmies.org.br URL: http://www.pmies.org.br
	Project Management Institute-Fortaleza Ceara	Tel.: 55 85 3216-7864 Fax: 55 85 88444049 Email: cassio@pmice.org.br URL: http://www.pmice.org.br
	Project Management Institute-Goiania-Golas	Tel.: 55-62-231-6562 Email: sulemagobata@yahoo.com.br URL: http://www.pmigo.org.br
	Project Management Institute-Parana	Tel.: 55 41-3016 2101 Fax: 55-41-3016 2102 Email: souza@mp21.com.br URL: http://www.pmipr.org.br
	Project Management Institute-Santa Catarina	Email: Jackson.rovina@euax.com.br URL: http://www.pmisc.org.br

Table 1. (*Continued*)

Country	Organization	Contact Information
	Project Management Institute-Manaus	Email: luciano.torres@pmiam.org URL: http://www.pmiam.org.br
	Project Management Institute-Minas Gerais	Tel.: 55 31-3280 3302 Fax: 55-31-3280 3302 Email: presidencia@pmimg.org.br URL: http://www.pmimg.org.br
	Project Management Institute-Recife Pernambuco	Tel.: 55-81-9978-9432 Email: Douglas.nobrega@pmipe.org.br URL: http://www.pmibr.org.br
	Project Management Institute-Rio Grande do Sol	Tel.: 55-51-3319-1757 Fax: 55-51-3319-1757 Email: marco.kappel@pmirs.org.br URL: http://www.pmirs.org.br
	Project Management Institute-Rio De Janeiro	Tel.: 55-21-2262-8985/2262-8579 Email: angelo.valle@fgymail.br URL: http://www.pmirio.org.br
	Project Management Institute-Sao Paulo	Tel.: 5511-5041-4144 Fax: 5511-5531-1920 Email: presidencia@mpisp.org.br URL: http://www.pmisp.org.br
	Brazilan Association for Project Management (ABGP)	Email: abgp@abgp.org.br URL: http://www.abgp.org.br/
Canada	Canadian Project Forum	Tel. (403) 228-00885 Fax: (403) 228-3953 Email: foster@cadvision.com
	Project Management Institute Canada	Email: http://www.pmi.ca/
Chile	Project Management Institute-Santiago	Tel.: 56-2 422 0865 Fax: 56-2 422 0866 Email: msalmona@manguehue.net URL: http://www.pmi.cl
Columbia	Project Management Institute-Santafe de Bogota	Email: abernate@cable.net.co URL: http://www.pmicolumbia.org
Costa Rica	Project Management Institute-San Jose	Tel.: 506-290-3455 Fax: 506-290-3455 (call before faxing) Email: acruz@isthmusit.com
Mexico	Project Management Institute-Mexico	Tel.: 525-604-0472 Fax: 525-604-2681 Email: gersico@prodigy.net.mx URL: http://www.pmichapters-mexico.org/mexico
	Project Management Institute-Guadalajara	Tel.: 52-(33) 3667-7444 Fax: 52-(33) 3641-9748 Email: rcadena@apspro.com.mx
	Project Management Institute-Nuevo Leon	Tel.: +52-81-8220-9419 Email: rheredia@global.t-bird.edu
	Project Management Institute-Puebla	Tel.: 52/222/230/9979 Email: ramirei2@vw.com.mx
Panama	Project Management Institute-Panama	Tel.: 507-210-8514

Table 1. (*Continued*)

Country	Organization	Contact Information
		Email: folivares@blx.com URL: http://ww.pmi-pa.org/
Peru	Project Management Institute-Lima	Tel.: 51-1-313-3200 x205 Fax: +51-1-437-1606 Email: vvillar@casopisoft.com.pe
Uruguay	Project Management Institute-Montevideo	Email: eduardo.fleisher@gmail.com URL: http://ww.pmi.org.uy
Venezuela	Project Management Institute-	Tel.: +58-212-950-2864 Fax: +58-212-950-2179 Email: jtorrivilla@edelca.com.ve URL: http://ww.pmi-v.org.ve
Asia Pacific	Project Management Institute	Address: 73 Bukit Timah Road #03-01 Rex House Singapore 229832 Tel: +65 6330 6733 Fax: +65 6336 2263 Email: customercare.asiapac@pmi.org
	The Asia Pacific Regional Project Management Forum	Tel: 61-2-9252 7277 Fax: 61-2-9252 7077
Australia	Australian Institute of Project Management (AIPM)	URL: http://www.aipm.com.au/html/default.cfm Email: info@aipm.com.au Phone: +61 02 8288 8700 Fax: +61 02 8288 8711
	Project Management Institute—Adelaide	Tel: +61-8-8464-1627 Fax: 61-8-8464-2140 Email: Richard.tormet@eds.com
	Project Management Institute—Canberra	Tel: +61-0402062-728 Email: smitchell@primavera-aus.com URL: http://www.pmichapters-australia.org.au/canberra/home.asp
	Project Management Institute—Melbourne	Tel: +613 414 443 602 Email: president@melbourne.pmi.org.au URL: http://www.pmichapters-australia.org.au/melbourne
	Project Management Institute—Sydney	Tel: +61-411 431 676 Email: president@sydney.pmi.org.au URL: http://sydney.pmichapters-australia.org.au/
	Project Management Institute—Western Australia	Tel: +61-422 532 775' Fax: 618-9207-3236 Email: president@wapmi.org.au URL: http://www.wa.pmichapters-australia.org.au/
Azerbaijan	Azerbaijan Project Management Association	Email: office @azpma.net URL: http://www.azpma.net/
China	Project Management Research Committee	Tel: +86-29-8492484 Fax: +86-29-8494869 Email: pmrc@nwpu.edu.cn URL: http://www.pm.org.cn
	Project Management Institute—Hong Kong	Email: wong.patty@hit.com.hk URL: http://www.pmi.org.hk

Table 1. (*Continued*)

Country	Organization	Contact Information
India	Project Management Associates	Tel: +91 120 242 0444/0463 Fax: +91 120 242 1484/1482 Email: pma1@vsnl.com URL: http://www.pma-india.org
	Project Management Institute—Bangalore	Tel: +91 90 2237 0400 Email: amarbhaskar@pmibangalorechapter.org URL: http://www.pmichapterbangalore.org/
	Project Management Institute—Chennai	Tel: +91 442623880 X6405 Fax: 91442628171 Email: smurali@covansys.com URL: http://www.pmi-channai.org/
	Project Management Institute—Mumbai	Email: president@mpmimumbaichapter.org URL: http://www.pmimumbaichapter.org/
	Project Management Institute—New Delhi	Tel: +91 120-531-5760 Fax: +91-120-531-5750 Email: upendra@astrowix.com URL: http://www.pminorthindia.org/
	Project Management Institute—Hyderabad	Tel: +91 40-23100494 Fax: 91-40-23100892 Email: confsec@pmi-pcc-org URL: http://www.pmipcc.org/
	Project Management Institute—Pune-Deccan	Tel: +91 20-39875003 Fax: +91-20 39875001 Email: sandeep.shouche@bmc.com URL: http://finance.groups.yahoo.com/group/pmi-pune-chapter/
	Project Management Institute—Trivandrum, Kerala	Tel: +91/471/2527441 Email: cbraiesh@nestec.net URL: http://www.pmikerala.org
Indonesia	Project Management Institute—Jakarta	Tel: +62817901057 Email: aprasetvo@xl.co.id URL: http://www.pmijakarta.org/
Japan	Engineering Advancement Association of Japan (ENAA)	Tel: +81-45-721-7606 Fax: +81-45-716-7833 Email: tanaka.01818@oct.jgc.com.jp
	Project Management Forum	Tel: 81-3-350244441 Fax: 81-3-3502-5500 Email: jpmf-adm@enaa.or.jp
	Project Management Institute—Tokyo	Tel: +81-3-5847-7301 Fax: 0081-3-3664-9833 Email: pmtksecr@pmi-tokyo.org URL: http://www.pmi-tokyo.org/
Kazakhstan	Kazakhstan Project Management Association	Email: prom@intelsoft.kz
Korea	PROMAT—Korean Institute of Project Management and Technology	Email: promat@promat.org
Malaysia	Project Management Institute—Jakarta	Tel: 603-269-84837 Fax: 603-269-82107 Email: chaney672001@yahoo.com URL: http://www.pmimy.org/

Table 1. (*Continued*)

Country	Organization	Contact Information
New Zealand	Project Management Institute New Zealand Incorporated	Email: president@pmi.org.nz URL: www.pmi.org.nz
	Project Management Institute	Tel: +64 4 970 2005 Fax: 64-3-351-4554 Email: president@pmi.org.nz URL: http://www.pmi.org.nz/
Pakistan	Project Management Association of Pakistan	Email: suhail@syscompk.com
	Project Management Institute—Islamabad	Email: info@pmi-islamabad.org URL: http://www.pmi-islamabad.org/
	Project Management Institute—Karachi	Tel: 92-300-82-92375 Fax: 815-377-6581 Email: mamirza66@wtmeca.net URL: http://www.pmikarachi.org/
	Project Management Institute –Lahore	Tel: +92(42)9203931-34 Fax: +92(42)920935 Email: khalid@pmupunjab.gov.pk URL: http://www.pmilhr.org.pk/
Philippines	Project Management Institute	Tel: +632-631-7487 Fax: +632-631-2786 Email: Patrick.ferrer@pmi.org.ph URL: http://www.pmi.org.ph/
Singeapore	Project Management Institute	Email: wayne.herbert@catalystprojectsolutions.com URL: http://www.pmi.org.sg/
Sri Lanka	Project Management Institute	Email: lalith36@hotmail.com URL: http://www.pmicolombo-srilanka.org/
Taiwan	Project Management Institute	Tel: +886-2-2523-5808 Fax: 886-2-25232090 Email: barry@pmi.org.tw URL: http://www.pmi.org.tw/
	Taiwan Project Management Association, China (TPMA)	Email: jpm@tpma-tw.org URL: http://www.tpma-tw.org
Thailand	Project Management Institute	Tel: +66 2 661 3850 Email: Michael@pmithai.org URL: http://www.pmithai.org/
EMEA	Project Management Institute	Address: 300, Avenue Tervueren B-1150 Brussels, Belgium Phone: +32-2-743 15 73 Fax: +32-2-743 15 50 Email: customercase.emea@pmi.org
Austria	Project Management Institute—Vienna	Tel.: +43 664 144 2300 Fax: +43 2243 9821515 Email: m.engelhardt@pmi-austria.org URL: http://www.pmi-austria.org
	Projeckt Management Austria	Email: office@p-m-a.at URL: http://www.p-m-a.at/
Belgium	Project Management Institute	Tel.: +32-47-395-1627 Email: chris.kindermans@pmi-belgium.be URL: http://www.pmi-belgium.be

Table 1. (*Continued*)

Country	Organization	Contact Information
Bulgaria	Bulgarian Project Management Association (BPMA)	Email: bpma@progject.bg URL: http://www.project.bg
Cameroon	Project Management Institute	Tel.: 301-669-1897 Email: francisdw@msn.com
Croatia	Project Management Institute	Tel.: 385-91-365-4730 Fax: 385-91365-3548 Email: sinsisa.krainovic@ericsson.com URL: http://www.pmi-croatia.hr
	Croatian Association for Project Management	Email: capm@grad.hr URL: http://www.capm.hr/
Czech Republic	Project Management Institute	Tel.: +420-261-307-337 Fax: +420-261-307-294 Email: petr.sestak@hp.com
	Project Management Association Czech Republic (SPR)	Email: chlupaty@ipma.cz URL: http://www.ipma.cz/
Denmark	Project Management Institute	Tel.: +45-2711-4142 Email: president@pmi-dk.org URL: http://www.pmi-dk.org
	Association of Danish Project Management	Email: info@projektforeningen.dk URL: http://www.projektforeningen.dk/
	Forening for Dansk Projektledelse	URL: http://www.projektforeningen.dk Phone: +45-42-26-78 77 Fax: +45-4824 06 50
Egypt	Project Management Institute	Tel.: +02-346-1046 Fax: +02-346-1046 Email: gamal.nassar-cec@yahoo.com
	Management Engineering Society (IES)	Email: rumes@rusys.eg.net URL: http://www.ese.eg.net/
Finland	Project Management Institute	Tel.: 358-50-380-8520 Fax: 358-9-665-771 Email: seppo.haimenen@nokia.com
	Project Management Association Finland	Email: pry@pry.fl URL: http://www.pry.fl/
France	Project Management Institute—Saint-Brieuc	Email: pierre-yves.thomas@laposte.net
	Project Management Institute—Centre Est	Tel.: 33386492600 Fax: 33386464700 Email: agenet@acta-mobilier.fr
	Project Management Institute—SUD	Tel.: 33 679 8498 00 Fax: 33/4/92/96/64/91 Email: operto@pmi-fr.org URL: http://www.pmi-fr.org/index.asp?url=main.asp&chap=35

Table 1. (*Continued*)

Country	Organization	Contact Information
	Project Management Institute—Hauts-de-France	Tel.: 33-03-20-21-59-72 Fax: 33-03-20-21-59-74 Email: cbredillet@pmi-fr.org URL: http://www.pmi-fr.org/index.asp?url=main.asp&chap=37
	Project Management Institute—Ile de France	Tel.: 03 44 29 09 89 Email: m.desrumaux@pmi-fr.org URL: http://www.pmi-fr.org/
	Association Francophone de Management de Projet (AFITEP)	Email: info@afitep.fr URL: http://www.afitep.fr/
	SMAP Association Francaise pour l' avancement du Management de Projet	Email: veronique.laborderie@wanadoo.fr
Germany	Project Management Institute—Berlin/Brandenburg	Tel.: 49-700-87437833 Fax: 49-700-87437832 Email: Steffi@triest.de URL: http://www.pmi-berlin.org/
	Project Management Institute—Cologne	Tel.: 49-2235-985-401 Fax: 49-2235-985-402 Email: president@pmicc.de URL: http://www.pmicc.org/
	Project Management Institute—Frankfort	Email: president@pmifc.de URL: http://www.pmifc.org/
	Project Management Institute—Munich	Tel.: 49-172-6300285 Fax: 49-721-4908824 Email: Thomas.wuttke@9pm.de URL: http://www.pmi-muc.org/
	GPM Deutsche Gesellshaft fur Projektmanagement e. V.	URL: http://www.gpm-ipma.de/ Phone: +49 0911/43 33 69-0 Fax: +49 0911/43 33 69-99
Greece	Project Management Institute	Tel.: +30 693 22.13.502 Email: Theofanis.Giotis@ITEC.edu URL: http://www.pmi-greece.org/
	Network of Project Managers in Greece (PM-Greece)	Email: pmgreece@pmgreece.gr URL: http://www.pmgreece.gr/
Hungary	Project Management Institute—Berlin/Brandenburg	Tel.: 361-1-2299200 Fax: 361-1-2299000 Email: imre.szalay@hp.com URL: http://www.pmi.hu/
	Project management Association Hungary (FOVOSZ)	Email: fovosz@fovosz.hu URL: http://wwwfovosz.hu/

Table 1. (*Continued*)

Country	Organization	Contact Information
Iceland	Project Management Association of Iceland (VSF)	Email: vsf@vsf.is URL: http://www.vsf.is/
Ireland	Project Management Institute	Email: Armin.Hamidovic@ericsson.com URL: http://www.pmi-ireland.org/
	Institute of Project Management Ireland	Email: info@projectmanagement.ie URL: http://www.projectmanagement.ie/
Israel	Project Management Institute	Tel.: +972-3-960-05-63 Email: shay@pmi.org.il URL: http://www.pmi.org.il/
Italy	Project Management Institute—Northern Italy	Tel.: +39-0245409029 Fax: +39-0245409303 Email: carlo.notari@poste.it URL: http://www.pmi-nic.org/
	Project Management Institute—Rome	Tel.: 39-02-520-33556 Fax: 39-2-520-43317 Email: franco.quarrella@snampro.co.uk URL: http://www.chapter.pmi.org/rome/main.htm
	Project Management Institute—Southern Italy	Email: presidente@pmi-sic.org URL: http://www.pmi.sic.org
	Associazone Nazionale di Implantistica Industriale	Email: segreteria@animp.it URL: www.animp.it
Kuwait	Kuwait Society of Enginners (KSE)/Kuwait PM Certification Body (KPMC)	Email: kse@kse.org.kw URL: http://www.kse.org.kw/
Latvia	Latvian National Project Management Association	Email: pasts@lnpva.lv URL: www.lnpva.lv
Lebanon	Project Management Institute	Tel.: +359 2 933 19 10 Fax: +359 2 933 19 90 Email: jimmy.char@bull.bg
Netherlands	Project Management Institute	Tel.: +31630177016 Email: richardvruler@wanadoo.nl URL: http://www.pmi-netherlands-chapter.org/
Nigeria	Project Management Institute—Lagos	Tel.: +23-480-320-02181 Email: ini.umoren@pmilagos.org URL: http://www.pmilagos.org/
Norway	Project Management Institute—Oslo	Tel.: +47-906 67 243 Email: elisabeth.svendsen@terramar.no URL: http://www.pmi-no.org/
	Project Management Institute—Western	Tel.: +47-99-52-3144 Email: post@pmi-now.org URL: http://www.pmi-now.org/
	Norwegian Association of Project Management	Email: else.dahl@tekna.no URL: www.prosjektledelse.com
Poland	Project Management Institute—Poznan	Tel.: 48-50-1002019 Email: richardvruler@wanadoo.nl

Table 1. (*Continued*)

Country	Organization	Contact Information
	Project Management Institute—Warsaw	Tel.: 48-60-143-8727 Email: pimalec@neostrada.pl URL: http://www.pmi.org.pl/
	Stowarzyszenie Project Management Polska	Email: biuro@spmp.org.pl URL: www.spmp.org.pl
Portugal	Project Management Institute	Tel.: +351 938-466-517 Fax: +351 253 510 250 Email: alexandre.rodrigues@pmi-portugal.org URL: http://www.pmi-portugal.org/
	Asociacao Potugesa de Getao De Projectos (APOGEP)	Email: info@apogep.pt URL: www.apogep.pt
Romania	Project Management Institute	Email: apienaru@yahoo.com URL: http://www.Pmi.ro/
	Project Management Romania	Email: office@pm.org.ro URL: www.pm.org.ro
Russia	Russian Project Management Association (SOVNET)	Tel: +7 095 913-7162 Fax: +7 095 913-9128 Email: sovnet1@cityline.ru URL: www.sovnet.ru
	Project Management Institute - Moscow	Tel.: 7-095-502-3194 Fax: 7-095-246-6309 Email: bazhenov@pmo.ru
	Project Management Institute—St. Petersburg	Tel.: 7-8912-237-0763 Fax: 7-812-237-0579 Email: info@pmi.spb.ru URL: http://www.Pmi.spb.ru/
Serbia and Montenegro	Project Management Association of Servia and Montenegro	Email: yupma@fon.bgt.ac.yu URL: http://www.upma.org.yu
Saudi Arabia	Project Management Institute	Tel.: 874-6646 Fax: 873-7828 Email: tofig.gabsani@aramco.com URL: http://www.pmi-agc.com/
Slovakia	Project Management Association of Slovakia	Email: sppr@sppr.sk URL: http://www.sppr.sk
Slovenia	Project Management Institute	Tel.: 386/1/3009/800 Fax: 386/1/3009/820 Email: saso.navkovic@ipmit.si URL: http://www.pmi-slo.org/
	Project Management Association of Slovenia (ZPM)	Email: info@zpm-si.com URL: http://www.zpm-si.com
South Africa	Project Management Institute	Tel.: +27 11 530 9700 Fax: +27 11 880 7079 Email: mbassuni@yahoo.com
	APM (SA)	Email: apmsa@whp.co.za URL: http://www.cranefield.ac.za/
Spain	Project Management Institute—Barcelona	Tel.: +34 934 016 647 Email: serer@pmi-bcn.org URL: http://www.pmi-bcn.org/

Table 1. (*Continued*)

Country	Organization	Contact Information
	Project Management Institute - Madrid	Tel.: 34-91-659291317 Email: jpuentes@direccciondeproyecvtos.org URL: http://www.pmi-es.org/
	Project Management Institute - Valencia	Tel.: +34 687 528 663 Email: info@pmi-valencia.org URL: http://www.pmi-valencia.org/
	Asociacion Espanola de Ingenieria de Proyectos (AEIPRO)	Email: secretaria@aiepro.com URL: http://www.aeipro.com
Sweden	Project Management Institute	Tel.: +46-73-684-0300 Fax: +46-73-270-8499 Email: patrick.bergstrom@semcon.se URL: http://www.pmi-se.org/
	Svenskt ProjektForum (Swedish Project Management Society)	Email: info@projektforum.se URL: http://www.projektforum.se
Switzerland	Project Management Institute	Fax: +41-61-641-7131 Email: james.greene@pmi-switzerland.ch URL: http://www.pmi-switzerland.ch/
	Schweizerische Gesellschaft fur Projektmanagement	Email: spm@spm.ch URL: http://www.spm.ch
Turkey	Project Management Institute	Tel.: 90-312-441-3700 Email: emre.ozbaqci@ankara.ilf.com
Ukraine	Project Management Institute	Tel.: +380-44-205-3280 Email: info@pmi.org.ua URL: http://www.pmi.org.ua/
	Ukrainian Project Management Association (UPMA)	Email: upma@upma.kiev.ua URL: http://www.upma.kiev.ua
United Arab Emirates	Society of Engineers United Arab Emirates (SEUAE)	Email: uaesoe@emirates.net.ae
	Emirates Project Management Associates	Email: epma@hct.ac.se
United Kingdom	Project Management Institute	Tel.: +44 (0) 208-751-5626 Email: president@pmi.org.uk URL: http://www.pmi.org.uk/
	Association for Project Management (APM)	Email: info@apm.org.uk URL: http://www.apm.org.ul/
Yugoslavia	Project Management Institute	Email: miroslav_jakovic@yahoo.com URL: http://www.pmi-yu.org/
Zambia	Project Management Associatio of Zambia	Email: pmazambia@yahoo.com

is delivered late, over budget, and full of bugs (i.e., defects). The purpose of these models and standards is to advance the state of practice of software engineering and to improve the quality of software and systems that depend upon software.

Capability Maturity Model

The Capability Maturity Model is a model for judging the maturity of an organization's software development practices and for identifying the key practices that are required in order to increase the maturity of these practices. The CMM was developed under the stewardship of the Software Engineering Institute at Carnegie Mellon University. The CMM is designed to help organizations improve the maturity of their software processes along an evolutionary path that runs from ad hoc, chaotic, processes to mature, disciplined processes. The CMM is organized into five maturity levels as defined below (29):

1. Initial The software process is characterized as ad hoc or even chaotic. Few processes are defined, and success depends on individual effort and heroics.
2. Repeatable Basic project management processes are in place to track cost, schedule, and functionality. The processes needed to repeat earlier successes on similar applications are in place.

Table 2.

Product	Product Link	Vendor	Vendor Link
@Work Business Productivity Suite	http://www.cpts.com/AtWorkSuite.asp	Critical Path Technical Services	http://www.cpts.com/newhome.asp
@Task Enterprise Project Management	http://www.attask.com/	AtTask	http://www.attask.com/
AceProject	http://www.aceproject.com/	WebSystems, Inc.	http://www.aceproject.com/contact.htm
AdHoc Gantt Chart for Lotus Notes & Domino	http://www.adhocsystems.com/ganttchart	Ad Hoc Systems	http://www.adhocsystems.com/
Artemis 7	http://www.aisc.com/product/1	Artemis International solutions	http://www.aisc.com/Company/0
Artemis 9000	http://www.aisc.com/Product/3	Artemis International solutions	http://www.aisc.com/Company/0
Artemis OnTrak	http://www.aisc.com/Product/8	Artemis International solutions	http://www.aisc.com/Company/0
Artemis Views	http://www.aisc.com/Product/2	Artemis International solutions	http://www.aisc.com/Company/0
Autotask	http://www.autotask.com/landing/project_management.htm	AutoTask Corporation	http://www.autotask.com/company/index.htm
Borland Tempo	http://www.borland.com/us/products/tempo/index.html	Borland	http://www.borland.com/us/company/index.html
CA Clarity	http://www.niku.com/project-portfolio-management-61.html	CA	http://www.niku.com/ww/about-clarity-81.html
Celoxis	http://www.celoxis.com/	Celoxis Technologes	http://www.celoxis.com/
Centric Project	http://www.centricsoftware.com/default.asp?url=products&section=centric_project	Centric Software	http://www.centricsoftware.com/
Copper 2005: Project Management Software	http://www.copperproject.com/	Element Software	http://www.copperproject.com
DecisionCharts for Microsoft Project	http://www.decisionedge.com/microsoft_decisioncharts.html	DecisionEdge	http://www.decisionedge.com
DOVICO Timesheet: Project Management	http://www.dovico.com/	DOVICO Software	http://www.dovico.com/about.html
Easyplan	http://www.astadev.com/software/easyplan/index.asp	Asta Development	http://www.astadev.com
EnterPlicity	http://www.teaminteractions.com/products.aspx	Team Interactions, Inc.	http://www.teaminteractions.com/
Enterprise Project Management Solutions	http://www.deltek.com/products/evp/default.asp	Deltek	http://www.deltek.com/
EPAM Project Management Center	http://www.epam-pmc.com/	EPAM Systems, Inc.	http://www.epam-pmc.com/company.html
eProject	http://www.eproject.com/?search=softwarenetwork_ppm	eProject	http://www.eproject.com/
eStudio: Web-based Project Management & more	http://www.same-page.com/	Same-Page	http://www.same-page.com/contact.html
faces	http://faces.homeip.net/	Open Source	
FastTrack Schedule	http://www.aecsoft.com/products/fasttrack/	AEC Software, Inc.	http://www.aecsoft.com/company/about/
Genius Enterprise Project	http://www.geniusinside.com/web/website.nsf/home	Genius Inside	http://www.geniusinside.com/web/website.nsf/home

Table 2. (*Continued*)

Product	Product Link	Vendor	Vendor Link
icTracker	http://www.ic-soft.com/products/ ictracker.htm	IC Soft, inc.	http://www.ic-soft.com/products/ictracker.htm
i-lign	http://www.ilign.com/	I-lign, ltd.	http://www.ilign.com/about.php
Intellect EPM: Executive Project Management	http://www.interneer.com/ sofnet/ppm.asp	Interneer	http://www.interneer.com
IntelligenceSoft web-based Project Management System	http://pms.intsoft.spb.ru/fp_aboutpms.asp	Intelligence Soft	http://www.intsoft.spb.ru/
Intellisys	http://www.webintellisys.com/	Intellisys, Inc.	http://www.webintellisys.com/company.html
LeadingProject	http://www.leadingproject.com/ en/main.php	Leading Project	http://www.leadingproject.com/en/contact/main.php
ManagePro and MProWeb	http://www.performancesolutionstech.com/ order_online.asp#	Performance Solutions Technology	http://www.performancesolutionstech.com/about.asp
Micro Planner Manager/X-pert	http://www.microplanning.com/ Default.aspx?tabid=2097	Micro Planning International	http://www.microplanning.com/Default.aspx?tabid=2090
Milestones	http://www.kidasa.com/	Kidasa Software, Inc.	http://www.kidasa.com/press/company_info.htm
MinuteMan Project Management Software	http://www.minuteman-systems.com/ products.htm	MinuteMan Systems	http://www.minuteman-systems.com/
MS Project and MS Project Server	http://www.microsoft.com/office/ project/prodinfo/default.mspx	Microsoft, Inc.	http://www.microsoft.com/
Open Workbench	http://www.openworkbench.org/	Open Source	
OpenAir Web-Native PSA Software Solution	http://www.openair.com	OpenAir	http://www.openair.com/home/about.html
OPMcreator	http://www.opmcreator.com/		http://www.opmcreator.com/
PERT Chart EXPERT	http://www.criticaltools.com/ pertmain.htm		
Portfolio Intelligence	http://www.3olivesolutions.com/ products_services.asp	3 Olive Solutions	http://www.3olivesolutions.com/
PPM6: Industry-leading Web Based Project Portfolio Management Solution	http://www.eproject.com	eProject	http://www.eproject.com/company/index.htm
Primavera	http://www.primavera.com/	Primavera Systems	http://www.primavera.com/about/index.asp
Project Collaboration Solution	http://www.concertosupport.co.uk/ lg-project-management-software.asp	Concerto Support Services	http://www.concertosupport.co.uk/
Project Insight. NET	http://www.projectinsight.net/	Project Insight	http://www.projectinsight.net/AboutUs/default.aspx
Project KickStart	http://www.projectkickstart.com/ html/pkswin3.cfm?riskyp	Experience in Software, Inc.	http://www.projectkickstart.com/html/aboutus.htm
Project Office Management System (POMS)	http://www.pragsoft.com/	Pragsoft Corp.	http://www.pragsoft.com/company.html
Project Scheduler 8.5	http://www.sciforma.com/products/ ps_suite/ps8_overview.htm	Sciforma Corp	http://www.sciforma.com/about_sciforma.htm
Project.net	http://www.project.net/index.jsp	Project.net	http://www.project.net/company.htm
Project_Insight	http://www.projectinsight.net/ ProductInformation/default.aspx	Metafuse,inc.	http://www.projectinsight.net/AboutUs/AboutMetafuse.aspx
ProjectST	http://www.projectst.com/	EBTek, LLC	http://www.projectst.com/about.html

Table 2. (*Continued*)

Product	Product Link	Vendor	Vendor Link
PS Next	http://www.sciforma.com/Products/PSNext/PSNext_Overview.htm	Sciforma Corp	http://www.sciforma.com/about_sciforma.htm
RiskyProject—Project Risk Management Software	http://www.riskyproject.com/	Intaver Institute	http://www.riskyproject.com/index-6.html
Sciforma Web/Desktop/Process Project Management solutions	http://www.sciforma.com/products/products.htm	Sciforma Corp	
Smooth Projects	http://www.smoothprojects.com/	123 Smooth Projects	http://www.smoothprojects.com/
TargetProcess	http://www.targetprocess.com/	Target Process	http://www.targetprocess.com/
Taskland—Affordable Project Management for Small- & Medium-sized Businesses	http://www.taskland.com/index.php	Taskland.com	http://www.taskland.com/ws/aboutus.php
TeamHeadquarters-Unified Team Management Solution	http://www.entry.com/products.html	Entry Software Corp	http://www.entry.com/
ThinMind—Expense	http://www.thinmind.com/	Shift Technologies, Inc.	http://www.thinmind.com/Corporate.htm
TurboProject	http://www.officeworksoftware.com/productfamily_TP.php	OfficeWork Software	http://www.officeworksoftware.com/contact.php
ValleySpeak Project Server	http://www.valleyspeak.com/	ValleySpeak	http://www.valleyspeak.com/company.html
VERTABASE PRO	http://www.vertabase.com/	Vertabase	http://www.vertabase.com/company_info.html
VPMi web-based project management office	http://www.vcsonline.com/	VCS	http://www.vcsonline.com/VCS/About_Us/VCS_About_Us.htm
WBS Chart Pro	http://www.criticaltools.com/wbsmain.htm	Critical Tools	http://www.criticaltools.com/compinfo.htm
Web TimeSheet: Project Costing and Time Tracking Software	http://www.replicon.com/	Replicon	http://www.replicon.com
Web-based Project Management	http://www.autotask.com/landing/project_management.htm	Autotask	http://www.autotask.com/company/index.htm
White Cloud Systems	http://www.whitecloudsystems.com/products.htm	White Cloud Systems	http://www.whitecloudsystems.com/

3. Defined The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the entire organization. All projects use this process, or a customized version of this process, for developing software applications.
4. Managed Detailed measures of the software process and product quality are collected. Both the process and product are understood and controlled in a quantitative way.
5. Optimizing Continuous process improvement has been implemented based on quantitative feedback from the process.

ISO 9000

In 1987, the International Organization for Standardization (ISO) published the core ISO 9000 standards. Founded in 1946, the ISO publishes worldwide standards for manufacturing, trade, and communications. The ISO 9000 family of standards relates to quality management and quality assurance and represents an international consensus on the essential features of a quality system. The ISO 9000 series "specify quality system requirements for use where a contract between two parties requires the demonstration of a supplier's capability to design and supply product" (29). More than 80 countries have adopted the ISO 9000 series as national standards. ISO 9001 provides the standard for quality assurance in "design/development, production, installation, and servicing." While ISO 9001 applies to software as well as the manufacture of hard goods, certain unique aspects associated with software prompted the development of a special guideline (ISO 9000-3) for applying ISO 9001 to the development, supply, and maintenance of software.

Among other things, ISO 9001 requires that a documented quality system be established, with procedures to control and verify the design and to control the distribution and modification of documents and data. ISO 9001 requires that production processes be defined and controlled, and that appropriate testing be performed prior to the release of the finished product. ISO 9000 also contains provisions

for product identification and traceability during all stages of production, delivery, and installation. These and other requirements articulated under ISO 9001 can be mapped against the CMM. While there are some areas covered by ISO 9001 that are not covered under CMM and vice versa, there are significant areas of overlap between the two. In general, CMM appears to be more detailed and more specific in that it is specifically tailored for a software process environment, while the ISO standards are intended to be more generic. In comparing and contrasting ISO 9001 versus CMM, Paulk et al. (29) conclude that "an organization that obtains and retains ISO 9001 certification should at least be close to Level 2" on the CMM.

SPICE

SPICE (Software Process Improvement and Capability determination) is an international initiative under ISO designed to serve as a standard for software process assessment. SPICE and ISO 9000-3 represent separate efforts by different bodies within ISO. SPICE incorporates the spirit of ISO 9000 (i.e., providing confidence in a supplier's quality management), while providing a framework for assessing potential suppliers'process capability. Inspired in part by the CMM, SPICE provides (among other things) a standard for the activities that are essential to good software engineering practice. Like CMM, these activities are structured according to increasing levels of process maturity.

IDENTIFYING AND MANAGING SOFTWARE PROJECT RISK

The models and standards just described are aimed at improving software quality; hence, they focus on software engineering and management processes that can reduce technical defects. It is important to remember, however, that many software project failures arise not because of technological problems but because of organizational or user-related issues. It is the assessment and control of these risks that we shall now discuss.

Key Risk Factors

Since the 1970s, both academics and practitioners have written about the risks associated with managing software projects [e.g., (30–58)]. A study conducted by Schmidt, Lyytinen, Keil, and Cule (37) represents one of the most rigorous investigations on the subject of software project risk. In this study, panels of experienced software project managers were assembled in different parts of the world—in this case, Finland, Hong Kong, and the United States. The experts were then asked to first identify specific risk factors and then rank and rate them in terms of their importance.

Perhaps the most interesting finding reported by Schmidt et al. (37) is that three independent panels, representing very different countries and cultures, selected a common set of 11 risk factors as being among the more important items. While there were differences across panels in the level of importance ascribed to some of these risk factors, the fact that all three panels independently selected these 11 risk factors suggests the existence of a universal set of risks, relevant around the globe. These 11 risk factors are listed below in decreasing level of importance (averaging across the three panels) from top to bottom:

1. Lack of top management commitment to the project
2. Failure to gain user commitment
3. Misunderstanding the requirements
4. Lack of adequate user involvement
5. Failure to manage end user expectations
6. Changing scope/objectives
7. Lack of required knowledge/skills in the project personnel
8. Lack of frozen requirements
9. Introduction of new technology
10. Insufficient/inappropriate staffing
11. Conflict between user departments

While emerging technology projects (e.g., client/server, object oriented, Web-based) involve greater risk than projects involving more mature technologies, it is interesting to note that just one of the 11 items directly relates to technological risk. The three top-rated risks involve management (not technology) issues and are discussed in more detail below.

Lack of Top Management Commitment to the Project

Many panelists saw the lack of top management commitment as a risk that overshadowed all others. In the words of one panelist, "if this is not present, then all other risks and issues may be impossible to address in a timely manner." Similar comments were made by other panelists:

> If you have top management support most of remaining risk areas can be managed and resolved . . . The events that can cause a project to fail can often be countered if top management is committed to the success of the project. These events can range from widespread changes in the organization to changing requirements. Executive support is critical to keeping the project "on track" in the wake of these changes.

Failure to Gain User Commitment

Another prime area of concern to the panelists was the failure to gain user commitment. User commitment was seen as critical for two reasons: (1) because it helps ensure that users are actively involved in the requirements determination process, and (2) because it creates a sense of ownership, thereby minimizing the risk that the system will be rejected by those it was created to serve. To some, strong user commitment was seen as something that could even compensate for a lack of executive commitment. The following remark offered by one of the panelists was typical:

> The users of the system to be delivered are the ultimate customer of the deliverable . . . If the users are not committed to a joint effort in which they are heavily involved in the effort, there is a high risk

> of assuming their detailed functional and business requirements. Without their commitment, they withdraw critical feeling of ownership and the project has a high chance of missing the mark. Even executive commitment lacking can be overcome by total customer/user commitment.

Misunderstanding the Requirements

Misunderstanding the requirements was also viewed as a critical risk factor because, in the words of one panelist, "requirements drive the entire project." Without a proper systems analysis to develop a complete and accurate set of requirements there is a distinct possibility of building a system that no one wants to use. For this reason, many panelists underscored the importance of understanding the requirements, with the following remark being typical:

> While [many] factors are very important and capable of defeating a project, it seems to me that this one is one of the very few that will cause failure every single time. If you don't know what you're shooting for, then there is absolutely no chance of achieving anyone's definition of success.

Assess Risks Early (and Often) During the Development Process

The 11 risk factors mentioned earlier can serve as a useful checklist for performing software project risk assessments. Such an assessment is particularly important in managing large, cross-functional projects that span multiple budget cycles. Unfortunately, for most software projects, such risk assessments are usually conducted on an infrequent and informal basis if they are even conducted at all (38).

Based on the study conducted by Schmidt et al. (37), it would appear that the majority of the 11 risk factors can be boiled down into two key areas to focus on from a risk management perspective: (1) customer mandate, and (2) scope and requirements. Each of these is discussed in more detail below, along with possible risk mitigation strategies.

Successful projects are very often those that have the commitment of both senior management and those who will actually use the system. Without a clear charter, or mandate, the project is simply not viable. Relevant questions for project managers to ask are: Does this project have senior management commitment? Does it have user commitment? In short, is there a clear charter or mandate for completing the project? Examples of specific risk factors that could be classified as being related to customer mandate include: lack of top management commitment, failure to gain user commitment, inadequate user involvement, and failure to manage end user expectations.

Risk mitigation strategies that emphasize relationship management are needed in order to control these risks. An essential element to this relationship building is the project manager's need to build and maintain trust with the users by meeting commitments. Once a project has started, project managers must periodically gauge the level of commitment from both top management and the user community to avoid being caught in a situation where support for the project suddenly evaporates. One approach to maintaining commitment is Theory-W (39), which involves structuring the project to meet the "win" conditions of various stakeholders. Another approach is to actively manage end user expectations. Problems with user acceptance can occur whenever user expectations are not realistic. As one project manager in the Schmidt et al. (37) study observed: "A project manager can do everything right, yet still fail if his/her clients expected more."

Another key area for risk management involves scope and requirements issues. Many of the risks that threaten software projects involve the ambiguities and uncertainties in this area. Relevant questions for project managers to ask are: What is inside the scope of the project and what is outside the scope of the project? What functionality is essential to be successful versus "nice to have?" In short, do I know what I am building? Examples of specific risk factors that could be classified as being related to scope and requirements include: misunderstanding requirements, changing scope/objectives, and lack of frozen requirements.

Risk mitigation strategies that emphasize the management of ambiguity and change are needed in order to control these risks. More often than not, it is impossible to pin down the exact requirements at the outset of a project, hence the popularity of various evolutionary approaches toward system development. As time progresses, the scope and requirements should become clearer. One tactic that is helpful in establishing the scope of a project is to specify what will not be included in the project. To avoid the common problem of scope creep, project managers should educate the user/customer on the impact of scope changes in terms of both project cost and schedule. To further guard against scope and requirements related risks, project managers must be willing to draw a line between functionality that is desirable versus that which is absolutely necessary.

Risk management, then, is a critical dimension of effective project management. The prospects for success on any given project can best be maximized by linking project outcomes to customer needs (40). Keil and Carmel (41) provide some insight concerning the types of links that can be established between customers and developers. In the final analysis, it is important to remember that effective project management means more than meeting schedules and budgets.

MANAGING THE PROJECT COMMUNICATION PROCESS

The communication process within a project is also of major importance. Decision makers require accurate and timely information on which to base decisions. There are three aspects that call for specific attention: accurate status reporting, provision for critical upward communication (whistleblowing) and the response to whistle blowing.

Project Status Reporting

Status reporting is subject to a combination of errors in perception and bias in reporting and receiving which is collectively called distortion. Research has found that project

managers tend to be overly optimistic in their perceptions and that executives receive information different from reality depending upon the risk level and bias applied by the project manager (42). Strategically important projects with ambiguous requirements tend to have the highest incidence of misreporting (43). Additionally, improperly applied efforts to ensure proper status reporting can create a cycle of mistrust, in which a spiral of defensive distortion and increased intrusion into the project create a climate of mutual suspicion (44).

To improve status reporting, executives should create a positive climate that focuses on project goals, while recognizing that problems are a normal part of project life and not necessarily a symptom of project manager incompetence. The climate should be such that it is expected that bad news will be accurately reported. An attitude toward problems as opportunities for project manager development rather than occurrences damaging to the manager's career should be fostered. Those working in the area of quality assurance should be positioned as knowledgeable and important contributors to project success as opposed to simply auditors seeking find problems (44). It is important to ensure that the QA/auditing staff are continuously trained in software development techniques and have a suitable career path within the organization. Additionally, by engaging in scheduled positive exchanges between QA/auditors and project teams in which they add value can increase the level of trust (44).

Tactically, executives should perform an assessment of the risk of project reporting distortion for each project. High risk projects of sufficient size should have special attention paid to them to reduce the incidence of reporting distortion. One way of approaching this is to pair an executive who is knowledgeable about software development and a good communicator with the project manager in order to provide counsel and assistance. Another approach is to ratchet up the requirements for intensive reporting and auditing although this should be done carefully to avoid issues with project morale or creation of the cycle of mistrust. In some cases it will be necessary to change the personnel either in the QA/Auditing or project management function (43).

Critical Upward Communication

This form of communication involves feedback that is critical of the performance of the project and may involve projecting poor project outcomes. Often called "whistle-blowing", this form of communication is critical to ensuring that project decision makers have correct information on which to base decisions about project direction. Yet, all too often such critical information fails to move up the project hierarchy (45), resulting in project failures that could have been prevented. Interviews of internal auditors (45) and experimental tests (46) have shown that much of this reluctance to blow the whistle is due to the perception that adverse personal consequences may result from being the "messenger" with the bad news.

Critical to fostering the upward flow of bad news is the creation of a healthy reporting environment. Negative consequences for communication of bad news should be suppressed. Management must by precept and example ensure that whistle-blowers are protected from retaliation (47). Indeed, rewards of some sort could be given to them (46). Auditing/QA organizations should be given appropriate funding and management support to do their jobs properly. They should be organizationally separate from the project organizations (45).

Tactically, there are a number of things that can be done. Increasing time pressure tends to increase willingness to report (47). Creating a sense of time urgency by using tightly drawn, but natural timeframes can encourage willingness to come forward. Staffing the project with members who have different risk propensities and perceptions will allow the manager to receive different views on the status of the project (47). Similarly, staffing the project with members of different cultural orientations can also improve willingness to report. Those who are of an individualistic culture are highly sensitive to organizational cues and will be motivated by rewards for whistle-blowing. Those of a collectivistic culture are sensitive to information asymmetry. Where information is capable of being hidden, they will tend to not report it in order to give time to the project team to solve the problems. In the opposite situation, they will tend to report bad news (48). Thus for project teams populated by those of collectivist culture, it would be advisable to use intensive monitoring to prompt upward reporting.

Response to Whistle-blowing

Here we are concerned with ensuring that when the whistle is blown, there is an appropriate organizational response to serious problems that could result in project failure. Unfortunately, decision makers frequently either don't hear, ignore, or dismiss the concerns of whistle-blowers, a phenomenon known as the deaf effect. Worse, they sometimes retaliate against the whistle-blower (45). There has not yet been much research in this area within the IS literature, however we can apply some insights from the whistle-blowing literature dealing with criminal wrongdoing. In this literature, Miceli and Near (49) proposed a model of whistle-blowing effectiveness in which they posit that characteristics of the whistle-blower, the complaint recipient and the wrongdoer, moderated by organizational support for the whistle-blower and wrongdoer, have an effect on the organization's willingness to change. They indicate that whistle-blowers who are credible and have power within the organization are more effective. Similarly, when the message is brought to an organization member with credibility and power, whistle-blowing tends to be more effective. However, if the wrongdoer is credible and powerful, that has a negative influence on the willingness to change (50). In three subsequent studies of a survey taken of whistle-blowers in the federal government, and a survey of internal auditors, they found that whistle-blowing was more effective when it was part of the whistle-blower's job, and when they used channels internal to organization for wrongdoing of a lesser magnitude and when they report to a person of high status or legitimacy in the organization. Another issue that they found was that the complaint recipients were often subject to bias and dismissed the com-

plaints inappropriately (51).

Managers should therefore attempt to make whistleblowing a role prescription for each employee by indicating to them that it is part of their job to inform their leadership and management of issues in their projects so that they can be resolved. Additionally, they should encourage taking the complaint high up in the chain of command to allow for resolution and should provide specific channels for whistleblowers to use when the chain of command is unresponsive.

BIBLIOGRAPHY

1. G. H. Anthes Killer apps, *Computerworld*, 73–74, July 7, 1997.
2. Software Magazine, Standish: Project Success Rates Improved Over 10 years, URL: http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish. Retrieved 11/21/06.
3. M. Keil Pulling the plug: Software project management and the problem of project escalation, *MIS Quart.*, **19** (4): 421–447, 1995.
4. J. Oates, UK gov holds EDS to account over crap system, The Register, 4/26/04, URL: http://www.theregister.co.uk/2004/04/26/child_support_agency_it_failure/, accessed 11/21/06.
5. J. Oates, CSA boss falls on sword over £456m IT system fiasco, The Register, 11/18/04, URL: http://www.theregister.co.uk/2004/11/18/csa_nukes_eds/, accessed 11/21/06.
6. M. Ballard, CSA: The most broken system of all? The Register, 6/30/06, URL: http://www.theregister.co.uk/2006/06/30/eds_csa/, accessed 11/21/06.
7. M. Keil J. Mann The nature and extent of IT project escalation: Results from a survey of IS audit and control professionals, *IS Audit & Control J.*, **1**: 40–48, 1997.
8. F. P. Brooks *The Mythical Man-Month: Essays on Software Engineering*, Reading, MA: Addison-Wesley, 1975.
9. T. DeMarco *Controlling Software Projects*, New York: Yourdon Press, 1982.
10. R. W. Zmud Management of large software efforts, *MIS Quart.*, **4** (2): 45–55, 1980.
11. T. Abdel-Hamid S. E. Madnick *Software Project Dynamics: An Integrated Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1991.
12. B. W. Boehm *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice-Hall, 1981.
13. T. K. Abdel-Hamid Investigating the cost/schedule trade-off in software development, *IEEE Software.*, **7** (1): 97–105, 1990.
14. T. K. Abdel-Hamid Understanding the "90% syndrome" in software project management: A simulation-based case study, *J. Syst. Softw.*, **8** (4): 319–330, 1988.
15. M. Shaw Prospects for an engineering discipline of software, *IEEE Software*, **7** (11): 15–24, 1990.
16. D. A. Taylor *Object-Oriented Technology: A Manager's Guide*, Reading, MA: Addison-Wesley, 1990.
17. R. J. Welke The shifting software development paradigm, *Data Base*, **25** (4): 9–22, 1994.
18. J. Iivari Why are CASE tools not used, Commun. *ACM*, **39** (10): 94–103, 1996.
19. C. F. Kemerer How the learning curve affects CASE tool adoption, *IEEE Software*, **9** (3): 23–28, 1992.
20. W. J. Orlikowski CASE tools as organizational change: Investigating incremental and radical changes in systems development, *MIS Quart.*, **17** (3): 309–340, 1993.
21. A. P. Young T. B. Clark *Project Management: A Practical Approach to Successful Project Performance*, Stone Mountain, GA: Young, Clark & Associates, Inc., 1994.
22. B. W. Boehm Software engineering economics, *IEEE Trans. Softw. Eng.*, **SE-10**: 4–21, 1984.
23. L. H. Putnam General empirical solution to the macro software sizing and estimating problem, *IEEE Trans. Softw. Eng.*, **SE-4**: 345–361, 1978.
24. A. J. Albrecht J. Gaffney Software function, source lines of code, and development effort prediction: A software science validation, *IEEE Trans. Softw. Eng.*, **SE-9**: 639–648, 1983.
25. H. A. Rubin Macroestimation of software development parameters: The Estimacs system, in *Proc. SOFTFAIR Conf. Softw. Develop. Tools, Tech. Alternatives*, Arlington, VA. Piscataway, NJ: IEEE Press, 1985.
26. C. F. Kemerer An empirical validation of software cost estimation models, *Commun. ACM*, **30** (5): 416–429, 1987.
27. C. F. Kemerer *Software Project Management: Readings and Cases*, Chicago, IL: Irwin, 1997.
28. The software selection project: Tips from a pro, PM Network, 28–40, September 1996.
29. M. C. Paulk *et al.* (eds.) *The Capability Maturity Model: Guidelines for Improving the Software Process*, Reading, MA: Addison-Wesley, 1994.
30. S. Alter M. Ginzberg Managing uncertainty in MIS implementation, *Sloan Manage. Rev.*, **20** (1): 23–31, 1978.
31. B. W. Boehm Software risk management: Principles and practices, *IEEE Softw.*, **8** (1): 32–41, 1991.
32. H. C. Lucas *Why Information Systems Fail*, New York: Columbia University Press, 1975.
33. R. N. Charette *Software Engineering Risk Analysis and Management*, New York: Intertext Publications, 1989.
34. F. W. McFarlan Portfolio approach to information systems, *Harvard Business Rev.*, **59** (5): 142–150, 1981.
35. H. Barki S. Rivard J. Talbot Toward an assessment of software development risk, *J. Manage. Inf. Syst.*, **10** (2): 203–225, 1993.
36. C. Jones *Assessment and Control of Software Risks, Englewood Cliffs*, NJ: Prentice-Hall, 1994.
37. R. C. Schmidt *et al.* Identifying software project risks: An international delphi study, Abstract published in *Proc. Int. Conf. Inf. Syst.*, Cleveland, OH: 1996, p. 446.
38. M. Keil, P. Cule, K. Lyytinen, R. Schmidt, A Framework for Identifying Software Project Risks, *Communications of the ACM*, **41** (11): 76–83, 1998.
39. K. Lyytinen, L. Mathiassen, J. Ropponen, Attention Shaping and Project Risk—A Categorical Analysis of Four Classical Risk Management Approaches, *Information Systems Research*, **9** (3): 233–255, 1998.
40. J. Ropponen K. Lyytinen, Components of Software Development Risk: How to Address Them? A Project manager Survey, *IEEE Transactions on Software Engineering*, **26** (2): 98–111, 2000.
41. R. Schmidt, K. Lyytinen, M. Keil, P. Cule, Identifying Software Project Risks: An International Delphi Study, *Journal of Management Inforamtion Systems*, **17** (4): 5–36, 2001.

42. S. Alter, S. Sherer, A general, but readily adaptable model of information system risk, *Communications of the Association for Information Systems*, **14**: 1–28, 2004.
43. L. Wallace, M. Keil, A. Rai, Understanding software project risk: a cluster analysis, *Information & Management*, **42**: 115–125, 2004.
44. D. Gotterbarn, S. Rogerson, Responsible Risk Analysis for Software Development: Creating the Software Development Impact Statement, *Communications of the Association for Information Systems*, **15**: 730–750.
45. J. Ropponen K. Lyytinen How software risk management can improve system development: An explorative study, *Eur. J. Inf. Syst.*, **6** (1): 41–50, March 1997.
46. B. W. Boehm R. Ross Theory-W software project management: Principles and examples, *IEEE Trans. Softw. Eng.*, **15** (7): 902–916, 1989.
47. R. H. Deane T. B. Clark A. P. Young Linking project outcomes to customer needs, *J. Inf. Syst. Manage.*, **13** (4): 1–11, 1996.
48. M. Keil E. Carmel Customer-developer links in software development, *Commun. ACM*, **38** (5): 33–44, 1995.
49. A. Snow, M. Keil, The Challenge of Accurate Software Project Status Reporting: A Two-Stage Model Incorporating Status Errors and Reporting Bias, *IEEE Trans. Engr. Mgmt*, **49** (4), November, 2002.
50. C. Iacovou, R. Thompson, H. J. Smith, Misreporting by Project Managers in Information Systems Projects: Antecedents and Impacts,Unpublished Manuscript, Wake Forest University, 2005.
51. H. J. Smith, C. Iacovou, R. Thompson, The IS Project Status Reporting Process: An Organizational Arms Race, Unpublished Manuscript, Wake Forest University, June, 2005.
52. M. Keil, D. Robey Blowing the Whistle on Troubled Software Projects, Communications of the ACM, **44** (4): 87–93, 2001.
53. H. J. Smith, M. Keil, G. DePledge Keeping Mum as the Project Goes Under: Toward an Explanatory Model, *Journal of Management Information Systems*, **18** (2): 189–227, 2001.
54. H. J. Smith, M. Keil the reluctance to report bad news on troubled software projects: a theoretical model *Info. Systems J.* **13**: 69–96, 2003.
55. B. C. Y. Tan, H. J. Smith, M. Keil, R. Montealegre Reporting Bad News About Software Projects: Impact of Organizational Climate and Information Asymmetry in an Individualistic and a Collectivistic Culture, *IEEE Trans. On Engr. Mgmt.*, **50** (1): 64–77.
56. M. Miceli, J. Near Blowing the whistle: the organizational and legal implications of companies and employees *Lexington Books*, 1992.
57. J. Near, M. Miceli Effective whistle-blowing Academy of Management Review, **20** (3): 679–708, 1995.
58. M. Miceli, J. Near What makes whistle-blowers effective? *Three field studies Human Relations*, **55** (4): 455–479, 2002.

MARK KEIL
MICHAEL J. CUELLAR
Georgia State University,
Atlanta, GA