Software Tech News

Your Source for Information in Software Engineering Technology.

Volume 2 Number 2

STN 2-2 Topic:
Risk Management

In This Issue **:**

Read additional Risk Management articles at:

www.dacs.dtic.mil/awareness/newsletters/listing.shtml

Risk Management Map

by Elaine Hall, Ph. D. - Level 6 Software

In software-intensive product development, relatively risk-free opportunities are long gone. Software risk is actually on the rise because it increases as system complexity increases. Managing risk is necessary when software risk prevents us from achieving our goals and objectives. People inherit risk at work by assuming one (or more) of the project roles. Unfortunately, people do not inherit the ability needed to manage the risk. The ability to manage risk is a developmental process that is learned through education and experience.

Managing risk is a lot like playing golf. Known risks on a golf course include sand traps and water hazards. We can recognize a golfer's skill level by how the person manages these risks, for example:

Novice: In a round of golf, novices have no idea how many balls they will lose in the water.

Beginner: Around the water hazard, beginners play their less expensive balls. They would rather lose old balls than new ones.

Intermediate: Because they know their capability with each iron, intermediates often switch clubs and lay up before they attempt to cross the water.

Advanced: Those who are advanced determine the length of the water hazard and select the appropriate club. They may push the limits of their capability or play it safe, depending on the margin needed to win.

Expert: Experts do not see the water as an obstacle. When they take aim, they account for both wind direction and velocity. They visualize the ball landing in the best position for their next shot.

Continued on page 9

DoD Data & Analysis Center for Software
http://www.dacs.dtic.mil

Risk Management: Finally Coming of Age

By Robert N. Charette, Ph. D. - ITABHI Corporation

Once More, with Vigor…

In 1969, the Deputy Secretary of Defense directed the secretaries of the armed forces to identify areas of high technical risk, perform formal risk analysis, and include explicit consideration of risk assessment, reduction and avoidance in managing weapon systems acquisition.

Unfortunately, with few exceptions over the past 25 years, active risk management has been more of an afterthought than a primary factor in Department of Defense (DoD) decision making. Instead, the most prevalent means for managing risks has been the "fix-on-failure" problem control approach – i.e., waiting for risks to occur before taking remedial action.

Only in the last four or five years has the application of formal risk management taken solid root in the management of DoD programs. While the causes vary, the increasing use of risk management can be associated with the ever increasing costs (in terms of financial, political and defense posture) of DoD program failure, the cutbacks in available resources, the proven success of risk management on DoD and commercial programs, and recent Congressional mandates.

Program costs have skyrocketed over the past decade. Major program costs regularly reach into the tens of billions of dollars. Even minor schedule slips can cost hundreds of millions of dollars. In an era of tight resources, being even slightly wounded makes a program vulnerable. As one DoD official succinctly put it,

> *"Be successful or join the wounded. And we are now shooting the wounded."*

As program costs have risen, Congress has been equally less amenable towards funding "make-up" programs for ones that fail. Often program failure means concept failure as well, with the result that it is extremely difficult to receive funding approval for a similar concept, even if the original program requirement still exists. The Navy's A-12 aircraft program is a prime example.

While the cost of failure has highlighted the need for active risk management, a stronger factor has been the success reported by programs applying formal risk management. For example, Hughes Aircraft (now part of Raytheon) aggressively used risk management on the highly complex, four-and-a-half year, 750-person Peace Shield air defense system [1]. Their risk management approach, which was used as a problem-preemption strategy, was credited with helping deliver Peace Shield 10% ahead of schedule and significantly below projected cost. Achieving success, which many observers believed was impossible at the time of Peace Shield's start, sparked the company into institutionalizing risk management for use on its other major programs.

Other DoD programs such as the V-22, E-6, F-18 and F-22 are also crediting risk management with significantly enhancing their program management's capabilities. In fact, data gathered from across all industry sectors by the Project Management Institute (PMI) demonstrates conclusively that project success is strongly correlated with the practice of risk management [2]. Rockwell Collins, for example, has recently determined that there is at least a 17% difference in the Cost Performance Index (CPI) between its projects that perform risk management and those that do not.

Continued on page 3

Continued from page 2

While cost of failure and the sweet smell of success have spurred the use of risk management, Congress has added its weight by the recent passage of the Clinger-Cohen act. This new law requires all but a few government programs and projects to perform formal risk assessments and to report those risks. For programs that decide to ignore Clinger-Cohen and later have a program hiccup, rest assured that they will be counted among the wounded – and we know what is happening to the wounded.

No Surprises

Providing new found insights to a decision-maker so that he or she can make informed decisions is the primary objective of risk management. Another way of putting it is risk management aims to keep the boss from being surprised.

However, to keep the boss from being surprised requires a set of comprehensive, coordinated and complementary management of risk and risk management practices. The management of risk approach addresses risk in a top-down, granular, periodic fashion, and concerns "command-level decisions", such as whether a program should be initiated, should it receive funding, has it passed a major milestone, etc. Its primary focus is on understanding the risks (and opportunities) that exist before plans are defined and/or put into operation. Risk management practice, on the other hand, concentrates on performing bottom-up, detailed, continuous assessment of risk (and again opportunity), concerning itself with addressing the day-to-day operational risks that a program faces. Together they provide a 360° 3-D view of the risk that might confront a program.

A management of risk approach is similar to what is performed during aircraft strike planning, with mission planners running through different attack scenarios, trying to pick the best attack routes having the fewest threats, defining the way points, etc. Risk management is similar to what a pilot does once the strike plan has been approved and the mission is launched. The pilot constantly checks instruments, gets updates from AWACS, checks for items the mission planners missed or couldn't foresee, etc., i.e., updates his or her situational awareness, while taking corrective actions to ensure the strike can be successful.

Both management of risk and risk management approaches follow a two-stage, repeatable and iterative process of assessment (i.e., the identification, estimation and evaluation of the risks confronting a program) and management (i.e., the planning for, monitoring of, and controlling of the means to eliminate or reduce the likelihood or consequences of the risks discovered). Both take a holistic or systems view of the risks likely to be encountered (from their own unique perspectives), and likewise take a systems view on how they should be mitigated. Both are done continually over the life of a program, from its initiation to its retirement. Both should be able to be paid for from existing or minimal increases (3-7%) in program administrative costs. Finally, both should be performed not only by government program officials, but also by contractors working in conjunction and cooperation with government. Open communication of risk is key to its successful management.

To ensure a complete understanding of the risks to a program, both management of risk and risk management practices need to be integrated into a program's measurement processes. Once linked, a program can understand its past trends, see its future trends, and be able to predict its future trends with some level of confidence. Work being done under the auspices of the Practical Software Measurement (PSM) effort is concentrating on how to make the linkage between risk and measurement easier [3].

Continued on page 4

Continued from page 3

Reckon, then Risk

It would be a mistake to view risk management as yet another impractical idea mandated by nameless bureaucrats that serves only to keep your project or program from achieving its objective – quite the opposite. By applying good risk management practice, your program will be able to not only take on more risk but also exploit opportunities that now have to be passed by. Further, time to act, rather than react, will be gained, along with a greater number of alternatives to choose from when problems are eventually encountered. As a result, the program will develop a risk taking ethic, one with a bias toward informed action.

Of course, risk management is not a panacea. It will not turn bad situations automatically into good ones, make the operating environment suddenly pleasing nor ensure every high risk can be eliminated or avoided. Further, even if risks are identified and mitigation plans developed, no guarantee exists that proper

actions will be taken in a timely manner. Risk management requires a belief in and commitment to the process of risk management by senior management. The current Y2K *problem* is a case study of what happens when *risk* warnings are ignored.

However, when practiced well, risk management can provide that extra edge needed to make a program successful, or at least keep that program from joining the wounded.

About the Author

Dr. Robert N. Charette is the President of ITABHI Corporation, an international risk management consultancy company. Dr. Charette is past Chairperson of both the SEI Risk Advisory Board and NSIA Software Committee, is a founding member of the PMI Risk SIG, and risk management advisor to the PSM project. Dr. Charette has written dozens of papers and several books on risk management, including *Software Engineering Risk Analysis and Management*, *Applications Strategies for Risk Analysis* and *An Introduction to the Management of Risk*.

Contact Information:

Dr. Robert N. Charette
ITABHI Corporation
11609 Stonewall Jackson Drive
Spotsylvania, VA 22553-4668
(540) 972-8150
Charette@erols.com

References

1. Sutherland, Chuck. “Peace Shield Risk Management.” *Proceedings of the 5th SEI Conference on Software Risk Management*. Software Engineering Institute, Pittsburgh, Pennsylvania, 1997.

2. Ibbs, C. Williams and Young-Hoon Kwak. *The Benefits of Project Management: Financial and Organizational Rewards to Corporations*. PMI Publications, Upper Darby, Pennsylvania, 1997.

3. McGarry, John et al. PSM. *Practical Software Measurement: A Foundation for Objective Project Management*. Washington, DC, OUSD (A&T) and the JLC Joint Group on Systems Engineering, 17 April 1998. See also www.psmsc.com.

Assessing Project Risk

By Shari Lawrence Pfleeger - University of Maryland

Adapted from *Software Engineering: Theory and Practice, by Shari Lawrence Pfleeger* with permission from Prentice-Hall.

Introduction

Many software project managers take steps to ensure that their projects are done on time and within effort and cost constraints. However, project management involves far more than tracking effort and schedule. Managers must determine whether any unwelcome events may occur during development or maintenance, and make plans to avoid these events or, if they are inevitable, minimize their negative consequences. A **risk** is an unwanted event that has negative consequences. Project managers must engage in **risk management** to understand and control the risks on their projects.

What is a Risk?

Many events occur during software development. We distinguish risks from other project events by looking for three things:[1]

1. **A loss associated with the event.** The event must create a situation where something negative happens to the project: loss of; time, quality, money, control, understanding, and so on. For example, if requirements change dramatically after the design is done, then the project can suffer from loss of control and understanding if the new requirements are for functions or features with which the design team is unfamiliar. A radical change in requirements is likely to lead to losses of time and money if the design is not flexible enough to be changed quickly and easily. The loss associated with a risk is called the **risk impact**.
2. **The likelihood that the event will occur.** We must have some idea of the probability that the event will occur. For example, suppose a project is being developed on one machine and will be ported to another when the system is fully tested. If the second machine is a new model to be delivered by the vendor, we must estimate the likelihood that it will not be ready on time. The likelihood of the risk, measured from 0 (impossible) to 1 (certainty) is called the **risk probability**. When the risk probability is 1, then the risk is called a **problem**, since it is certain to happen.
3. **The degree to which we can change the outcome.** For each risk, we must determine what we can do to minimize or avoid the impact of the event. **Risk control** involves a set of actions taken to reduce or eliminate a risk. For example, if the requirements may change after design, we can minimize the impact of the change by creating a flexible design. If the second machine is not ready when the software is tested, we may be able to identify other models or brands that have the same functionality and performance and can run our new software until the new model is delivered.

We can quantify the effects of the risks we identify by multiplying the risk impact by the risk probability, to yield the **risk exposure**. For example, if the likelihood that the requirements will change after design is .3, and the cost to redesign to new requirements is $50,000, then the risk exposure is $15,000. Clearly, the risk probability can change over time, as can the impact, so part of a project manager's job is to track these values over time, and plan for the events accordingly.

There are two major sources of risk: generic risks and project-specific risks. **Generic risks** are those common to all software projects, such as misunderstanding the requirements, losing key personnel, or allowing insufficient time for testing.

Project-specific risks are threats that result from the particular vulnerabilities of the given project. For example, a vendor may be promising network software by a particular date, but there is some risk that the network software will not be ready on time.

Continued on page 6

Continued from page 5

Risk Management Activities

Risk management involves several important steps, each of which is illustrated in Figure 1. First, you assess the risks on your project, so that you understand what may occur during the course of development or maintenance. The assessment consists of three activities: identifying the risks, analyzing them, and assigning priorities to each of them. To identify them, you may use many different techniques.

If the system you are building is similar in some way to a system you have built before, you may have a checklist of problems that may occur; you can review the checklist to determine if your new project is likely to be subject to the risks listed. For systems that are new in some way, you may augment the checklist with an analysis of each of the activities in the development cycle. By decomposing the process into small pieces, you may be able to anticipate problems that may arise. For example, you may decide that there is a risk of your chief designer's leaving during the design process. Similarly, you may analyze the assumptions or decisions you are making about how the project will be done, who will do it, and with what resources. Then, each assumption is assessed to determine the risks involved.

Finally, you analyze the risks you have identified, so that you can understand as much as possible about when, why and where they might occur. There are many techniques you can use to enhance your understanding, including system dynamics models, cost models, performance models, network analysis, and more.

Now that you have itemized all risks, you must use your understanding to assign priorities to the risks. A priority scheme enables you to devote your limited resources only to the most threatening risks. Usually, priorities are based on the risk exposure, which takes into account not only likely impact but also the probability of occurrence.

The risk exposure is computed from the risk impact and the risk probability, so you must estimate each of these risk aspects.

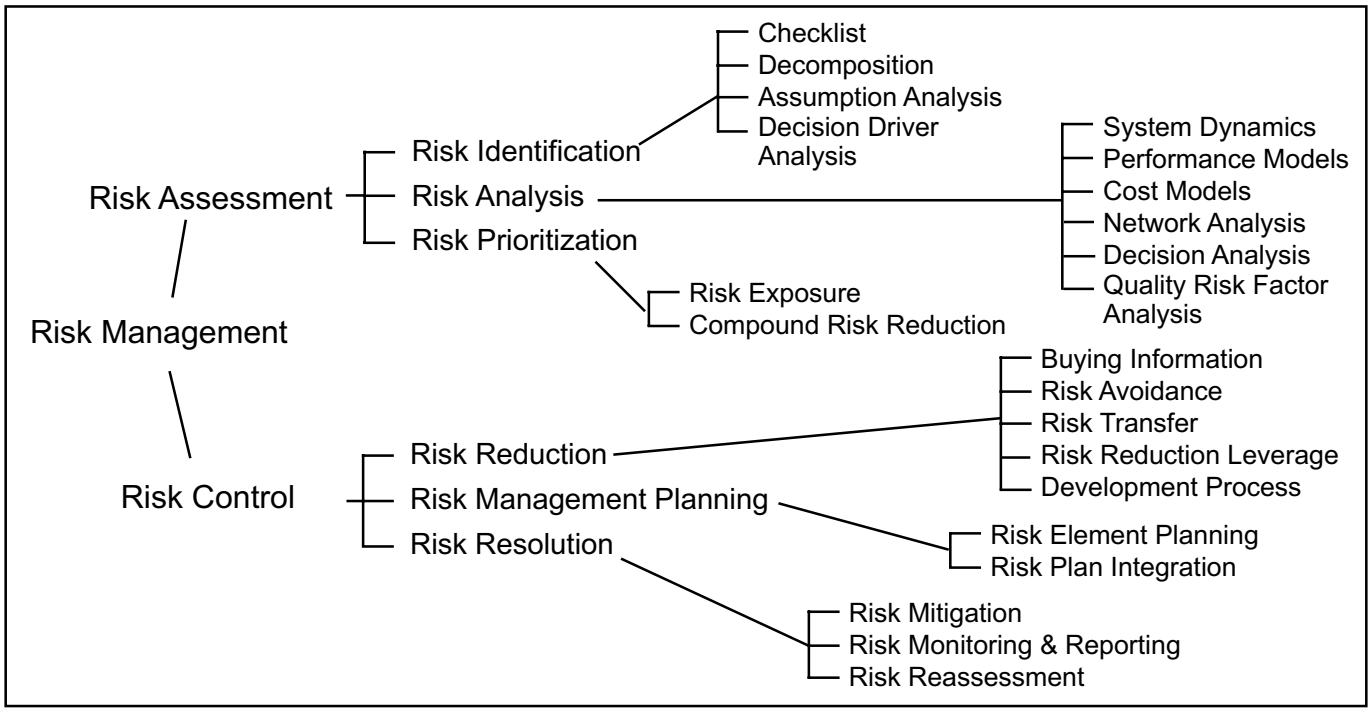


Figure 1. Steps in Risk Management

Continued on page 7

Continued from page 6

To see how the quantification is done, consider the analysis depicted in Figure 2. Suppose you have analyzed the system development process, and you know you are working under tight deadlines for delivery. You will be building the system in a series of releases, where each release has more functionality than the one that preceded it. Because the system is designed so that functions are relatively independent, you are considering testing only the new functions for a release, and assuming that the existing functions still work as they did before. Thus, you may decide that there are risks associated with not performing regression testing: the assurance that existing functionality still works correctly.

For each possible outcome, you estimate two quantities: the probability of an unwanted outcome, P(UO), and the loss associated with the unwanted outcome, L(UO). For instance, there are three possible consequences of performing regression testing: finding a critical fault if one exists, not finding the critical fault (even though it exists), or deciding (correctly) that there is no critical fault. As the figure illustrates, we have estimated the probability of the first case to be 0.75, of the second to be 0.05, and of the third to be 0.20. The likelihood of an unwanted outcome is estimated to be $0.5 million if a critical fault is found, so that the risk exposure is $0.375 million.

Similarly, we calculate the risk exposure for the other branches of this decision tree, and we find that our risk exposure if we perform regression testing is almost $2 million. However, the same kind of analysis shows us that the risk exposure if we do not perform regression testing is almost $17 million. Thus, we say (loosely) that more is at risk if we do not perform regression testing.

Risk exposure helps us to list the risks in priority order, with the risks of most concern given the highest priority. Next, we must take steps to control the risks. The notion of control acknowledges that we may not be able to eliminate all risks.

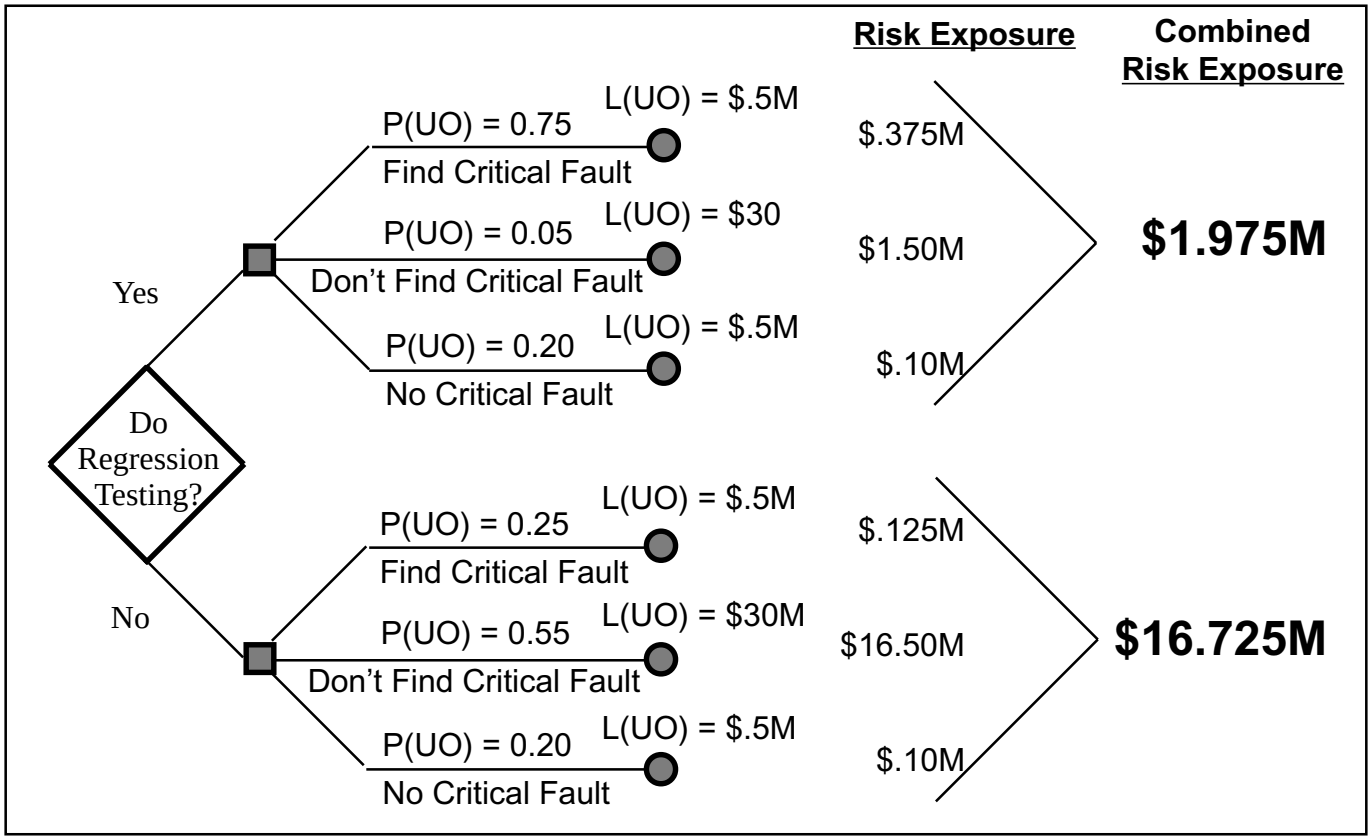


Figure 2. Example of Risk Exposure Calculation

Continued on page 8

Continued from page 7

Instead, we may be able to minimize the risk, or mitigate it by taking action to handle the unwanted outcome in an acceptable way. Therefore, risk control involves risk reduction, risk planning, and risk resolution.

There are three strategies for risk reduction:

1. Avoiding the risk, by changing requirements for performance or functionality
2. Transferring the risk, by allocating risks to other systems or by buying insurance to cover any financial loss should the risk become a reality
3. Assuming the risk, by accepting it and controlling it with the project's resources

To aid decision-making about risk reduction, we must take into account the cost of reducing the risk. We call **risk leverage** the difference in risk exposure divided by the cost of reducing the risk. In other words, risk reduction leverage is (risk exposure before reduction – risk exposure after reduction)/(cost of risk reduction).

If the leverage value is not high enough to justify the action, then we can look for other, less costly or more effective reduction techniques. In some cases, we can choose a development process to help reduce the risk. For example, prototyping can improve understanding of the requirements and design, so selecting a prototyping process can reduce many project risks.

It is useful to record your decisions in a risk management plan, so that both customer and development team can review how problems are to be avoided, as well as how they are to be handled should they arise. Then, we should monitor the project as development progresses, periodically reevaluating the risks, their probability, and their likely impact.

Table 1. summarizes what Boehm has identified as the top ten risk items[2]. When assessing risk on your own project, you can begin with this list, and determine if any of the items might apply. Then, you can expand your list, based on past history and your understanding of the project's goals and limitations. Boehm identifies ten risk items, and recommends risk management techniques to address each of them.

Contact Information

Shari Lawrence Pfleeger
4519 Davenport Street NW
Washington, DC 20016-4415
(301) 405-2707
Fax: (301) 405-3691
s.pfleeger@ieee.org

Table 1: Boehm's top ten risk items

1. Personnel Shortfalls:	Staffing with top talent; job matching; team-building; morale-building; cross-training; prescheduling key people.
2. Unrealistic Schedules and Budgets:	Detailed, multisource cost and schedule estimation; design to cost; incremental development; software reuse; requirements scrubbing.
3. Developing the wrong software functions:	Organizational analysis; mission analysis; operational concept formulation; user surveys; prototyping; early users' manuals.
4. Developing the wrong user interface:	Prototyping; scenarios; task analysis.
5. Gold-plating. Requirements scrubbing:	prototyping; cost-benefit analysis; design to cost.
6. Continuing stream of requirements changes:	High change threshold; information-hiding; incremental development (defer changes to later increments).
7. Shortfalls in externally-performed tasks:	Reference-checking; pre-award audits; award-fee contracts; competitive design or prototyping; team-building.
8. Shortfalls in externally-furnished components:	Benchmarking; inspections; reference checking; compatibility analysis.
9. Real-time performance shortfalls:	Simulation; benchmarking; modeling; prototyping; instrumentation; tuning.
10. Straining computer science capabilities:	Technical analysis; cost-benefit analysis; prototyping; reference checking.

References

1. Rook, Paul, "Risk Management for Software Development", ESCOM Tutorial, 24 March 1993.
2. Boehm, Barry W., "Software Risk Management: Principles and Practices", *IEEE Software* 8(1), pp. 32-41, January 1991.

Continued from page 1

No two golf courses or software projects are ever the same. For this reason, software engineers, like golfers, must develop general skills for managing risk through practice. To progress from risk management novice to expert, you can use the Risk Management Map described in the book *Managing Risk: Methods for Software Systems Development* [1]. Developed from empirical data on software-intensive projects (1992-1997), the Risk Management Map charts the course for increasing the ability to manage software risk. As shown in Figure 1, the map contains five evolutionary stages: Problem, Mitigation, Prevention, Anticipation, and Opportunity.

Risk Management Map

The Risk Management Map is a practical guide to understanding the path to increasing your ability to manage risk by transitions through five stages. At each stage, a vision provides the direction for your journey.

Stage 1: Problem

The problem stage describes the circumstances when risk identification is not seen as positive. It is characterized by a lack of communication, which causes a subsequent lack of coordination. People are too busy solving problems to think about the future. Risks are not addressed until they become problems, because either management was not aware of the risk or inaccurately estimated the risk's probability of occurrence. Since management reaction to hearing risks is typically to shoot the messenger, most people will not deliver bad news. Crisis management is used to address existing problems. People learn that fire fighting can be exciting, but it causes burnout.

Stage 2: Mitigation

The second stage, the mitigation stage, details the shift from crisis management to risk management. Management now incorporates risk management technology by asking, "What can go wrong?" and "What are the consequences?" This stage is characterized by an introduction to risk concepts. That is, people become aware of risks but do not systematically confront them. Since their knowledge and experience using risk management are limited, they may be unsure of how to communicate risks. In this stage, managers use risk management to reduce the probability and consequence of critical risks by implementing a contingency plan if the original plan fails.

Stage 3: Prevention

The prevention stage discusses the shift from risk management

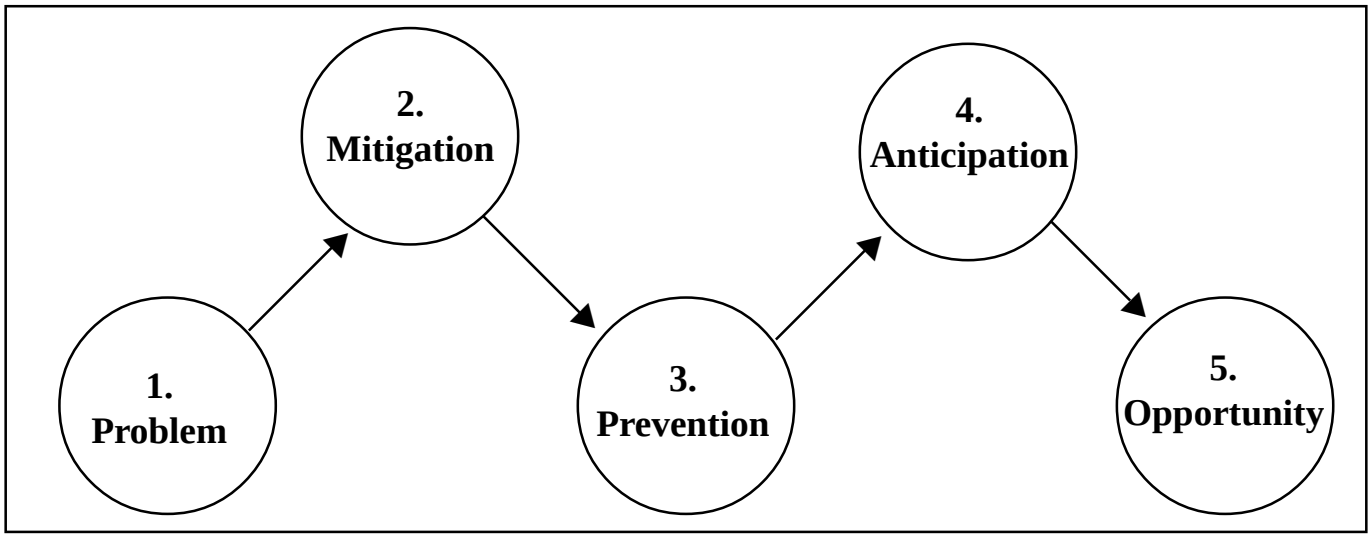


Figure 1. Risk Management Map

Continued on page 10

Continued from page 9

viewed as a manager's activity to a team activity. This third stage is a transitional one where the approach changes from avoidance of risk symptoms to identification and elimination of the root cause of risk. It is characterized by team and occasional customer involvement, as managers understand that risk management is a dynamic process that cannot be performed in isolation. Instead of focusing on cost and schedule risk (a management perspective, usually a symptom of technical risk), a focus on technical risks leads to a discovery of the source of risk. Prevention is a turning point from a reactive to a more proactive approach to risk management. Most people are experienced and comfortable in risk identification but are unsure how to quantify risks.

Stage 4: Anticipation

The fourth stage, the anticipation stage, describes the shift from subjective to quantitative risk management through the use of measures to anticipate predictable risks. It is characterized by the use of metrics to anticipate failures and predict future events. The project team and customer use risk management to quantify risks with reasonable accuracy to focus on the right priorities. A proactive approach to attacking risks and assessing alternatives is used. Alternatives are easier to compare using a quantitative approach. By this early warning system, anticipated problems are avoided through corrective action.

Stage 5: Opportunity

The final stage, the opportunity stage, is a positive vision of risk management that is used to innovate and shape the future. Potentially the most powerful paradigm shift is in perceiving risks as chances to save money and do better than planned. Risk, like quality, is everyone's responsibility. Professional attitudes of engineering excellence allow for open communication and individual contribution. We admit that there are things that we do not know and allow for their existence using a best-case, worst-case scenario. People understand there is an opportunity cost associated with every choice, and knowing these trade-offs improves their decision-making ability. In the hands of the many, a positive expectation of using risk management to exceed established goals becomes a powerful weapon.

The structure of the Risk Management Map is similar to the arrow of a state-transition diagram that takes you to another node. As shown in Figure 2, a "stage" transition takes you to a higher level of risk management capability. Stages describe incremental enhancements in the capability to manage risk. To achieve the next stage of development, you need the vision, goals, and strategy provided by the Risk Management Map.

Map Architecture:

The underlying structure of the Risk Management Map supports the transition between stages. *Stages* provide incremental enhancements in the capability to manage risk. *Vision* guides the way to the next stage. *Goals* are

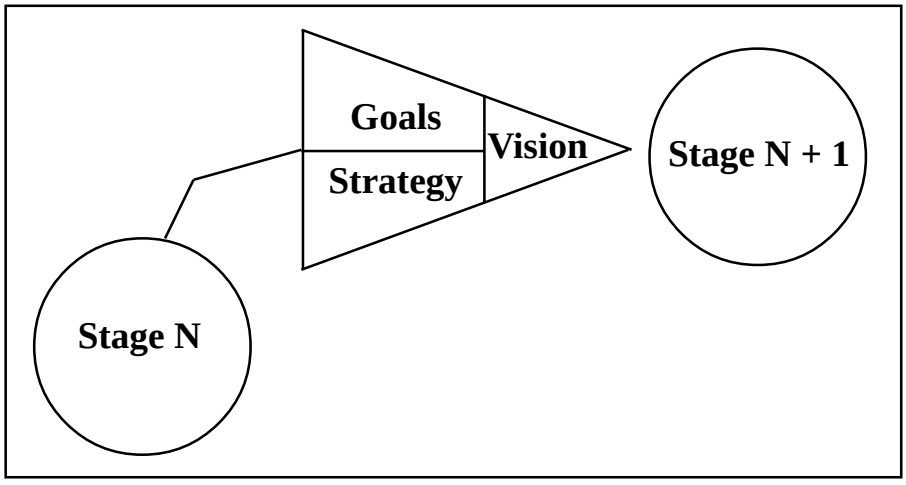


Figure 2. Map Architecture

Continued on page 11

accomplished to achieve the vision. *Strategy* is the activity that supports goal attainment.

Vision: Vision is an ideal state of the practice that guides the journey. It acts as a driving force that provides the motivation required to continue our effort. For each stage, a vision provides direction and guides the way to the next stage. The Risk Management Map paints a picture of five progressive stages through visions of competitive advantage, customer satisfaction, increased predictability, and maximized opportunities.

Goals: To achieve the vision, you must accomplish goals. The Risk Management Map provides the goals to bring each vision into reality. It is based on evolving the major factors that affect risk management capability: people, process, infrastructure and implementation. [Note: Chapters in *Managing Risk* describe each of the following map goals.]

People are a critical factor in communicating the issues, concerns, and uncertainties in their work that translate to risk. Goals for the people are Stage 1: Problem, Stage 2: Mitigation, Stage 3: Prevention, Stage 4: Anticipation, and Stage 5: Opportunity.

Process is a major factor because it describes the steps to predictable risk management results. Process goals are identify risk, analyze risk, plan risk, track risk, and resolve risk.

Infrastructure is a major factor because it establishes the culture that supports use of risk management. Infrastructure goals are develop the policy, define standard process, train risk technology, verify compliance, and improve practice.

Implementation is a major factor because it assigns to the project the responsibility and authority to execute the plan. Implementation goals are establish the initiative, develop the plan, tailor the standard process, assess risk, and control risk.

Strategy: Strategy is the activity that supports goal attainment. The Risk Management Map provides a strategy to realize each goal. It specifies an approach to achieve goals and yields activity to check for results. If the results do not support goal attainment, you can make tactical adjustments. [Note: Subsections of each chapter in *Managing Risk* outline the strategy the map provides and arrange the required activities in their proper order.]

Transformation from risk management novice to expert is a process of gradual growth and change. The Risk Management Map provides a practical focus needed for this evolution. The ability to manage risk is a "use-it-or-lose-it" proposition. You must apply your ability to manage risk to achieve the control, higher return, or opportunities that you envision. If you develop the skill to manage risk but choose not to use this ability, you will lose your competitive edge. Knowledge without action is insufficient to derive the benefits of risk management.

About the Author

Elaine M. Hall is founder of Level 6 Software and author of *Managing Risk: Methods for Software Systems Development* (©1998 www.awl.com/cseng/titles/0-201-25592-8/ - Addison Wesley SEI Series in Software Engineering). Dr. Hall is chair of the International Council on Systems Engineering Risk Management Working Group.

Contact Information

Dr. Elaine M. Hall, Ph.D.
Level 6 Software
530 Franklyn Ave.
Indialantic, FL 32903
Phone/Fax: (407) 728-RISK
www.level6software.com

Footnotes

1. Hall E. *Managing Risk: Methods for Software Systems Development*. Reading, MA: Addison Wesley, 1998.2.

Software Acquisition Risk - A Perspective

By Martin L. Shooman - SAIC & Polytechnic University and Ernest Lofgren - SAIC

Introduction

The word risk means many things to many people just as the word systems conveys many associations. In general, one wishes for high quality software delivered on time within budget. The risk associated with a computer project is either a qualitative or quantitative measure of the probability of meeting the project goals.

The risk is always two fold: a risk to the developer, and a risk to the purchaser of the software. The risk to the purchaser is the risk of obtaining poor quality software with a late delivery. In the most extreme case the developer may fail to deliver the software within any reasonable time period or may deliver a product which is so far below the quality and reliability required, that it almost unusable. The risk to the developer of the software is a cost overrun. This overrun may be due to the last minute addition of resources to meet schedule, perfecting the software when schedule is overrun, or to correcting an excessive number of errors after delivery. In the extreme case, the developer loses future sales because of the poor reputation of his product, or non-competitive future bids which are adjusted upward to reflect the costs of the last project.

For convenience we will separate these issues into software acquisition risk (cost and schedule) and software reliability.

Software Acquisition Risk

One measure of software acquisition risk is the cost of development. This is the cost to the producer for developing the software. If the scheduled development effort and procedures are inadequate for the project and the requirements, then either a poor quality product ensues or extra cost must be expended for additional testing and/or rewriting of the software. Thus, accurate estimates of the required cost are a necessity for the developer to gauge the risk in not meeting objectives should the estimate be in error, or if unexpected problems ensue. The user of the software, often represented by the government or commercial contract officer in large projects, must also make such calculations. A delayed delivery of a product or delays while the producer tries to eliminate enough bugs so that the product is usable also carries a cost penalty and must be treated as a risk. Sometimes such delays have a moderate impact, while in other cases they may be very costly.

Table 1. Factors Requiring Cost Estimation

- The project team members change significantly.
- There are significant changes in management.
- There is a significant change in requirements.
- There is a significant change in specifications.
- The cost budget for the project changes.
- The project schedule changes.

Development Costs

Development costs generally determine the success or failure of a computer project. An estimate must be made of development costs at the beginning of a project and must be updated when needed during the life of the development life cycle. In a “classical textbook” project the requirements are agreed to at the beginning of the project, an accurate cost estimate is made, and the project progresses on schedule and within budget. The costs are tracked during the project execution and follow fairly closely the initial estimates. Cost estimates include both the total

Continued on page 13

Continued from page 12

cost as well as the monthly expenditures during the development cycle. These cost projections are checked with the expenditures monthly (perhaps weekly). This procedure tends to minimize the risk of cost overruns by giving early warnings of major deviations from projected costs. Significant deviations between projections and expenditures, either positive or negative, require careful investigation to determine whether they represent slippage or acceleration of the project.

The primary cost risks are those listed in Table 1. A significant problem occurs when there are significant changes in the team members or the management. A few key members may leave the project, or there may be many major defections across the board due to aggressive hiring tactics of competitors. The contractor is still required to meet the contract objectives despite such changes, unless the customer is willing to renegotiate, perhaps for a no cost time extension.

Changes in requirements, cost, specification and schedule coming from the customer are not uncommon. In each case these must be the subject of a renegotiation between the customer and the contractor. As a practical matter, one must guard against the case of creeping escalation where the developer agrees to a succession of small changes in requirements or specifications which add up to a significant change that avoids renegotiation.

Software Reliability

Introduction

Software reliability is the probability that a software product will not fail during a time period, resulting in failure of the larger system in which it is embedded. In general, software failures are the result of residual errors not found in testing that are excited by a confluence of particular inputs and state of the system. The reliability level required of the system must be a function of the task it performs, thus, one can tolerate fewer crashes per year of air traffic control software than for the Windows 95 operating system.

Reliable Software vs. Software Reliability

Many use the term reliable software to refer to the use of various development procedures and processes to develop high quality reliable software on schedule and within budget. Most of the development procedures incorporated in the Software Engineering Institute's Capability Maturity Model (CMM) or the ISO9000 procedures are techniques for developing high quality software.

Software Reliability Modeling is the group of techniques for predicting the actual reliability which the software will achieve when development stops. The two common measures which are predicted are the software failure rate per hour of operation or the related meantime between software failures. Either of these parameters leads to a simple probability function which gives the probability of success or failure within a given time interval. A number of the most promising techniques for modeling software reliability are contained in the ANSI Software Reliability Standard.

Reliability as a Measure of Utility

In general the reliability of the system impacts strongly the usefulness of the system. An unreliable communication system requires duplicate channels, repeated calls, or delays to compensate for the lack of reliability, all of which carry a penalty. Thus, the risk of missing the reliability objective for software is a measure of the utility of the product. Modeling the reliability of the software and its variability as a function of the development parameters allows one to quantify the risk. Such a calculation should be carried out by both the developer and the customer.

Continued on page 14

Continued from page 13

Alternate Architectures Affect Reliability and Risk

Sometimes alternate development procedures or different architectures may be preferred, even if they are slightly more expensive or take a little longer, if they have a smaller risk. In some cases, redundant software may be a technique to raise reliability and to lower risk. Redundant software, version A and version B, is developed by two independent groups using the same specifications but different approaches. Computer A runs software A and computer B runs software B. The occurrence of an error in software A may bring down computer A, however, there is a good chance that software B does not contain this same error and that operation continues after a switch to computer B running software B. Thus, reliability increases and risk decreases.

Contact Information

Martin L. Shooman,
Professor of Computer Science &
Electrical Engineering
Polytechnic University
Glen Cove, New York 11542
shooman@rama.poly.edu

Ernest Lofgren,
Scientific Applications International Corporation (SAIC)
7 West 36St. - 10th. Floor
NYC, NY 10018

References

ANSI/AIAA American National Standard Recommended Practice for Software Reliability, R-013-1993, Feb. 23, 1993.

Boehm, Barry W., *Software Engineering Economics*, Prentice-Hall, NY, 1981.

Musa, J. D, et al., *Software Reliability Measurement, Prediction, and Application*, McGraw Hill, NY, 1987.

Shooman, Martin L., *Software Engineering: Design, Reliability, and Management*, McGraw-Hill, NY 1983.

Shooman, Martin L., *Reliability of Fault-Tolerant Computer Systems and Networks*, John Wiley and Sons, NY 1999.

InformationTechnology (IT) is Risky

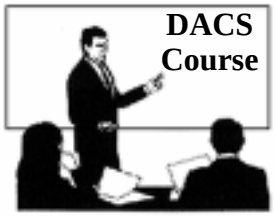
An Arthur Andersen Survey of CEOs, presidents, CFOs and Board Members at more than 150 global companies reveals the need to look more carefully at IT Risk.

- One in three executives does NOT have any IT Risk Management process in place; only half of those that do are confident the process is strong enough.
- Two out of three executives say their companies do NOT understand IT related risks well enough.
- Only 13% of executives believe IT strategy is well integrated with business strategy.
- Technology professionals are responsible for daily management of IT related risk at 51% of companies.

Source:
"Managing Business Risks in the Information Age", a study by Arthur Andersen and the Economist Intelligence Unit Ltd., 1998

For the complete study visit:
www.mbria.com

Network Reliability & Survivability Course



Course Date: November 9-10, 1998
Presented By: DoD Data & Analysis Center for Software
Instructors: Jay Bennett, John Healy and Spilios Makris

COURSE DESCRIPTION

This course will provide an overview of network reliability and survivability. It will introduce the structure of telecommunications networks to show how network architectures interact with reliability. It will explain how network reliability and survivability are managed in today's public networks. It will explain techniques for quantifying risks as a tool for decision making in designing and managing networks. One particularly important set of measurements analyzes reports of major telecommunications outages made to the Federal Communications Commission (FCC).

The course will also focus on:

- Network reliability and survivability during national emergencies, to support National Security and Emergency Preparedness (NSEP)
- Software contributions to network reliability, and the role software has played in recent catastrophic outages
- The evolution of networks and the implications on network reliability

These additional areas of focus are areas of active on-going study, so the course will introduce issues in these areas rather than existing solutions.

This two-day training course is intended for managers, policy-makers, and researchers who want to understand the implications of network reliability.

COURSE OUTLINE

Day 1

- Introduction
- Overview of Telecommunications Networks
- Network Outages: How they occur and what their impacts are
- How Risks are Managed

Day 2

- Quantifying Risk and Reliability as a Tool for Decision Making
- Network Reliability as a Component of National Security and Emergency Preparedness
- The Role of Software in Network Reliability
- New Risks in the 21st Century

LOCATION

2560 Huntington Avenue
Alexandria, Virginia 22303 USA
(703) 960-4906

On-Site options available. Call the DACS Customer Liaison for details.

REGISTRATION

Preregistration is required.
Note: This course is limited to 25 people.

Course fee is $995.

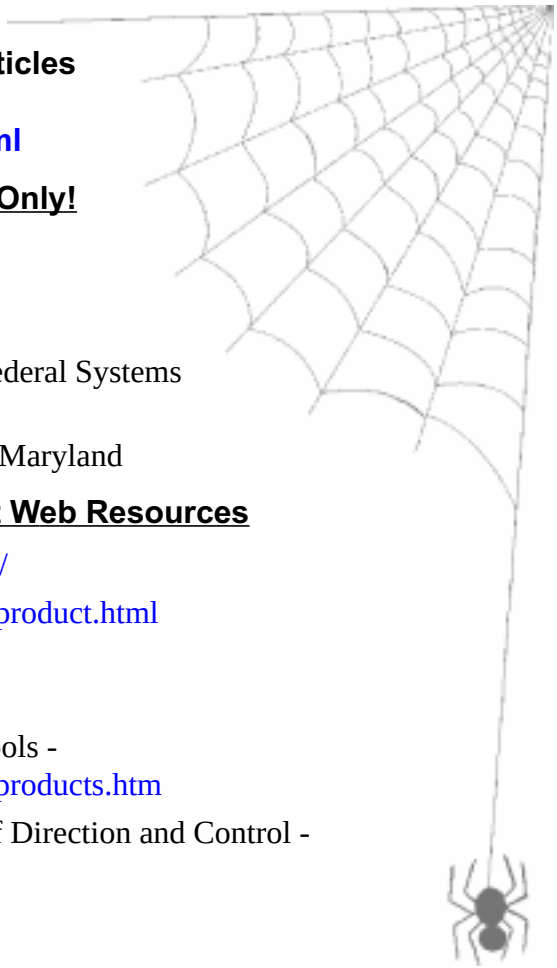
All checks should be made payable to
ITT Systems Corporation.
Company check, personal check or a
DD Form 1155 are acceptable payment options.

For further information please contact:
Anne Robinson, DACS Customer Liaison
DoD Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400
(315) 334-4905; Fax (315) 334-4964
cust-liasn@dacs.dtic.mil

Register on-line at: www.dacs.dtic.mil/training/networkrel/network.rel.shtml

Software Tech News on the World Wide Web

This newsletter in its entirety AND three additional articles are available on the web at:
www.dacs.dtic.mil/awareness/newsletters/listing.shtml



Additional Articles Available on the Web Only!

Software Risk Management - The Practical Approach
George Holt, MEI Technology Corporation

Risk Indicators
Joseph Kasser, U of Maryland and Victoria Williams, Keane Federal Systems

Riskit: Increasing Confidence in Risk Management
Jyrki Kontio, Nokia Telecommunications and Vic Basilli, U of Maryland

Other Software Risk Management Web Resources

DoD DACS Risk Management Topic Area - www.dacs.dtic.mil/

Acquisition Systems Management - www.acq.osd.mil/api/asm/product.html

CrossTALK Newsletter: Risk Management Issue - http://stsc.hill.af.mil/CrossTalk/1997/apr/dodacquisition.html

Defense Acquisition Deskbook: Risk Management Software Tools - www.acq.osd.mil/te/programs/se/risk_management/tools_and_products.htm

Program Managers Notebook: Risk Management as a Means of Direction and Control - www.dsmc.dsm.mil/pubs/pmnotebook/pmn4-5.htm

Risk Management Guide for DoD Acquisition - www.dsmc.dsm.mil/pubs/gdbks/risk_management.htm

Software Engineering Institute (SEI): Risk Management Frequently Asked Questions (FAQ) - www.sei.cmu.edu/organization/programs/sepm/risk/risk.faq.html

Software Engineering Institute (SEI): Risk Management Overview - www.sei.cmu.edu/organization/programs/sepm/risk/risk.mgmt.overview.html

Software Program Manager's Network (SPMN) - http://spmn.com/

US DoD/DTSE&E Risk Management - www.acq.osd.mil/te/programs/se/risk_management/index.htm

---

DoD Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400

First-Class Mail
U.S. Postage
P A I D
Colo. Spgs., CO
Permit No. 745

Return Service Requested

DoD DACS Products & Services Order Form

Name: Position/Title:

Organization: Acronym:

Address:

City: State: Zip Code:

Country:

Telephone: Fax:

E-mail:

Product Description	*Format*	*Quantity*	*Price*	*Total*
The DACS Information Package				
❑ Including: Software Tech News newsletter, an introduction to the DACS and a Products & Services Catalog	Document		FREE	FREE
Empirical Data				
❑ Architecture Research Facility (ARF) Error Dataset	Disk		$ 50	
❑ NASA / Software Engineering Laboratory (SEL) Dataset	CD-ROM		$ 50	
❑ NASA / AMES Dataset	CD-ROM		$ 50	
❑ Software Reusability Dataset	Disk		$ 50	
❑ DACS Productivity Dataset	Disk		$ 50	
Technical Reports		*FREE with Spreadsheet* →		
❑ A Business Case for Software Process Improvement	Document		$ 25	
❑ ROI from Software Process Improvement Spreadsheet	Diskette		$ 40	
❑ A History of Software Measurement at Rome Laboratory	Document		$ 25	
❑ An Analysis of Two Formal Methods: VDM and Z	Document		$ 25	
❑ An Overview of Object-Oriented Design	Document		$ 25	
❑ Artificial Neural Networks Technology	Document		$ 25	
❑ A Review of Formal Methods	Document		$ 25	
❑ A Review of Non-Ada to Ada Conversion	Document		$ 25	
❑ A State of the Art Report: Software Design Methods	Document		$ 25	
❑ A State of the Art Review: Distributable Database Technology	Document		$ 25	
❑ Electronic Publishing on the World Wide Web: An Engineering Approach	Document	*SALE Item!* →	**$ 5**	
❑ Object Oriented Database Management Systems	Document		$ 25	
❑ Software Analysis and Testing Technologies	Document		$ 25	
❑ Software Design Methods	Document		$ 25	
❑ Software Prototyping and Requirements Engineering	Document		$ 25	
❑ Software Interoperability	Document		$ 25	
❑ Software Reusability	Document		$ 25	
Bibliographic Products				
❑ Rome Laboratory Research in Software Measurement	Document		$ 25	
❑ DACS Custom Bibliographic Search	Diskette		$ 40	
❑ DACS Software Engineering Bibliographic Database (SEBD)	CD-ROM		$ 50	

Method of Payment:

❑ Check ❑ Mastercard ❑ Visa

Number of Items Ordered ☐ *Total Cost* ☐

Credit Card # _____ Expiration Date _____

Name on Credit Card _____ Signature _____

Mail this form to: DACS Customer Liaison
DoD Data & Analysis Center for Software
P.O. Box 1400, Rome, NY 13442-1400

Telephone: (315) 334-4905
Fax: (315) 334-4964
E-mail: **cust-liasn@dacs.dtic.mil**

This form is also on-line at: www.dacs.dtic.mil/forms/orderform.shtml

About the Software Tech News

Software Tech News Editorial Board Members

The Software Tech News is produced by the
DoD Data & Analysis Center for Software (DACS)

The DoD DACS is:

A United States Department of Defense Information Analysis Center (IAC)

Administered by the Defense Technical Information Center (DTIC), Ft. Belvoir, VA

Technically managed by Air Force Research Laboratory - Information Directorate (AFRL/IF), and Operated by ITT Systems Corporation, Rome, NY

Contacting the DACS:

DACS Customer Liaison:

E-mail: cust-liasn@dacs.dtic.mil
Telephone: (315) 334-4905
Fax: (315) 334-496

or

Visit the DACS Home Page for other great resources on Risk Management and 21 other Software Technology Topic Areas.

www.dacs.dtic.mil