

Software Development Risk Management

By

Karl Gallagher

August 6th, 2002
CSCI 510 Report
Prof. Boehm

Introduction

Software project managers can never have complete certainty over the course of their project. Changes in the customer's needs, loss of key personnel, and technical difficulties in implementing the software can all cause delay, cost overruns, or failure to meet the specified performance. Project plans include margins to guard against such problems, but the history of software development is littered with projects that exceeded their allotted budget and schedule by a factor of two or more, and others that were never debugged enough to be usable. To increase the chances of success the potential problems in a project must be identified and attacked as early as possible.

Some tasks can be carried out with almost zero risk of failure, but they are all repetitive ones—making a hamburger at McDonalds or some other routine production task. Software development is always about creating something new, and often trying to push the state of the art. A “zero risk” software project would have goals so timid that it couldn't justify deploying the product. The stakeholders must find the acceptable level of risk for their project—aggressive enough that the goal is worth attaining, but without risking a complete failure. Measuring the actual risk involved in the project requires active risk management by the project team.

Risk management requires effort by most or all of the team, and must be taken seriously or it will quickly degenerate into an empty ritual. Choosing the proper risk management approach is one of the most critical decisions the project manager can make. A too-burdensome process will detract from the team's ability to perform their primary tasks and increase the danger of discarding risk tracking completely. An insufficiently detailed process could overlook risks, fail to prioritize them properly, or neglect the follow-up needed to ensure they are handled properly.

A project's risk management plan must be chosen to match the size of the project team and the complexity of its task. The project leadership must make a sincere commitment to risk management to ensure that it does not get forgotten. Properly executed, this will ensure that the project stays on track and any problems that arise are dealt with before they grow large enough to destroy the whole project.

This paper will concentrate on appropriate risk management processes for software development projects. Various candidate processes will be analyzed and discussed and recommendations made for the risk management processes best used for projects of different sizes. The material drawn upon includes risk management techniques for all project types, not just software.

Origins of Risk Management

Risk management as a part of organizing a project has always been a concern. The Bible quotes Jesus speaking on the risk of project cost exceeding available funds¹. Other major

¹ Holy Bible, Luke 14:28-30.

projects in history have had to manage multiple risks—the historical novel The Pillars of the Earth revolves around the calamities affecting the construction of a medieval cathedral, many of them problems that also worry modern software project managers (late delivery of key components, loss of key personnel, funding shortages, design changes causing system failures, and introduction of new technology).

Modern management theory originally viewed risk as something to be eliminated, not lived with, but this changed in the 1970's and 80's as cost and schedule overruns proved risks were unavoidable. Up to that point managers had needed to claim their projects were zero-risk to maintain funding and stakeholder support—while some realized risk issues had to be dealt with, they were not discussed openly. Once risk was viewed as something to be traded against the other key variables of cost, schedule, and performance, risk management went from an informal art of the best project managers to a defined discipline.² Government guidelines and books on recommended practices shortly followed.

Risk Management Phases

A wide range of approaches have been developed for conducting risk management. All of them require the developers to identify risks to the project, analyze them to find their severity, then follow a process to ensure they are dealt with. Each of those phases has a number of options for executing them that vary widely in complexity. The project can choose completely different approaches for each phase, but in practice will use ones of equivalent detail.

Identification Phase

Brainstorming

The most basic method for identifying risks on a program is by “brainstorming”—getting as many people as practical to list potential problems on the project. This brings most of the domain knowledge available to bear on the problem and can bring out risks that are unique to the project. This cannot guarantee a comprehensive list of risks, but is easy to implement and is a good first step. Additional support for identifying risks can be found by bringing in consultants or other experts, either as part of the brainstorming effort or to objectively assess the project and point out possible risks.

Checklists

More formal methods for finding risks include using checklists of known risk factors and recording which ones apply to this project. Companies have developed checklists for in-house use.³ The Software Engineering Institute at Carnegie-Mellon has developed a detailed and comprehensive checklist for all software projects that is publicly available.

² Edgar, John D, “Controlling Murphy: How to Budget for Program Risk”, in Boehm, 1989, pp. 282-291.

³ McFarlan, F. Warren, “Portfolio Approach to Information Systems”, in Boehm, 1989, pp. 20-21.

Technical Report CMU/SEI-93-TR-6 is available on the web and provides a list of questions covering a wide range of potential risk causes. Many of the questions are branching, pursuing an issue in more depth depending on the answer given.

WBS and Requirements Documents

A good formal risk identification method is to use the project Work Breakdown Structure (or other document listing components of the project) to methodically discuss each section and bring out what risk would affect it. This will produce a comprehensive list but many risks will be minor compared to the project's driving risks. The project requirements document can also be examined, finding what risks may arise from each requirement. A similar approach is use the personnel organization chart to have each part of the project team look at the risks affecting its area of concern.

FMEA

Failure Modes and Effects Analysis is a formal method for analyzing designs for potential flaws. It follows the detailed flow of a design or a process to find all potential failures and their effects. Developers, possibly with outside consultants, examine the system together and brainstorm to find the failure modes. Each mode is then analyzed for its impact and probability, and mitigation plans developed, usually in the form of a design change or new procedure.

This is an excellent method for finding technical risk issues but cannot be the only risk identification method for the project for two reasons—it cannot be implemented until the system design is sufficiently detailed for the analysis and it will overlook risks arising from non-technical areas (personnel skills, funding cuts) or affecting issues other than technical performance (cost growth or schedule slips).

Domain History

Risk identification efforts should look at similar projects or past ones in the same organization or mission area and see what kinds of risks they faced. Their major problems will probably also arise in the new project. The development team should include engineers who worked on the past projects and draw upon their experience. If none of them are available to transfer to the new project, they should be consulted during brainstorming and project reviews. Other companies' projects with similar goals should be reviewed for possible risk items, either by consulting literature detailing the course of the project or bringing in consultants with appropriate experience. The other projects should be examined not just for the problems that they had but what risks that they had faced and how they were avoided.

Analysis Phase

Lists or Sorted Lists

The simplest method for dealing with the identified risks for a project is simply to list them as a reference. This involves no additional effort and still provides a handy reference for decision-making. In a low-overhead operation with a staff of one or a handful this may be the best choice as the listmaker also is the user and has more knowledge of the issues than any outside process could add. In any larger project, the list will be of limited use as there is no way to assess the severity of the risks relative to each other, and the credibility of the list is limited to the confidence in the listmaker.

The next step up in sophistication is to sort the list of risks in order of severity. This will require a consensus of the project's stakeholders on the relative danger of each risk, which may be hard to achieve (see "different subjective ratings" under "pitfalls", below). The sorting process will increase the credibility of the risk process (if all stakeholders are considered) and the new list will provide much better guidance to management on which issues should receive additional resources. The risks on the top of the list obviously deserve more attention and resources than the rest. The danger with this option is there is no way to tell the relative importance of two risks—a pair of risks adjacent on the list may be practically tied, or one could be more than twice as significant as the next.

Subjective Ratings

A start on identifying relative risk values may be done by subjectively rating the severity of a risk on an arbitrary scale, either "high" to "low" or in some range of numerical values (1-5 or 1-10 are typical)⁴. Breaking ties between risks of the same values still must be done subjectively. The main advantage of this method is that it supports decisions on how to allocate resources for mitigating risks. The distribution of higher-level ratings will give a much better idea of which risks must be actively combated and which can be tolerated.

As an example, Figure 1 shows a list of risks for a sample project. If they were only sorted in priority order the manager would be tempted to distribute resources to all of them equally, or (if resources were limited) to only give the top risk attention. Viewing the ratings shows that the first three items all deserve significant attention and the other two can be postponed or given minimal resources.

Risk	Rank	Rating
User interface acceptance	1	High
Retaining key programmers	2	High
Response time	3	Medium-High
Code size	4	Medium-Low
I/O speed	5	Low

Figure 1. Example Risk List

⁴ Boehm, Barry W., "Software Risk Management: Principles and Practices", IEEE Software, Jan. 1991.

Probability and Consequence Ratings

A significant increase in the usefulness of risk assessments can be achieved by developing separate ratings for the likelihood of the risk event occurring and the consequences of that event. The likelihood can be estimated much more accurately by itself than in combination with the consequence, and vice versa. This allows higher fidelity and confidence in the results of the risk analysis.

The two ratings can then be combined to provide a single value for the risk, allowing the project’s risks to be sorted by these values. The function for combining the individual ratings to get the value for the risk must be specified in the risk management process for the project. Different functions can produce different distributions of risks over the spectrum of values. Figure 2 shows two examples of functions for combining probability and consequence ratings into a risk value. Figure 2a shows the DoD Risk Management Guide table, which rates probability and consequence on a scale of “a-least severe” to “e-most severe” and divides the possible combinations almost evenly among the “L – M – H” scale. In contrast, Fig. 2b shows probability and consequence ratings from 1 to 10. Multiplying the values results in almost two-thirds of the possible combinations falling in the bottom third of the “0 – 100” risk value scale.

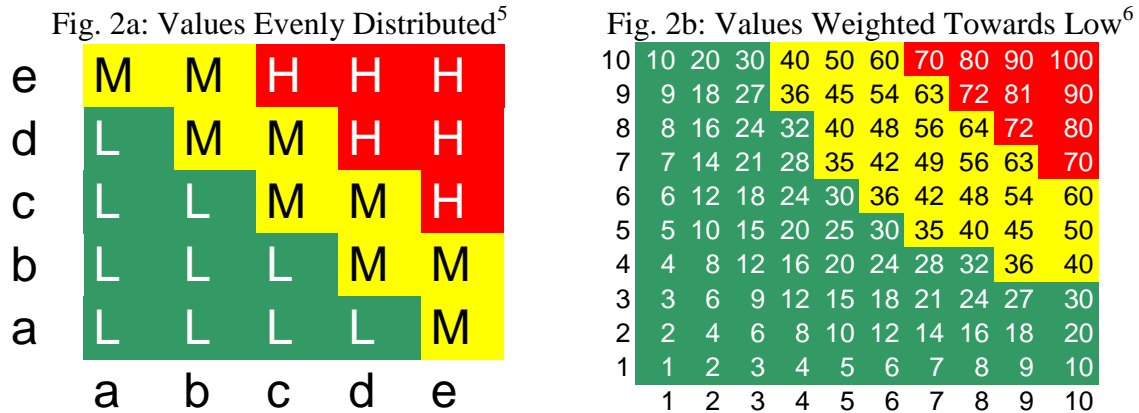


Figure 2. Comparison of Risk value calculations.

Note that the scale of probability ratings can be very different from one technical culture to another. NASA is very sensitive to low-probability events, as shown in Figure 3. All other risk processes found in the literature would consider a probability of 10^{-6} equivalent to zero.

⁵ Defense Systems Management College, “Risk Management Guide for DoD Acquisition”, Feb. 2001, p18.

⁶ Boehm, Barry W., “Software Risk Management: Principles and Practices”, IEEE Software, Jan. 1991.

Probability Level	NASA ⁷	DoD ⁸	Contractor ⁹
Most probable	10 ⁻¹ or higher	Near certainty	0.8 or higher
	10 ⁻¹ to 10 ⁻²	Highly likely	0.6 to 0.7
	10 ⁻² to 10 ⁻³	Likely	0.4 to 0.5
	10 ⁻³ to 10 ⁻⁶	Unlikely	0.2 to 0.3
Least probable	less than 10 ⁻⁶	Remote	0.1 or less

Figure 3. Probability/Likelihood Sensitivities.

While a risk’s probability can be measured by a single scale, the consequence can have effects on the project’s cost, schedule, and performance. A single risk can have radically different effects on each of those areas—a performance failure could have no effect on the development time or budget—so they must be evaluated separately. The best way to get comparable consequence ratings for cost, performance, and schedule is to create scales where the worst consequence would result in canceling the project—a complete failure to meet performance requirements or unacceptable cost/schedule overruns. The rest of the scales would go down proportionately to “zero risk” in the same number of steps. Figure 4 shows an example of such scales.

RATING	PERFORMANCE	COST GROWTH	SCHEDULE SLIP
5	Totally inadequate	> 20%	> 9 month
4	Degradation affecting usability	> 15%	> 7 month
3	Degraded	> 10%	> 5 month
2	Slight reduction	> 7%	> 3 month
1	Change to plan; meets requirement	> 2%	> 1 month
0	No impact	< 1%	None

Figure 4. Example consequence risk scales (hypothetical example, based on Raytheon proprietary original)¹⁰.

Calibrated Probability Ratings

A very successful technique for estimating risk levels has been to develop calibrated scales for measuring various factors contributing to the project risk. Scales are created for each of the factors considered drivers in avoiding risks—for example, requirements maturity, configuration management, experience in the technology being used, definition of external interfaces, etc. These scales are not numerically calibrated—instead they assess the state of each factor at different risk probability levels and describe them in a

⁷ NASA Code Q/Office of Safety and Mission Assurance, “Risk Management Procedures and Guidelines”, NPG: 8705.x, Dec. 2001.

⁸ Defense Systems Management College, “Risk Management Guide for DoD Acquisition”, Feb. 2001, p16.

⁹ Raytheon Company C³I Systems Segment, “National Polar-orbiting Operational Environmental Satellite System (NPOESS) Risk Management Plan (EMD Phase)”, Oct. 2001, p. 13. Raytheon proprietary document.

¹⁰ Ibid., p. 14.

way that allows a developer to quickly assess where his project is currently. Figure 5 shows an example of such a scale for evaluating the maturity of the project requirements.

Calibrated Risk Scale for Theory-W Requirements Maturity	
7	Stakeholders disagree over requirements
6	No win-win negotiations held
5	Win-win negotiation session held with some stakeholders
4	Win-win negotiation session held with all stakeholders
3	Win-win results documented
2	All stakeholders have commented on requirements document
1	Final draft under review
0	All stakeholders agree on final draft

Figure 5. Example calibrated risk scale (hypothetical example, based on TRW proprietary scales)¹¹.

The risk level in each scale is used to determine the probability of risks occurring. Rather than attempting to estimate a numerical probability, the developers can consult the appropriate scales and find a rating for the risk. The highest rating off the scales is then used as the “probability” value for the risk, and combined with a consequence rating as above to generate the risk assessment.

Developing a suitable family of risk scales requires intensive effort, and is only suitable for an enterprise-level risk management process that will allow the scales to be used for many projects. The scales developed at TRW under Steve Carman’s leadership have proven very valuable. They are also closely held proprietary information.

Mathematical Analysis

Some risk issues are best analyzed through mathematical approaches. A common technique is to use Monte Carlo analysis on the project cost and schedule estimates. To find the possible variation in the project completion time, each task in the schedule would have three estimates for its duration—a minimum, maximum, and most likely. A sample schedule is then generated, choosing a random duration for each task within those limits (using either a triangle or bell curve distribution). The total duration for the sample schedule is recorded, then the process is repeated with a new set of random estimates for each task. After some hundreds or thousands of runs, a set of data points for the schedule exists that provides statistical estimates of how likely it is to be done at different points in time¹². The same type of analysis can be done for the project development budget.

A more simple technique is to analyze risks by assigning a numerical probability value and loss amount to each risk. These are then multiplied together to get an “expected cost” value, which is used as the severity rating for the risk. Applying this to all the risks found in the project allows for precise rankings by the expected cost. Mitigation plans

11 TRW Space & Electronics Group, “National Polar-Orbiting Operational Environmental Satellite System (NPOESS) Risk Management Plan”, D31393 G, Jan. 2002, p. A-7. TRW proprietary document.

12 Billick, Mike, “Schedule Risk Management”, Dec. 2001, p. 17. TRW proprietary document.

can be evaluated by finding their effect on the probability and loss, producing a lower expected cost. That savings can be compared to the cost of the mitigation plan to see if it is cost effective compared to other options.

The trouble with this technique is the classic “garbage in – garbage out” problem. The results of the analysis are only as valid as the inputs, which can be very hard to estimate. The probability of an event occurring is typically dependent on many factors, few of which can be easily quantified. Extensive experience is required to create a good probability estimate and is not always available for a project team. A range of values may be used in place of a point estimate, but this can make it harder to compare risks with overlapping ranges. A strong advantage of using ranges is that they prevent managers or other people less familiar with the process from placing excessive confidence in a precise value that may not reflect the actual state of the risk.

Quantifying the consequence of a risk can be harder. Cost risks are the easiest—once the consequence is estimated it can be used in the calculations. Schedule risks can be translated into cost by estimating the additional costs of working the extra time or lost sales from being late to market. The dollar impact of a performance risk can be very hard to calculate, particularly for government projects. Commercial projects can find the impact of a performance shortfall by measuring how many customers would pass up the product.¹³ The project’s contract may specify specific penalty fees if a requirement is not met. But many projects have performance requirements written as a step function—a 5% shortfall has no penalty, a 10% shortfall cancels the entire project. These are not easily quantified and are much better handled by the methods above.

Process Phase

Identifying and even properly assessing the risks is not useful unless the information is applied to the conduct of the project. Allocating the personnel, budget, and other resources of the project must include the tasks needed to mitigate the risks, or all the risk assessments are useless. KPMG conducted a study of failed software projects that found 38% had some form of risk management in place, but half of them had ignored it during the course of the project.¹⁴ As with the assessment methodologies, the techniques for on-going risk management can vary widely in their complexity and effort required.

Reference List

The simplest method is to maintain a “reference list” of the project risks to be consulted when project decisions are being made. This has minimal overhead and can still be useful as a reminder to the developers to consider risk issues as part of all decisions. The

13 Gallagher, Karl, “Selecting a Launch Vehicle Design For Maximum Financial Success”, paper for AE 590, July 2002.

14 Glass, Robert L., *Software Runaways*, Prentice Hall, 1998, p. 14.

list should contain a short description of each risk and the actions planned to mitigate it. This can be very effective if the team's leadership is committed to risk management as part of the project goals.

Formal Decisions and Tracking

A more formal process would designate specific people to lead the risk management effort and to track each risk. A "Risk Management Board" (RMB) made up of the senior project managers would make the decisions on risk issues. A specific developer would be assigned to each risk to analyze it and develop a "Risk Handling Plan" (RHP) to describe the planned mitigation actions. The RHP would be approved or modified by the RMB and then made part of the project plan. A risk manager or coordinator would track the progress of all risks, organize the RMB meetings, and provide training to developers in using the risk process and analysis tools.

Decision Categories

A more sophisticated method for making decisions on risk mitigation efforts is to consider more options than directly attacking the risk.¹⁵ The "control" option is the usual default, with the project setting up additional tasks to either eliminate the cause of the risks or provide alternatives to reduce the consequence if it occurs. However, throwing resources at every risk that comes along could threaten the project's ability to perform its primary tasks. Also, the development team may not be in the best position to mitigate the risk. The "transfer" option requires the RMB to identify another organization that is better suited to the risk and convince it to carry out the risk mitigation efforts. For example, a subcontractor may transfer an interface risk to the prime, or a prime contractor could transfer foreign government issues to its government sponsor agency. "Acceptance" of a risk requires the stakeholders to agree that the risk justifies reducing the goals of the project, accepting lower performance to eliminate the risk. "Assumption" is the least pleasant option—the risk would be tracked with no active effort put into mitigating it. This is sometimes necessary due to resource limitations, but should always be done as a conscious decision, not by letting it "fall through the cracks."

Monitoring

Risk efforts must continue on an on-going basis to ensure project success. The project leadership should check the status of all risks and RHPs at regular RMB meetings. Project milestones should include briefings on the status of the risk management efforts. The developer assigned to execute the RHP for each risk should report regularly on progress.

Forecasting

A risk handling plan is easier to monitor if it includes a detailed forecast of how the risk will be reduced. In addition to describing and scheduling the tasks, the RHP should list

¹⁵ Conrow, Edmund H., Effective Risk Management: Some Keys to Success, AIAA, 2000, p227.

how each of the risk assessments changes at the completion of a task or other milestone. This should include the expected probability and consequence values, including values for any calibrated scales used. This will show how much the risk has been reduced at each milestone, and which aspects of the risk are being mitigated. The RHP forecasts can be combined to provide a forecast of the expected risk reductions for the project as a whole, which is a useful guide for customers and other stakeholders.

For an example of forecasting, a new project would begin at “6” in Figure 5 (p. 8). A forecast could be made that a win-win negotiation would be held in 2 weeks, the results would be documented in a week after that, the draft would be reviewed over two weeks, and then a final draft would be agreed to by all stakeholders in one more week. The forecast would look like this:

Time (wks)	0	1	2	3	4	5	6	7
Requirements Maturity Level	6	6	4	3	3	2	1	0

Any deviations from the forecast would be easily visible and the impact on the project risk level can be quickly calculated.

Customized Planning Stage

For a large project that will have a complex risk process, or a project in an enterprise that has a standard risk process, creating a project-specific “risk management plan” (RMP) should be part of the project start-up activities. Recycling an RMP from another project carries several dangers—it could have an inappropriate paperwork load, it could lack the required level of detail for the project, or it could have analysis scales focused on different issues. An enterprise-wide RMP has similar issues, and will need to be reduced to just that needed by the project. The new RMP should be developed by the risk coordinator or a manager to focus on the key issues of the project. For example, commercial and government-funded projects will need different checklists and analysis scales for evaluating problems with the customer. Once the analysis tools and processes for the project have been selected and documented training materials should be developed so all team members can learn the risk management methods as they join the team.

Recommended Processes

The appropriate risk management process depends on the type of project it is intended for. A small project run by a newly-formed team cannot spare the resources for a high-overhead process. Conversely, a large company running many projects can use a common process for all of them with a support structure in place to allow new projects to learn from how risk management was handled on previous projects.

Small Projects

A small project (less than 10 developers) cannot spare the time for the whole team to learn a complex process. But the complexity is unnecessary, as this type of project allows all members of the team to be familiar with all aspects of the project. This makes

it easy to informally develop a consensus on the severity of the project's risks. A sorted list of the identified risks can serve as a reference for all team members, and the risk mitigation measures can be incorporated into the regular project plan. The commitment of the entire team to risk management is essential—no infrastructure is in place to force anyone to do it.

Medium Projects

A medium size project (10 to 50 developers) is in the worst position for risk management—too large for all developers to form a consensus on the risks, but without the resources needed to support a complex formal risk management process. A full time risk manager is probably not affordable, but one developer should be designated as the risk lead in addition to his other responsibilities. This should not be one of the top managers—their workload is too heavy to keep risk as their top priority and it will be buried under other issues. Risks should be analyzed with subjective ratings but reviewed by the project manager or an RMB. A simple RHP should be written for each risk and monitored by the risk lead.

Large Projects

A project with 50 or more developers can and should have a full risk management program. One developer should be assigned to coordinate the risk activities, facilitating the identification efforts, training the rest of the team on analysis techniques, and leading the regular risk board meetings and tracking the progress on mitigation efforts. A single project will not have the resources to develop calibrated scales, so risks should be evaluated with uncalibrated probability and consequence scales, with Monte Carlo analysis used for the overall schedule and cost estimates. Each risk should have a lead developer assigned who is responsible for analyzing it, producing an RHP, and reporting status regularly.

Enterprise-Level Risk Programs

A large organization running projects of different sizes in parallel can support a high-quality risk management program. By setting up a risk management function outside the projects, it can support development of calibrated risk scales, update the risk program with lessons learned from different projects, help projects develop customized risk management plans, and provide training in risk analysis methods and processes to developers before they are assigned to a project.¹⁶ Projects can rely on the enterprise risk scales, Monte Carlo tools, and other methods while only using the material necessary for the immediate work. Having developers trained and experienced in risk management joining the project team allows a quick start to risk management efforts and better understanding of its methods and goals.

Implementing an enterprise-wide risk management program requires significant support both from the management and the worker-level culture. The payoff from investments in

¹⁶ Carman, Steve, "Project Risk Management Overview", Nov. 2001, p. 54. TRW proprietary document.

risk processes can take several years to be felt beyond a single project. If the company has a high rate of personnel turnover risk training will not be worthwhile because the trainees will probably leave the company before they are assigned to a project that will use their risk skills.¹⁷

TRW Space and Electronics has had significant success in growing its new risk management process from a small effort to develop scales used on a single sensor development project to one used on many large system development projects. While not all S&E projects use the new system it is steadily becoming more widespread. The contribution of risk management to several projects' success has made it more accepted. Support from the management level is providing not just training but additional tools such as web-based risk tracking software. TRW is well on its way to meeting the ideal described above.

Summary of Recommendations

	Small	Medium	Large	Enterprise
Identification				
Brainstorming	X	X	X	X
Checklists	O	X	X	X
WBS/Reqs	O	X	X	X
FMEA		O	O	X
History			O	X
Analysis				
Lists				
Subjective Ratings	X			
Prob./Cons. Ratings		X	X	
Calibrated Ratings				X
Monte Carlo		O	X	X
Process				
Reference List	X			
Formal Tracking		X	X	X
Decision Categories		O	O	X
Monitoring	O	X	X	X
Forecasting			O	X
Planning Stage			X	X

Figure 6. Summary of methods recommended for different project sizes.

X = Recommended. O = Optional.

17 DeMarco, Tom and Timothy Lister, Peopleware: Productive Projects and Teams, 2nd Ed., Dorset House, 1999, p. 107.

Risk Management Pitfalls

Many projects have had good risk management plans in place but still failed because the team was not committed to taking risk seriously. This is such a strong factor that experts have found the success of a risk management effort is unrelated to how technically sophisticated the analysis tools and process are¹⁸. There are a number of traps that can ruin a perfectly planned risk management plan.

Lack of Resources

If the project manager does not have any resources that can be applied to mitigating risks there is little use in active risk management. This can be a danger for small projects and/or start-up companies which have only obtained enough resources for their basic plan. In such situations the available resources must be concentrated on the “success contingency” or there will be no chance of profit. Some projects may be short of resources because of optimistic management assumptions that did not allow any contingency in the budgets¹⁹. All stakeholders in such a venture should be fully aware that they are unlikely to achieve all the project goals and could wind up with a complete failure.

For External Use Only

The interest among DoD and other government customers in risk management has created a new danger—the risk management plan created solely for the customer and external reviewers without being applied internally. In this situation the risk processes and handling plans may be prepared with careful formatting and multiple review signatures, but with the actual content bearing little or no resemblance to reality. Management decisions on such a project run ahead of the risk documentation, which must be updated to show that the task management has assigned is relevant to a newly-created risk. This type of risk process can still have beneficial effects for the project if the customer is flexible in requirements. Risks identified to the customer can often be solved through changes in requirements if a way to reduce the risk while not changing key performance goals is proposed. TRW’s NPOESS project has had considerable success in influencing customer requirements through the risk management process.

Differing Subjective Views

In the absence of calibrated risk assessment scales, team members will have to form a consensus on the severity of risks or accept the judgement of a designated authority. Consensus can be difficult to achieve given the differing viewpoints of (for example) project managers and technical specialists. The most practical approach has often been for the specialist with the greatest knowledge of the risk issue to prepare a detailed assessment as part of the Risk Handling Plan document. The RHP is then reviewed by managers and other applicable specialists at a Risk Management Board meeting and sent

18 Conrow, *ibid.*, p. 48.

19 Rosenau, Milton D. Jr., Successful Project Management, 3rd Ed., John Wiley & Sons, 1998, p. 163.

back for rework if other factors need to be taken into account. While sometimes requiring multiple iterations this has produced firm consensus on the status of risks and allowed all relevant information to be brought to bear.

Other methods of converging on a risk value have weaknesses that can cause major problems later in the program. A nominal consensus can be obtained by averaging the inputs of stakeholders or voting on the best value, but this process does not allow for additional information to be brought into the discussion. It also does not give appropriate weight to the team members with the best knowledge of the subject. The initial value assigned will probably have to be revisited soon as more knowledge of the issue is brought to light and these frequent changes will hurt the credibility of the process.

Assigning one person as the official judge of risk values can lead to even greater problems. In any reasonably complex system no one will have enough domain knowledge to completely grasp all possible risks, let alone the views of all stakeholders toward them. The judge's decisions will regularly be considered biased by other stakeholders, reducing the credibility of the risk management process and the motivation of other participants to make the process a success. This is most frequently seen in the form of the project manager setting risk assessments and the technical staff considering the risk process irrelevant to the actual problems they face.²⁰

Failure to Follow-Through

Even if agreements are made on risk mitigation plans there must be support for follow-through on all management levels. The Denver International Airport baggage handling system was an embarrassment nationwide for DIA in large part due to failed risk management. The contractor assigned to implement the system, BAE Automated System, had originally signed up to build a much smaller system only supporting one airline. DIA liked the concept and wanted it scaled-up to support the whole airport. BAE considered this scope change a major risk and was reluctant to undertake the project, only agreeing after getting DIA's agreement on several risk control measures. These concentrated on giving BAE personnel top priority access to all key areas to allow as much time as possible for implementing the new system. Unfortunately, the other contractors working on DIA were unwilling to give way to the baggage handling system workers who continuously found their areas being worked on by other personnel or simply blocked off from access. DIA would not support BAE and it attempted to complete the system under those constraints rather than accuse its biggest customer of breach of contract. The failure to be ready by DIA's opening date was probably no surprise to any of the people actually working on the baggage handling system. When the City of Denver then demanded penalty payments BAE responded with the previously-avoided breach of contract lawsuit. The eventual compromise involved a significant scale-back in the complexity of the deployed system.²¹

²⁰ Conrow, *ibid.*, p. 88.

²¹ Glass, *ibid.*, pp. 23-51.

Aversion to Documenting Some Risks

A good risk management process can be compromised if the project management is averse to documenting certain risks or to stating their rating accurately. This is a severe danger when a project is in competition with another for a government award.

Minimizing risk values will be seen as part of the “sales pitch” need to obtain customer approval. Some risks, particularly ones relative to the team’s processes, will be ignored on the grounds that raising them as an issue will hurt the credibility of the rest of proposal. For example, if the “subcontract management” process is listed as a risk the managers would fear the customer discounting all performance from subcontractors. This attitude may remain in place even when the issue causes serious problems on the program, fully visible to the customer.

Excessive Overhead

If a risk management process is too high a burden on the project staff, they will not use it effectively and may even evade the process at every opportunity. TRW’s NPOESS project has a recurring problem of new or remotely located personnel generating risk analyses and handling plans hastily or incompletely, even though considerable effort went into minimizing the workload of risk management. Another project at TRW had a complex risk management process with a full-time manager responsible for implementing it. The paperwork involved was strongly resisted by the team, and when the risk manager left the process was allowed to fall into disuse. By the time they were briefed on a new risk management process the project’s management team considered it impossible to meet any of the project’s cost or schedule goals. Had a risk management process with less sophistication been implemented and accepted by the team it could have averted this fiasco.

Recommended Risk Management Plan for CSCI 577 Student Teams

A student team is by definition a small project. Their risk process will need to be low-overhead to make it easily usable. Since the team members will be working closely together and have similar backgrounds they will have an easy time reaching consensus on risk issues. 577a teams should identify risks with an initial brainstorming session supported by a class-specific checklist, followed by a risk assessment of the SRD and/or OCD documents. The initial 577a lecture includes a “primary risk items” list that can be used as the checklist. 577b teams should review the 577a risk list and then go over the SRD and SSAD to identify new risks. For simplicity the risks should be analyzed with subjective ratings, using only a High/Medium/Low scale (Fig. 7).

		Consequence		
		Low	Medium	High
Probability	High	Medium	High	High
	Medium	Low	Medium	High
	Low	Low	Low	Medium

Figure 7. Risk evaluation matrix for student teams.

The team should create a list of the risks, their ratings, and the planned actions to deal with them. This list would be reviewed every two weeks at a team meeting. The review should begin with the addition of any newly-identified risks to the list. Then for each risk the team should discuss whether its rating has changed and whether the mitigation efforts are on track. If any risk is no longer a danger it should be removed from the list. The updated list is then recorded on the form in Fig. 8 and submitted to the faculty for tracking purposes (preferably through a webpage form, though paper would do).

Team Name/Project:				Date:	
New Risks:					
Name	Probability	Consequence	Rating	Mitigation Actions Planned:	
New Risk 1	M	H	H	Planned actions.	
New Risk 2	etc.				
Existing Risks:					
Name	Probability	Consequence	Rating	Mitigation Plan	Status
Risk 1	M	M	M	Planned action	On schedule
etc.					

Figure 8. Form for reporting risk status.

Risks should also be briefed at the LCO and LCA review. Customers should be made aware of the project risk list and their input solicited. This can be a useful technique for urging customers to provide input on other issues if their lack of participation is labeled a risk.

The emphasis in this risk management program should be that the students will have a better chance of succeeding if they monitor their risks carefully. If they think of it as only a monitoring method for the faculty, the teams will not put in the required effort and the data submitted will be of poor quality (This is the “external use only” pitfall described above). The customers and faculty should be open to mitigating risks by changing requirements or providing outside support. If the only resource available for dealing with risks is overtime by team members they will quickly learn to avoid finding risks and may have friction between team members over willingness to supply overtime²². The risk plan should be updated yearly based on feedback from 577 students. Creating improved risk identification checklists and appropriate calibrated risk scales could be a useful investment for future classes.

Conclusion

A project manager has a wide range of techniques to choose from in creating a risk management plan. At the simplest end, risk management can be informal reminders to keep the team aware of the risks and how they are being controlled. The opposite extreme will have a risk management program with full time staff, detailed forms and

²² DeMarco, *ibid.*, p. 180.

checklists, training programs, formal review of risk handling plans, and careful monitoring of progress.

For any project there will be a particular level of risk management effort that represents the best trade-off between effort and benefits. Generally, the larger the project or organization the more effort it can afford to invest in risk management. Small projects have the advantage of being able to create a consensus of the entire group on risk issues, which reduces their need for formal support systems. The large organizations need the complex risk analysis tools since comparing risks in different areas of the project is difficult without a standard reference. So the level of risk management should reflect not just the resources available but how much the development team needs a common yardstick.

Once the risk management program is in place the project leadership must support it, ensuring that risk activities don't get brushed aside by more immediate concerns. If there is no active support, the risk management efforts will cease or become only window dressing. This may also force the management to press outside stakeholders to meet their commitments, since many risks are driven by customers or suppliers.

CS 577 classes can be a good laboratory for a minimal risk management system. The teams are small, work together closely, and have the common background of being in the same graduate program. This lets the risk management approach rely more on pooling the team's knowledge of problems instead of forcing them to learn a complex risk assessment methodology. The success of the risk efforts will depend on whether teams feel they are doing this to save themselves trouble in the future or solely to generate data for the class archives.

Risk management is a very useful set of tools for avoiding project disasters. Managers must choose the right tools for their situation and provide continuous support to the risk reduction efforts, but that investment can easily pay for itself by preventing one or two critical problems from destroying the project.

Bibliography

Books:

Holy Bible (King James Version)

Boehm, Barry W., Software Risk Management, IEEE Computer Society Press, 1989.

Conrow, Edmund H., Effective Risk Management: Some Keys to Success, AIAA, 2000

DeMarco, Tom and Timothy Lister, Peopleware: Productive Projects and Teams, 2nd Ed., Dorset House, 1999

Follet, Ken, The Pillars of the Earth, Signet, 1990.

Glass, Robert L., Software Runaways, Prentice Hall, 1998

Rosenau, Milton D. Jr., Successful Project Management, 3rd Ed., John Wiley & Sons, 1998

Published Articles:

Boehm, Barry W., “Software Risk Management: Principles and Practices”, IEEE Software, Jan. 1991.

Edgar, John D, “Controlling Murphy: How to Budget for Program Risk”, in Boehm, 1989, pp. 282-291.

McFarlan, F. Warren, “Portfolio Approach to Information Systems”, in Boehm, 1989, pp. 20-21.

Official Standards Documents:

Defense Systems Management College, “Risk Management Guide for DoD Acquisition”, Feb. 2001.

NASA Code Q/Office of Safety and Mission Assurance, “Risk Management Procedures and Guidelines”, NPG: 8705.x, Dec. 2001.

Raytheon Company C³I Systems Segment, “National Polar-orbiting Operational Environmental Satellite System (NPOESS) Risk Management Plan (EMD Phase)”, Oct. 2001. Raytheon proprietary document.

TRW Space & Electronics Group, “National Polar-Orbiting Operational Environmental Satellite System (NPOESS) Risk Management Plan”, D31393 G, Jan. 2002. TRW proprietary document.

Unpublished Articles and Presentations:

Billick, Mike, “Schedule Risk Management”, Dec. 2001, p. 17. TRW proprietary document.

Carman, Steve, “Project Risk Management Overview”, Nov. 2001, p. 54.

Gallagher, Karl, “Selecting a Launch Vehicle Design For Maximum Financial Success”, paper for AE 590, July 2002.

acronyms

RHP – Risk handling plan

FMEA – Failure modes and effects analysis

RMB – Risk management board

RMP – Risk management plan

SRD – System Requirements Definition document.

OCD – Operational Concept Definition document

SSAD – System and Software Architecture Definition document