Getting Started with Software Risk Management

Joyce A. Statz, Ph.D.
Vice-President
TeraQuest Metrics, Inc.
Post Office Box 200490
Austin, Texas 78720-0490
email: statz@teraquest.com
telephone: (512) 219-0358
fax: (512) 219-0587

Susan Tennison
Systems Analysis
Texas Department of Information Resources (DIR)
Post Office Box 13564
Austin, Texas 78711-3564
email: susan.tennison@dir.state.tx.us
telephone: (512) 475-2107
fax: (512) 475-4759

As published in *American Programmer*, *Volume 8, No. 3, 1995*

Getting Started with Software Risk Management

Joyce Statz and Susan Tennison
As published in *American Programmer*, Vol.8, No.3, March, 1995

Stories about software projects gone awry are plentiful. Note the recent problems with the baggage-handling system at the new Denver airport. Consider the California Department of Motor Vehicles' expenditure of $44M in an attempt to overhaul driver and car registration systems, only to have the whole effort stopped with no new applications complete[1]. We hear of problems in the Federal Aviation Administration's work to modernize the national air traffic control system. Many PC users are affected by the stretching schedules for Microsoft Windows 95. All of these troubled projects incur huge losses in dollars, time, and credibility for the organizations involved.

While these cases are well-publicized and the impacts are enormous, they are all too representative of problems throughout the industry that must be addressed by effective risk management. In some organizations $10 million or $100 million are at stake; in others, it may be several thousand dollars. Risk management is essential for projects that are key to an organization's success, no matter what the size.

Techniques evolving in the industry can be successfully applied in most of these cases to identify the risks, analyze the exposure of the organization, build a risk management approach, and monitor execution of the plan for handling the risk. The approach we outline here has proven useful to organizations that are facing serious risks but have never before attempted a risk management effort. In many cases, these organizations are at level 1 or 2 of the Software Engineering Institute's (SEI) Capability Maturity Model [6]. They have only a modestly defined software process and no sophisticated data from their previous history. We have found that informal techniques work best initially, allowing organizations to gather data along the way to enable more sophisticated analysis in future projects.

In this article, we will report the experiences of the information resources organizations within the agencies of the State of Texas. The General Appropriations Act enacted by the Texas legislature for the 1993-1994 biennium requires all agencies and universities to follow a careful quality assurance review process for major information technology (IT) projects. This statute defines a "major project" as any IT project identified in an agency operating plan whose development costs are over $1 million and that exhibits one or more of the following characteristics:

- Requires a year or more to reach operational status
- Involves more than one agency or government
- Materially alters work methods of agency personnel and/or delivery of services to agency clients.

In response to this mandate, staff from the Legislative Budget Board, the State Auditor's Office, and the Department of Information Resources developed a set of guidelines for agencies to follow in performing a quality assurance review [7]. While the legislature was most concerned about identifying potential problems early in the project life cycle, the team developing the guidelines expanded that focus to incorporate prevention and management techniques as well. The process the team defined is a serious project planning and risk management approach. This approach includes a careful assessment at the start of the project, followed by plans that focus on key risks, and monitoring throughout the project to ensure that standard project plans and the risk management plans are effective.

When the team reviewed all of the major IT projects detailed in the agency plans, it identified about 180 projects that could be subject to quality assurance review. After screening those projects, the team designated about 60 projects for which agencies were to complete an initial project risk analysis questionnaire. Of those projects, 30 were required to submit a project development plan and 19 were

[1] The California DMV told the State Legislature and the Department of Finance that its costs for the project were $44.3 million. But a California State Auditor's report written in August 1994 revealed that the actual project costs were at least $49.4 million. See "The Department of Motor Vehicles and the Office of Information Technology Did Not Minimize the State's Financial Risk in the Database Redevelopment Project," California State Auditor Report #94107 (Sacramento: Bureau of State Audits, August, 1994).

required to submit a post-implementation evaluation review. Of the 30 projects, 21 are being actively monitored and 3 have been required to have an independent risk analysis conducted. Because of the staffing constraints at the monitoring agencies, the projects being monitored have been prioritized and only the most risky projects have been selected.

Prior to the legislature's mandate, only a few of the state agencies with high visibility projects were doing any risk management. Most had few, if any, project management practices in place. As a result of the mandate and the new guidelines, the majority of the agencies are now aware of the need for sound project management, are getting training, and are beginning to improve their approaches.

Structuring the Approach with a Model

Among the models defined for software risk management, several capture the process needed in a way that is easy to describe to those getting started with risk management. Barry Boehm's model [2] cites the key steps needed to identify, analyze, prioritize, and create plans to manage, resolve, and monitor risks. The SEI model [4] has a similar set of activities but also includes the notion of performing tasks on a cyclical basis; that is, identifying, analyzing, planning, tracking, and controlling the risks throughout a projects life cycle.

This model, as shown in Figure 1, also promotes communication throughout the cycle --

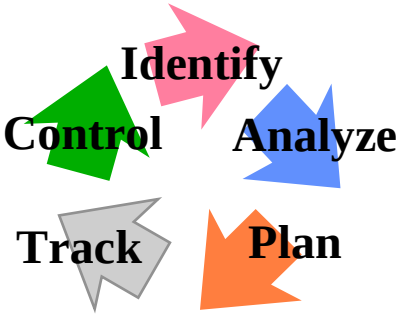

Fig. 1. SEI Risk Management Model

communication about risk information, plans, and progress -- in an environment that encourages everyone to expose risks for examination and management.

Honest communication of the risks to a project and of progress in managing them is essential. Experiences such as that of AMR Information Services, Inc., show that a lack of communication can lead to failure, even if the risks are known [5]. In this case, the company halted work on an advanced reservation system known as CONFIRM, after three and a half years of effort and an investment of more than $125 million. Max Hopper, AMR Information Services chief, commented:

> *Some people who have been part of CONFIRM management did not disclose the true status of the project in a timely manner. This has created more difficult problems -- of both business ethics and finance -- than would have existed if those people had come forward with accurate information [5].*

In the sections that follow, we will outline an initial approach to risk management that uses the SEI model as its foundation. The techniques we propose throughout are based on work being done in the industry. In each case, these can evolve to a more sophisticated version as the organization gathers more data on its performance and on relevant risks.

Identifying the Risks

Identifying risks to a project requires examining many possible sources, including:

- Business strategies and objectives
- Economic conditions
- Changing organization structures and focus
- Technological approach
- Project team personnel
- Project management abilities and expertise
- Budget and costs
- Schedule commitments
- Product performance requirements
- Development process and tools
- Customer and user interaction and needs

Elements in the list above need to be considered from the perspectives of both the software project and the overall system being developed. Key risks to the overall project arise from the interdependencies of a software component with other components such as hardware, service definition, training, and deployment timing. In a recent attempt to install a new system in Texas, for example, the deployment publicity was underway before the software project was ready. As a result, a story in the local newspaper reported as a problem something that could easily have been avoided. Similar situations arise when software is ready before the training or service procedures have been deployed. These are common problems that can be avoided with risk management.

The specific risks that can arise vary by organization. We have found several approaches that are appealing to teams analyzing risks for the first time. One approach is to use a questionnaire that asks

open-ended questions about the various areas of possible risk. The SEI's taxonomy-based risk identification approach is an example of this; it covers 13 major areas of risk to a project with about 200 questions [3].

Another approach, preferred by many and easily extensible with local interpretations, is a risk factor chart like the extract shown in Figure 2. When an organization first establishes its risk management program, it can tailor the chart to include areas of high, medium, and low risk that are observed in the organization. As the organization's risk management experience evolves, it can update the chart to reflect that learning.

The Texas agencies have used a form of this chart tailored specifically to the types of risks that arise with clients and development teams there. For example, a number of significant risks arise when clients who have never been involved with automation projects before are asked to provide sound requirements for one now under development. Also, the agencies, even more than private corporations, are vulnerable to changing politics and management; long-term projects often suffer from too much attention during one administration and too little during another.

The risk factor chart is intended to be flexible and adaptable to each agency's needs. It was initially developed to be a basic analysis tool, allowing each agency to add specific factors unique to it. The chart is also simple to use, so that the risks can be identified expeditiously and the effort can then be devoted to the management of the identified risks.

Customer/User Factors	Low Risk Characterization	Medium Risk Characterization	High Risk Characterization	Rating
User Involvement	users highly involved with project team, provide significant input	users play minor roles, moderate impact on system	minimal or no user involvement; little user input	
User Experience	users highly experienced in similar projects; have specific ideas of how needs can be met	users have experience with similar projects and have needs in mind	users have no previous experience with similar projects; unsure of how needs can be met	
User Acceptance	users accept concepts and details of system; process is in place for user approvals	users accept most of concepts and details of system; process in place for user approvals	users do not accept any concepts or design details of system	

Figure 2. Sample Risk Factor Chart

Analyzing the Risks

As teams use a factor chart, a questionnaire, or other sources of risk ideas, they build a list of candidate risks and describe as much as they can about those risks. In most projects, a team can find more risks than it can afford to mitigate completely. It takes time and money to avoid or mitigate most risks - sometimes just to monitor and observe changes in status. Thus, the risks a team can actively manage are few - usually 5 to 10, depending on the complexity of each risk.

Analysis, then, is focused on understanding the risk exposure to the project from each risk and selecting for risk management those that are most serious. For teams without an extensive history of their development activities, the lack of data allows no formal analysis of the risks, and therefore simple approaches such as Boehm's Top 10 ranking [2] are quite useful.

With a factor chart, teams can begin identifying how serious a risk might be for their project (high, medium, low). Using definitions of impact (loss) that are specific to them, they can further stratify the impact along a continuum of 1 to 10; if they can map the impact to dollar value, so much the better. However, the latter is not common. Using their perceptions, perhaps in a Wideband Delphi approach [1], they can determine the likelihood that each risk will arise. This probability of occurrence, when multiplied by the loss estimate, gives a total risk exposure for each of the risks.

Teams can organize their results in a Top 10 risk chart such as the one shown in Figure 3. Generally, teams must review such charts carefully to make sure the ordering really matches the ranking in real life. Numbers alone don't always tell the story.

ID	Risk	P	L	RE	Approach
1	Staff Availability - Too few C++ experts available	70	9	630	Contract now for more
2	Length of Development Schedule - Design schedule too tight	50	9	450	Enforce Delphi estimates
3	Requirements Stability - rapidly changing	50	7	350	Review impact, cost each time
4	Product Definition - Report function is weak	20	9	180	Review with user council
5	Product Definition - Motif I/F unacceptable to users	25	6	150	Review with user council
6	Reusable Components - library is unreliable	10	6	60	Identify second supplier
7	Use of Defined Processes - Gold plating threat	20	3	60	Inspect artifacts to preceding one
8	Supplied Components - XXX interface unstable	10	6	60	Contract with YYY
9	Response - Real time response too slow	5	6	30	Simulate and test ASAP
10	Supplied Components - OODB unreliable	5	5	25	Review with chief scientist

P - Probability of Occurrence L - Anticipated Loss RE - Risk Exposure

Figure 3. Top 10 Risk Chart

There is typically a gap in the ranking, such that teams can see which risks float to the top and are worthy of their detailed consideration as they build their risk management plans. In the example in Figure 3, while the team ought to monitor all of these issues to closure, only the top five merit serious concern and planning.

Building a Plan

For teams without extensive project documentation, the simplest way to build and maintain their plan is to include a description of their risk management approach in their software development plan and to carry the specifics about current risks and current detailed plans in living documents. The current risks being monitored and reviewed might be entered in spreadsheets like the Top 10 chart. In addition to the data shown in the Figure 3, such a chart also must show who is working on each item and when the action is to be complete. Some teams also track in this chart any contingency plans they intend to use should the risk management approach fail. In the Figure 3 example, a contingency plan for risk 3 might be that if a significant schedule change is needed due to changes in requirements (perhaps a three-month schedule addition), a user conference would be called to rate and rank alternatives and to consider an additional release of the product with these changes.

Each of the serious risks might also be documented in a one- or two-page detailed risk item tracking form like that shown in Figure 4. It is helpful to keep such material on-line where anyone on the project team can review it and where those responsible for maintaining status can do so easily. These, like many other project artifacts, are often entered into groupware products such as Lotus Notes; easy access enables the project team to communicate quickly and accurately. As the SEI risk management model indicates, such communication is key to a successful risk management program.

Tracking and Controlling to the Plan

Throughout the lifetime of the project, the team and its management need to monitor the risk management plan in tandem with the rest of the project plan. They can use the Top 10 chart as a focal point, ensuring that the current set of risks is known and under control. In addition, they must focus on the details of the important risk items. This requires using good measures of progress for each risk item. Depending on the specific risk, the measures may be number of people on staff, time for delivery of a supporting piece of software, the number of errors in reviews, the schedule impact of new or changed requirements, and so on. These measures track the progress in keeping the exposure to the risk under control, or they

identify a point (trigger value) at which the contingency plan should be activated. Figure 4 shows that the detailed tracking for a specific risk item should include both the current measure of progress (to schedule or level of effort) and the trigger value for a contingency plan (amount of schedule slip or effort above what was planned).

In addition to monitoring the risks on its current list, the team needs to be alert to new risks that enter its environment as the project proceeds. When the team detects a new risk, it should integrate that risk into its Top 10 list and its planning mechanisms. Regular project reviews should include an examination of the current risk list and how it compared to previous lists to ensure that the team is taking action in a timely fashion to ensure project success. For a project that is more than a year long, a full reassessment of risks is useful at key milestones, such as completion of the initial design phase. The risk lists must be actively used and reviewed; these are not mere action item lists for historical purposes.

Texas agencies are seeing the benefits of their risk tracking and management. In one case, an agency put off an update of database support software because review of other organizations' experience showed that using this software would jeopardize the performance of several key applications. Contrast this with the experience of the same agency before it implemented risk management practices: the agency successfully pilot tested a mainframe application in one configuration but had to pull it back after installing it to other mainframes which had slightly different software release levels from the pilot system.

Risk ID: 10/94-02	Factor Type: Schedule	Report Date: 1/9/95
Probability: 50	Loss: 9	Risk Exposure: 450
Description of Risk: The design schedule is perceived by the development team to be too tight. Team members fear having to drop features or deal inadequately with the issues in order to meet the schedule.		
Current Management Plan: Delphi estimates have been made for each work breakdown structure element during the design phase. These estimates need to be monitored, and when a change is needed, the Delphi estimates must be updated and new projections for completion dates set. Date Started: 11/15/94 Date to Complete: 3/15/95		
Status/Measures: First three items completed just slightly over budgeted time.		
Contingency Plan: If the estimates show that the phase will exceed its current design phase goal by more than 10 calendar days and/or more than 15 person-days of effort, set up a feature tradeoff meeting with marketing to decide whether to modify the content or the schedule. Trigger Value: design completion date is beyond 3/29/95 or effort is more than 135 person-days for design		

Figure 4. Sample Risk Item Tracking Form

Learning from the Experience

Once a project is complete, the team needs to conduct a post-project analysis to gather its lessons learned. Among the items considered with respect to risk management should be:

- Were identified risks successfully managed and avoided?
- Were any risks addressed with more resources than justified?
- Were any new risks encountered that were not anticipated?

Positive responses to the first question indicate that the team needs to continue using current methods. If the team used more resources than necessary, it must make an adjustment in its planning in future projects. Identification of new risks may lead the organization to modify its standard risk factor charts so that these risks are considered in future projects.

The lessons learned must be captured in the artifacts that support the organization's risk management plan. Tailored risk factor charts provide an excellent tool for this. Well-defined procedures for conducting risk analysis ensure that projects can repeat their success with rating and ranking the risks to which they dedicate resources.

In addition to answering the questions above, the post-project analysis should provide data for the organization's growing history of performance. Cost, schedule, size, and effort data are the fundamentals that will support increasingly more accurate project planning and risk analysis over time. As project teams improve their processes and project management, these data become more useful and a more reliable predictor of success.

Part of the Texas agencies' quality assurance process is a post-implementation evaluation review. Such a review currently focuses primarily on how well the implementation met the goals of the project. It is being modified to include an analysis of the risk management and project management lessons learned, both positive and negative. Agencies are just beginning to use the post-implementation evaluation review, so feedback is minimal at this point.

Why Bother with Risk Management?

Software development projects are becoming more complex, with increasing demand for shorter development cycles from the marketplace, which screams for quick delivery. As teams build their plans to meet customer needs, they must consider the risk factors that can cause failure and build management plans to address those risks. When teams don't manage their risks, high-visibility projects are vulnerable to the bad press the Denver airport system has received. For smaller projects of less visibility, it may be the health of the organization that is at risk. In each case, the project team needs to judge the impact of not creating a risk analysis and risk management plan; this is the first risk element to consider.

While the first attempt to follow the process we've outlined will take an organization more time than its current project planning, this time will decrease as the organization learns what risks are most likely to occur in its work. This learning can be captured nicely in the tailored factor chart, which should be updated each time a project completes its post-project analysis or any time a team finds significant risks during its analysis phases. As the base of knowledge expands, the project team will be able to detect more of the risks early, plan ways to avoid them, and see the benefits outweigh the cost of risk management.

About the Authors

Joyce Statz is Vice President of TeraQuest Metrics, Inc., an Austin, Texas-based company that provides training, process development consulting, and measurement services to organizations improving their software processes. She has 15 years of experience in the design, implementation, and management of leading edge software systems at Texas Instruments.

Dr. Statz taught computer science at Bowling Green State University for five years, and she is a founder and board member of the Software Quality Institute of the University of Texas. She is the curriculum coordinator and an instructor for the Software Project Management Certificate Program. She is involved with a number of professional organizations and is Program Chair for the upcoming Fifth International Conference on Software Quality.

Dr. Statz may be reached at TeraQuest Metrics, Inc., P.O. Box 200490, Austin, TX 78720-0490 (512/219-0358; fax (512/219-0587; Internet: statz@acm.org)

Susan Tennison is a systems analyst for the Texas Department of Information Resources (DIR), an agency responsible for the oversight of all state agencies' planning for information resources. DIR also provides statewide strategic planning for information resources,

statewide telecommunications planning and services, cooperative purchasing for information resources, and serves as a service bureau for other agencies.

Ms. Tennison has worked for state government in Texas for 22 years as a programmer, analyst, project leader, and project manager. She has been a director of data processing at a small agency and manager of a group of analysts at a large agency. She currently works with the quality assurance review process by serving as a resource to the Quality Assurance Team and by monitoring several projects.

Ms. Tennison can be reached at the Department of Information Resources, P.O. Box 13564, Austin, TX 78711-3564; (512/475-2107; fax 512/475-4759; Internet: susan.tennison@dir.state.tx.us)

Next Steps

A sample risk factor chart is available from TeraQuest Metrics, Inc. for anyone who would like to use it as a starting point in defining project risks and indicators. This sample has about 60 factors, clustered in 11 areas of risk, with descriptions of high, medium, and low characterizations of each factor. The chart is an Excel spreadsheet, available on diskette (specify PC or Macintosh) for a nominal fee of $5 to cover duplication and mailing. If you are interested, contact Joyce Statz at TeraQuest Metrics, Inc., P.O. Box 200490, Austin, TX 78720-0490 (Internet: statz@acm.org).

A hard copy of the "Guidelines for Quality Assurance Review" for Texas state agencies is available from the Texas Department of Information Resources at no cost. If you are interested, send your name and address to Department of Information Resources, P.O. Box 13564, Austin, Texas 78711-3564.

The guidelines are also available on line through the Internet by means of the Texas Information Highway. To access the Texas Information Highway, use the following:

via Telnet: bbs.dir.state.texas.us
(enter DIRBBA as user name)

via Gopher: info.state.texas.us
port 70

via World Wide Web:
http://www.state.texas.us
After reaching the home page, select "Department of Information Resources," then select "Statutory Information," then "Instructions & Guidelines," then "Quality Assurance Guidelines."

If you have questions about the Texas Information Highway, call 512/475-4715.

Locating This Article

This article was published in the March, 1995 issue of *American Programmer*, Volume 8, Number 3, pages 23-30. The issue focuses on risk management and includes four other articles.

References

[1] Boehm, Barry. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[2] Boehm, Barry. "Software Risk Management: Principles and Practices," *IEEE Software*, vol.8, no. 1 (January, 1991), p. 32-41.

[3] Carr, Marvin J., Suresh L. Konda, Ira Monarch, F. Carol Ulrich, Clay F. Walker. *Taxonomy-Based Risk Identification*, CMU/SEI-93-TR-006. Pittsburgh, PA: Software Engineering Institute, June, 1993.

[4] Chittister, Clyde, Robert Kirkpatrick, and Roger Van Scoy. "Risk Management in Practice," *SEI Technical Review*, Pittsburgh, PA: Software Engineering Institute, 1993.

[5] Oz, Effy. "When Professional Standards are Lax, the CONFIRM Failure and its Lessons," *Communications of the ACM*, vol. 37, no. 10 (October, 1994), p. 29-36.

[6] Paulk, Mark C., Bill Curtis, Mary Beth Chrissis, and Charles V. Weber. *Capability Maturity Model for Software, Version 1.1*, CMU/SEI-93-TR-24. Pittsburgh, PA: Software Engineering Institute, February, 1993.

[7] State of Texas Legislative Budget Office, Department of Information Resources, Office of the State Auditor. "Guidelines for Quality Assurance Review." Austin, Texas: Department of Information Resources. February, 1994.