DYNAMIC POSITIONING CONFERENCE

September 28-30, 2004

Risk

Software Risk: Why Must We Keep Learning From Experience?

Don Shafer, Chief Technology Officer
Athens Group, Inc., Houston, Texas, USA
www.athensgroup.com

Return to session directory

Abstract

Is there software risk in the dynamic positioning regime? Cited by Kongsberg Offshore A.S. in HSE Evaluation Report No. 00-4002: “The statistics show that the software failure is about four times as frequent as the hardware failure and slightly more frequent than the pure thruster failure.” Based on IMCA data, the percentage of “Loss of Position Class 1” DP problems that were caused by software for a recent 5 year period was 33%. Can we mitigate the risk? FMEA, FMECA and good software engineering practices will go a long way toward reducing today’s DP software risks. This is not rocket science but the lack of good engineering practices. The Airbus 300 series and the latest Boeing 7x7 aircraft are completely fly by wire. The airbags in your automobile have autonomous processors with embedded software. Embedded medical devices contain processors run by software. We would never tolerate the number of software failures in these devices that occur on DP systems. Why don’t they fail at a 33% rate? This paper is a complementary tutorial in software risk to the presentation. Included here are complete sets of software and hardware life cycle processes along with a mitigation model for identifying, managing and eliminating software risk within DP systems. Several recent incidents were analyzed within the presentation to show how these processes would have mitigated the potential for failure. Readers can implement these processes in their own organizations to reduce software failures.

Introduction

Software risk management is the formal process in which risk factors are systematically identified, assessed, and mitigated. The determination of the risk in a project either due to external or internal causes is a major part of project management. The Project Management Body of Knowledge[1] defines risk as “A subset of project management that includes the processes concerned with identifying, analyzing, and responding to project risk. It consists of risk identification, risk quantification, risk response development and risk response control.” In order to aid the project manger in risk determination and management, this chapter will answer these questions:

- What is risk management all about?
- What are some risk management models?
- How are risks identified?
- How are risks analyzed and quantified?
- How are risk responses developed and risks controlled?
- What are the steps in developing a risk management plan?

Where does Risk Occur in DP Software Development?

Risk management begins with the exploration of the concepts leading up to acceptance of a software development project. A good project manager is a good risk manager. Risk management continues throughout the life cycle until the product is delivered. Risk analysis and contingency planning continue through the implementation stages of the product life cycle. Risks are analyzed and prioritized on no less that a weekly basis and the current top ten risk list is presented at each weekly project status meeting. The only way risk mitigation occurs is through working the risks with the project team. Figure 1 shows how risk management fits within the software project management life cycle.

If you do NOT follow a formal life cycle in your DP software development: this is your first major risk!

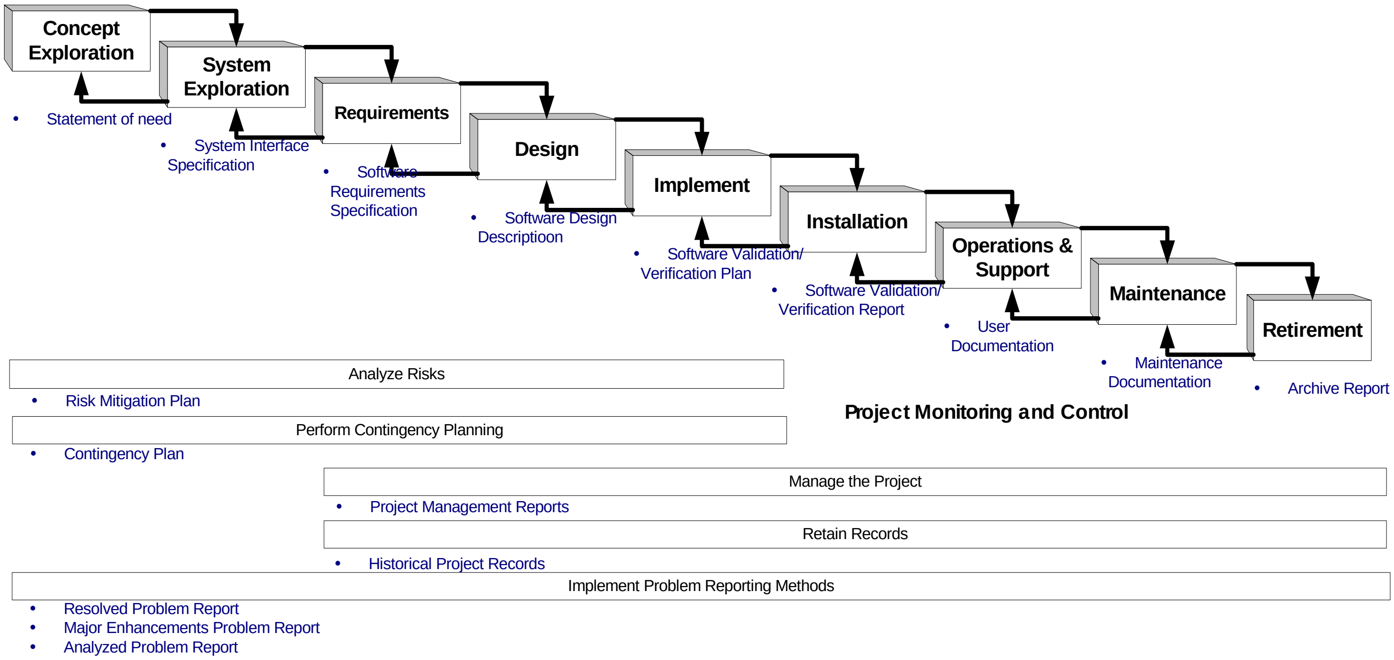

Figure 1. Where Risk Management Occurs in the Product Development Life Cycle

What is Risk Management?

Risk management is about understanding the internal and external project influences that can cause project failure. Once the project plan is built, a risk analysis should be performed on it. The result of the initial risk analysis is a risk plan that should be reviewed regularly and adjusted accordingly. The main purpose of risk management is to identify and handle the uncommon causes of project variation. This is captured in a formal process in which risk factors are systematically identified, assessed, and provided for.

Within our software domain, the SEI definition is more than adequate: "Risk is the possibility of suffering loss." In a software development project, the loss describes the impact to the project which could be in the form of diminished quality of the end product, increased costs, delayed completion, or outright project failure. Risk is uncertainty or lack of complete knowledge of the set of all possible future events. It can be classified as either favorable or unfavorable future events. Strictly speaking, risk involves only the possibility of suffering harm or loss. Risk can be categorized as:

- Internal, within the control of project manager, and
- External, outside the control of project manager.

A software development project plans is only the best educated guess that can be made for planned events. Much can happen throughout the life cycle of the project that was not incorporated into the plan. This is variation. A good project manager minimizes variation through process management. The project manager deals with risk resulting from these three general classes:

- Known knowns – these are risks that are known to the project team as both a category of risk and a reality of this project. An example of this is that not having an executive sponsor for a large project places continued funding at risk. In this project, if there is no executive sponsor, this is then a known type of risk and it is known to exist on this project. A known known risk could also be a category of risk that has been mitigated on this project. These are noted and described in the Project Management Plan.

- Known unknowns - these are risks that are known to the project team as a category of risk but not known as a reality on this project. An example of this is that not having access to the ultimate end user is a risk in that requirements may not be correctly identified. In this project, if it is unknown whether there is access to the ultimate end user, this is then a known type of risk and it is unknown if it exists on this project. These are described in the risk management plan where they are prioritized and worked on a weekly basis.

- Unknown unknowns - these are risks that are unknown to the project team as both a category of risk and a reality of this project. Although project managers use broad categories of risk, an unknown unknown can arise in the technology area. An example of this is if the project must use a specific technology solution because it is dictated by the terms of the contract for the project. Even though this in itself is a risk, with no experience in the tool, the project manager cannot know all the potential risks inherent in the tool's use. These can only be addressed in the most general of way by putting in place a budget for contingencies.

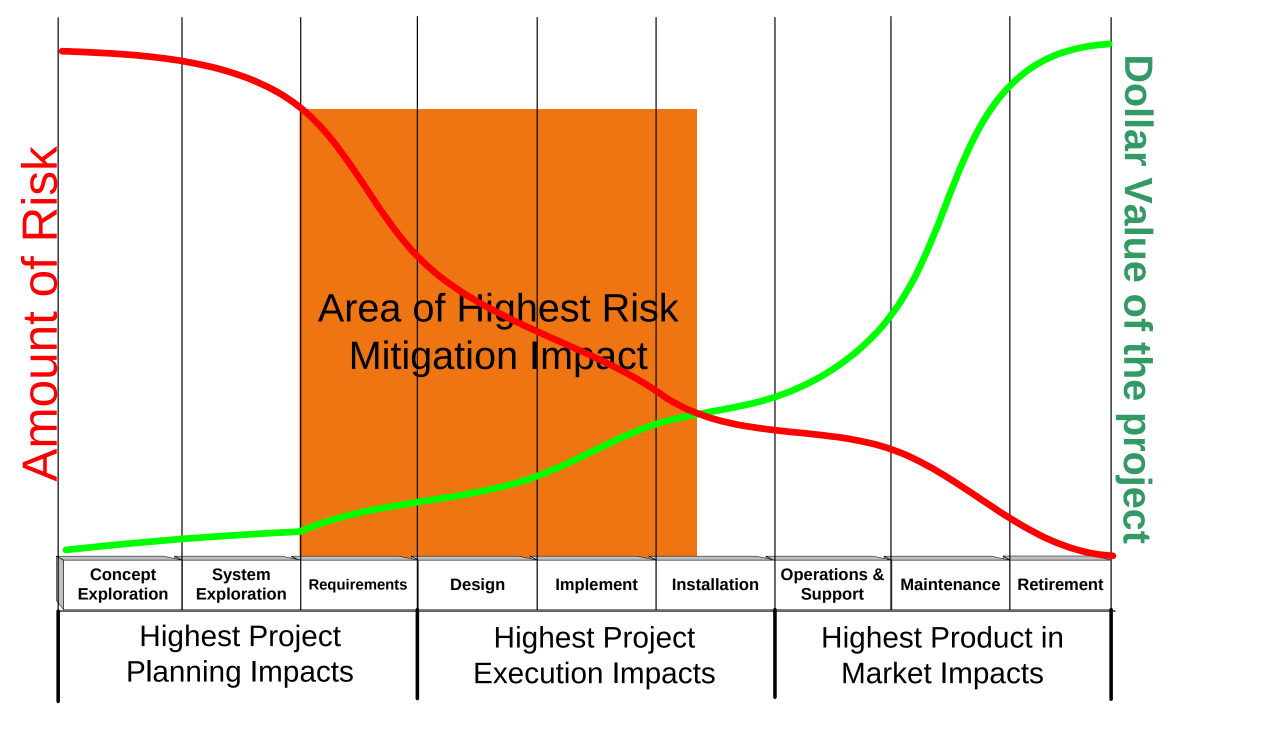

Figure 2. Project Risks During the Life Cycle

Using both the project management and risk management plans, the project manager begins to identify contingency budgets. Figure 2: Project Risk in the Life Cycle, shows the relationship between risk and the dollar value of the project over the life cycle. Mapped across the IEEE 1074 project and product live cycle phases, the project investment gradually increases through the end of the requirements phase. The concept and system exploration along with requirements are the first three life cycle phases and are the phases where project planning has the greatest impact on risk mitigation. The inherent project risk is highest in these three phases and drops through project execution.

Design, implementation and installation phases have the highest project execution risk reduction potential. In a world with experienced project managers and well behaved projects, the risk continues to be reduced and the dollar value of the project investment smoothly increases. The final three phases, operations and support maintenance, and retirement, have the lowest software development risk and the highest dollar investment. These three phases derive the highest risk impact from the product market.

The part of the figure labeled “Area of Highest Risk Mitigation Impact” covers requirements, design, implementation and part of installation. This is the area of the project where the project manager has the most impact on risk mitigation. As long as risks are determined and mitigated, the amount of risk will smoothly decrease and the project investment will continue on its predicted path. If risks are not identified and mitigated, the project cost will rapidly increase.

Project managers, as they are identifying the risks within the project life cycle and possible mitigation tactics, need to identify their level of risk tolerance. Varying by individual and organization, Figure 3: Variations in Risk Tolerance, was derived from comparative responses to alternate decision acts. A line going from the origin to the upper right corner at a 45-degree angle would represent neutral risk. This line represents the line of equilibrium points between the amount of dollars at stake and the probability of the risk event occurring. Risk seeking individuals and teams follow the upper curved line, increasing the potential loss due to the risk event occurring. Risk avoiders are below the neutral line. Although risk may be avoided there is an opportunity cost occurring below the neutral line. As more money is invested over time to avoid risk that will not occur, that money is lost for other investments. The opportunity to invest those monies is lost and the profit that could have been made the opportunity cost. At a minimum, it is the interest lost by investing the monies in risk free government bonds.

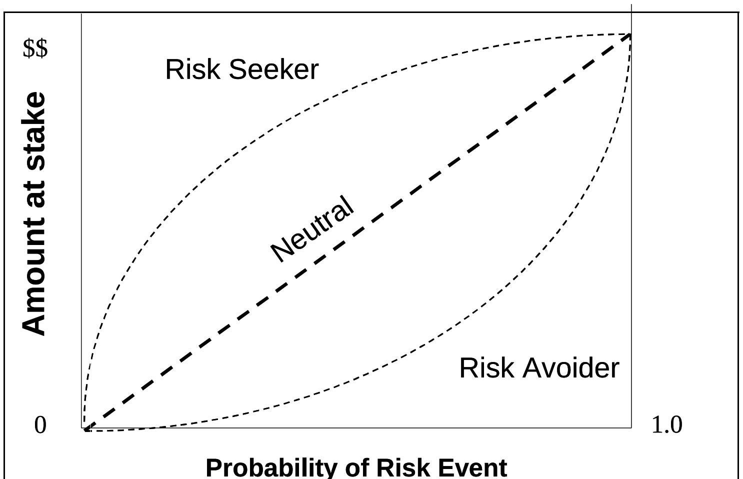

Figure 3. Variations in Risk Tolerance

Business risks must be separated from the project idea of a “pure risk”. Business or inherent risk is the chance for either profit or loss, which is associated with any business endeavor. Pure or insurable risk, only involves the chance for a loss. Examples of these losses are: direct property loss, indirect consequential loss, personnel loss, and legal liability. Direct property losses include assets insurance, auto collision, fire, and theft. Examples of indirect consequential loss are: contractor’s protection for indirect losses suffered by a 3rd party, removal of debris, and replacement of equipment. Legal liability is protection against legal actions for: design errors, public injury, and project performance failures. Finally, personnel pure risk examples are workman’s compensation and employee replacement costs.

Part of what risk management is “all about” is risk quantification. Concepts of risk quantification are:

- Risk Event: The precise description of what might happen to the project.
- Risk Probability: The degree to which the risk event is likely to occur.
- Amount at Stake: The loss if the outcome is unsatisfactory.
- Risk exposure: The overall liability potential of the risk.

Relevant DP Risk Management Model

Barry Boehm’s Risk Management Process was first presented in the article “Software Risk Management” published by IEEE Computer Society Press in 1989. Figure 4 shows the graphic representation of the model. Risk management consists of the two activities of risk assessment and control. Risk assessment is further divided into risk identification, analysis and prioritization.

Using checklists, decision-driver analysis and problem decomposition, risk identification is accomplished. For problem domains where the project manager and team have previous experience, checklists can be developed to guide in insuring all previously “known knowns” risks are identified for this project. For projects that are in a new domain or a dramatically different technology from the team’s experience, decision driver analysis and problem decomposition are used. With these tools, the project team can take a deeper look into the problem domain for which the software will be developed and decide on the general classes of risks to be faced.

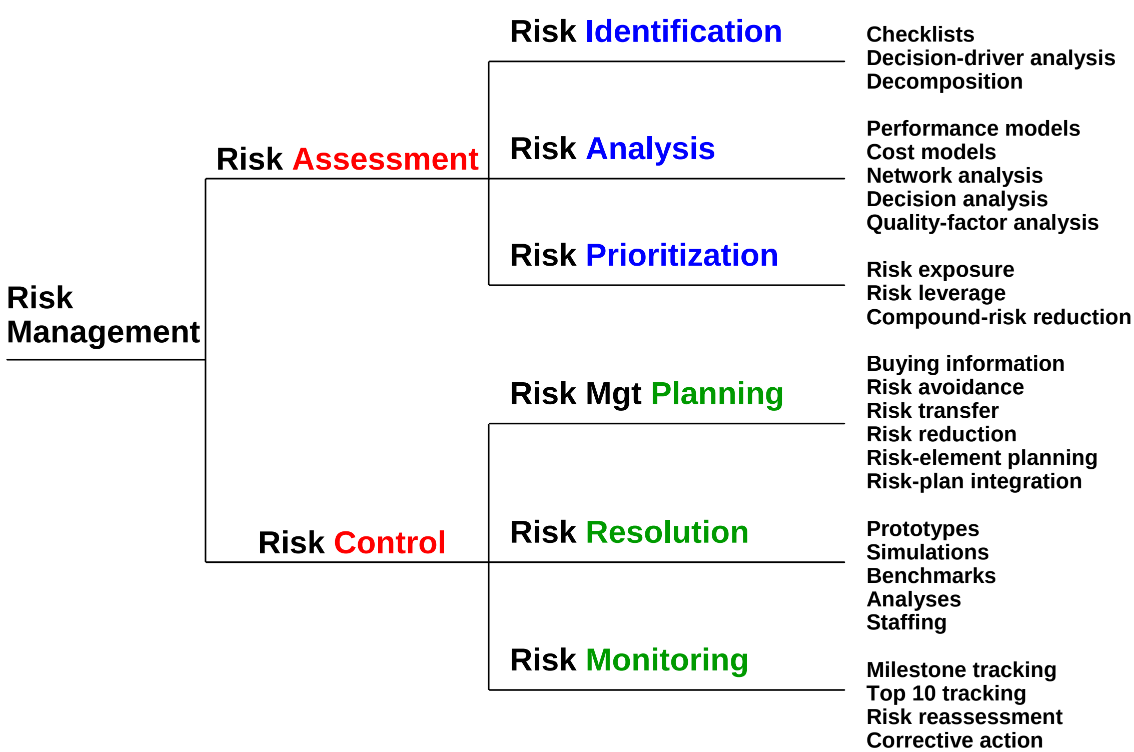

Figure 4. Boehm’s Project Risk Model

Analysis of the risks identified is done through modeling performance and cost, and analyzing network, decision and quality factors. Performance and cost models allow the project manager to produce what-if scenarios based on performance and cost variables. The values of these variables are estimated based on the inherent knowledge of the problem domain. Advanced Monte Carlo statistical techniques can be added to gain further analysis area coverage. Network, decision and quality factor analyses provide the project team with enhanced views of the information developed during problem decomposition in risk identification.

After the risks have been identified and analyzed, their relative potential for occurrence and impact on the project must be determined. This risk prioritization allows the project team to focus on those critical few risks that will have the greatest potential for causing project failure. The calculation of risk exposure has been previously described. This should be done for each high priority risk. Risk leverage is a further quantification of risk exposure. First calculate the current risk exposure (RE) and then the RE after completion of mitigation efforts. Calculate the costs of the risk mitigation efforts. Subtracting the RE after from the RE before and dividing the result by the mitigation cost derive the measure of the relative costbenefit. Compound risk reduction is simply the decomposition of multi-factored risks into single factor risks so they can be prioritized within the risk mix.

Risk control consists of risk management planning, risk resolution and risk monitoring. As with risk assessment, these three components are supported by sets of tools and techniques.

Risk management planning uses the tools of buying information, and risk avoidance, transfer, reduction, element planning and plan integration. Buying information is another way of saying, "hire the experts!" It can consist of contracting with subject matter expert consultants, subscribing to databases of topical information and subscribing to research services.

Risk avoidance is simply finding a way to re-structure the project and product to avoid that risk. Risk transfer usually involves buying insurance against the occurrence of the risk. Risk transfer, on the other hand, is the actual transfer of the responsibility for that part of the project with the inherent risk to another organization.

Risk element planning and risk plan integration work together in the structuring of the project plan. By decomposing the risk into its constituent parts, each element of the risk can be separately addressed and solved. This is the divide and conquer strategy to risk mitigation. Risk plan integration takes these separate elements and incorporates their solution into the overall project.

Risk resolution is accomplished through prototypes, simulations, benchmarks, analyses and staffing. At this point in the risk model, the mapping to Boehm's spiral model of software development becomes very apparent. Prototypes, simulations and benchmarks usually involve additional tools and capabilities. These tools have tremendous payback in risk reduction and mitigation, but there must be an investment in the tools and training to realize these benefits.

Milestone tracking, top 10 risk tracking, risk reassessment and corrective action provide the tools for risk monitoring. These tools are all part of the steps that a project manager takes to implement complete risk management. They will be discussed in the section on how to develop a risk management plan.

Identifying Risks

The process of risk identification is accomplished using the same tools as any analysis task. Start out with the team and the customer brainstorming possible risks to develop lists of "known unknowns". Use checklists of problems from prior projects retrieved from the project repository or knowledgebase. Examine all project assumptions in the project plan for the slightest hint of risk. Pay special attention to those that assume a "rosy future" where "everything works." Interview stakeholders for risk identification and quantification.

Take the work breakdown structure and network diagrams from the project management plan and look for precedence bottlenecks. These will show up as tasks that require many other tasks to complete before they can begin. These are the real choke points in the project planning network and have the highest risk reaction with schedule slips. Sometimes, flowcharting a process helps spot risky areas. If the process is not familiar, draw the flow of execution to see all the dependencies to successful completion. Examine the sources of key decisions in the project. Look for decision drivers considering these different types of risks:

- Technical
- Operational
- Political
- Legal
- Regulatory
- Market

- Social
- Internal
- External

The three basic risk areas – supportability, technical and programmatic – increase risk to technical quality, cost and schedule. Keep in mind that cost and schedule are always inherently risky. Table 1 shows possible risks for these risk sources. Technical risks are a major part of the software development business since software is the driver of high technology. Programmatic sources arise from the process of trying to manage the software development project. As the software product nears completion, the risks inherent in the software delivery, installation and maintainability are very real and obvious risks.

Technical Sources	**Programmatic Sources**	**Supportability Sources**
Physical Properties	Material Availability	Reliability and Maintainability
Material Properties	Personnel Availability	Training & Training Support
Radiation Properties	Personnel Skills	Equipment
Testing and Modeling	Safety	Human Resource Considerations
Integration and Interface	Security	System Safety
Software Design	Environmental Impact	Technical Data
Safety	Communication Problems	Facility Considerations
Requirements Changes	Labor Strikes	Interoperability Considerations
Fault Detection	Requirements Changes	Transportability
Operating Environment	Political Advocacy	Computer Resources Support
Proven or Unproven Technology	Contractor Stability	Packaging, Handling, Storage
System Complexity	Funding Profile	
Unique or Special Resources	Regulatory Changes	

Table 1: Risks Mapped to Sources

Analyzing and Quantifying Risks

There are some old and new risk analysis tools and techniques to use. The previously discussed tools for analyzing the identified risks are:

- Brainstorming
  1. Offer risk analysis ideas without judgment or evaluation
  2. Build on the ideas offered
  3. Repeating until all ideas on risk analysis are exhausted
- Delphi Method
  1. Select a panel of experts (isolated from each other and unknown to one another)
  2. Prepare and circulate a questionnaire about a risk
  3. solicit risk handling approaches & opinions
  4. share all responses & statistical feedback with entire group
  5. repeat until converge on a consensus approach

New analysis techniques that project managers and teams can use for risk analysis are:

- Sensitivity Analysis
  1. choose a few variables with big impact to the plan
  2. define a likely range of variation
  3. assess effect of changing them on project outcome
- Probability Analysis
  1. similar to Sensitivity Analysis
  2. adds a probability distribution for each variable, usually skewed to eliminate

optimism

- Monte Carlo Simulation
  1. similar to Probability Analysis
  2. assign randomly chosen values for each variable
  3. run simulation a number of times to get a probability distribution for the outcome
  4. produces a range of probabilities for the outcome
- Utility Theory
  1. comprehends decision maker's attitudes toward risk
  2. viewed as theoretical
- Decision Tree Analysis
  1. graphical method
  2. forces probability considerations for each outcome
  3. usually applied to cost and time

The analysis techniques lead directly into the quantification of the risk – assigning a numeric value to an individual, cluster or class of project risk. The project manger must keep in mind the one, most critical aspect of risk quantification. All of the numeric values are derivatives of best estimates, also known as guesses. Since the time at which these risks are predicted to occur has not yet arrived, there is no certain knowledge of what, if any, impact the risk will really have on the project. The job here is to quantify the relative risk of one compared to many and predict its impact on the project.

Quantification starts with computing the project's exposure to the identified risks through the calculation of the risk exposure factor. Probability used in conjunction with decision trees provides a mechanism for quantifying risk of multiple alternatives. For example, if there is a $100,000 bonus for being early with an aggressive schedule (only 18% chance of attainment), but a $250,000 penalty for being late with any schedule (being conservative gives a 90% chance of being on-time or early), should we pursue an aggressive or conservative schedule?

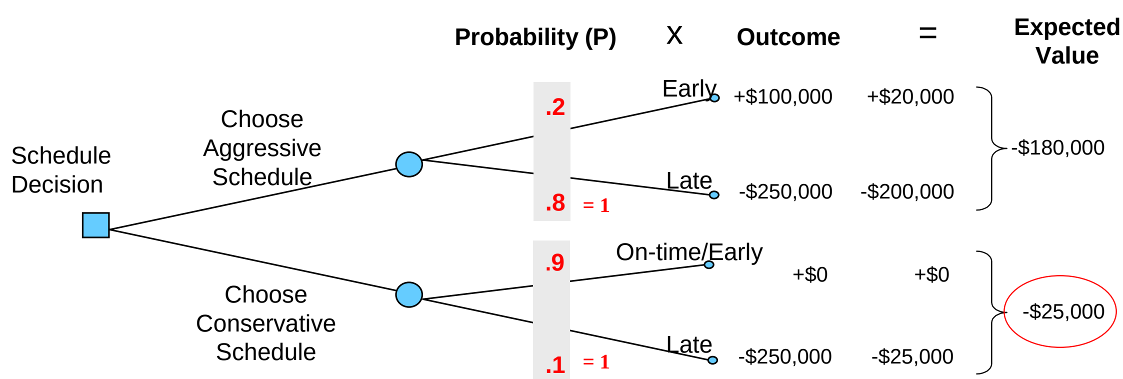

Figure 5. Decision Tree Example

The decision tree example, Figure 5 shows that by choosing an aggressive schedule the potential for risk is a loss of $180,000 while the conservative schedule shows a loss of only $25,000. In this situation, the project manager needs to work on reducing the risk further on the conservative schedule

Developing and Controlling Risks

Here are examples of key engineering development risks and treatments:

1. Unrealistic budget and schedule
   - track all estimates and actuals; understand the team's performance level
   - understand where all team member's time really goes because there are always

overhead activities in any organization
- don't allow the client to talk you into an unrealistic estimate

2. Personnel shortfalls
   - plan for training in areas needed for the project
   - establish a learning pattern for team members throughout the project's life
   - cultivate teaming relationships with knowledgeable parties
3. Developing wrong capabilities
   - insist on meeting with the customer
   - prototype and demonstrate planned approaches

The project risk management plan will contain all the identified risks and mitigation plans where appropriate. The risk response development can handle identified risks in three ways:

1. Accept - Do nothing; accept consequences in an active or passive fashion.
2. Transfer - Move the loss to a third party through a contract, get a warranty or buy insurance.
3. Mitigate - Reduce the impact or probability by using contingency planning or a reserve or eliminate the cause by using alternative software development strategies.

Prepare appropriate responses for each risk item by answering these questions:

1. Who is responsible for action?
2. When the action is due?
3. What is the metric to watch?
4. What is the metric trigger value?

ID	Risk Item	Trigger	Value	Risk Exp.	Resolution Approach	Who	Date
1	Too few Engr experts	10	12	630	Contract now for more	PM	1/15
2	Design schedule tight	25	28	450	Enforce Delphi estimates	PM	ongoing
3	Report function weak	20	25	180	Review with customer	PLdr	2/15
4	Interface too different	10	20	150	Review with customer	PLdr	2/15
5	New requirements	5	5	150	Review cost each time	PM	ongoing
6	"Goldplating" threat	15	15	120	Hold to Rqmts document	PLdr	ongoing
7	Unknown quality	3	6	60	Get second supplier	PM	2/1
8	Wall unstable	10	6	60	Investigate braces	Engr	2/15
9	Timing problems	5	6	30	Simulate and test	Engr	ongoing
10	New technology risky	5	8	10	Review w/ chief scientist	PLdr	by stage

Table 2. Risk Response Table

Table 2 shows a risk response table for the top ten project risks. Each risk has an identifier and a description. The metric value to watch is shown along with the trigger. For each risk, the value exceeds or is equal to the trigger. This type of table should be reviewed at no less than a weekly basis.

Risk response management requires a regular review of all risks for changes. The top ten risks are reviewed on at least a weekly basis. They may be the same as the risks on the response table, as shown in Table 3. The difference in the two tables is that the probability and loss are shown as the components of the risk exposure.

ID	Risk Item	Prob	Loss	Risk Exp.	Resolution Approach	Who	Date
1	Too few Engr experts	70	9	630	Contract now for more	PM	1/15
2	Design schedule tight	50	9	450	Enforce Delphi estimates	PM	ongoing
3	Report function weak	20	9	180	Review with customer	PLdr	2/15
4	Interface too different	25	6	150	Review with customer	PLdr	2/15
5	New requirements	30	5	150	Review cost each time	PM	ongoing
6	"Goldplating" threat	30	4	120	Hold to Rqmts document	PLdr	ongoing
7	Unknown quality	10	6	60	Get second supplier	PM	2/1
8	Wall unstable	10	6	60	Investigate braces	Engr	2/15
9	Timing problems	5	6	30	Simulate and test	Engr	ongoing
10	New technology risky	5	2	10	Review w/ chief scientist	PLdr	by stage

Table 3. Top Ten Software Project Risks

Risk Categories

The project risk management plan models 12 categories of potential risk to any specific project:

1. Mission and Goals – any project accepted must fit within the organization's mission and goals. Projects accepted that do not fit within the organization create tensions that effect all projects. For example, assume an organization exists whose mission is to develop software for internal corporate manufacturing and whose goal is to produce the most effective, custom software for the organization's factories. If they were to accept a project to build a general-purpose software package to be sold commercially, this would be extremely risky because it goes against their current mission and goals.
2. Organization Management – any project chosen must be buildable within the current or planned organization. A disorganized or non-existent organization cannot succeed in delivering a software project. An example of this risk is a sales organization that closes a development project with no input from the executing organization. The project is "thrown over the wall" to a development organization that has no team available and no process for building the type of system sold.
3. Customer – all projects must have a strong customer commitment to its success. A software development project requires extensive input from the customers and end users. Without this input, the best development process will only produce a system that works well but may not meet the end users' real needs. The risk here is that the assigns inexperienced people to the development team who do not have adequate problem domain experience to guide the technical trade offs needed for the software developers.
4. Budget/Cost – this category is the one that usually gets the most attention and is affected by all other categories. Project managers focus on the budget and cost because these are the most widely used measurements of a project's success. Understanding project size, having good historic information on similar projects and completely understanding the external influences, such as technology, are the main ways to reduce this category's risk.
5. Schedule – the greatest risk here is that schedule dates are imposed externally from the development team. If the development team does not have any input into the completion and delivery dates for the project, there is very little chance that the schedule will be met. Software development teams must be part of developing and modifying the project schedules.
6. Project Content – all projects generate artifacts that are in addition to the final,

contracted for deliverables. One of the major components is documentation of requirements, design and the target system in which the software will reside. If this information does not exist, is in error or inconsistent the risk is very high that project knowledge will be lost and the schedule or product content will greatly suffer.

7. Performance – these risk factors are related not to specific, delivered system execution times but to key software development criteria. Some of the major risk areas here are related to the performance of the system during testing. The ability to do complete coverage testing of all modules and their interfaces is critical. Inadequate testing is a contributor to project failure.
8. Project Management – category relates to both the management processes for the project and the manager of the project. Risk exists not only in the lack of or inadequacy of management processes but in the experience level of the project manager. It is not true that a good project manager can manage any project. Project managers need domain experience and understanding of project management processes.
9. Development Process – this category is focused on processes that reduce overall risk and improve delivered product quality. Development processes are not concerned with specific tools such as programming languages, tool builders or code generators. It is focused on configuration management processes, quality assurance practices and analyses of alternatives.
10. Development Environment – focuses on the physical environment of facilities, hardware platforms and software development tools. Risk is present in not only the lack of adequate tools but in inadequate facilities. Not having a co-located team, or not having adequate meeting space, customer interviewing space and workrooms greatly increases the risk. Teams need face-to-face contact on a regular basis.
11. Staff – this category is one area that risk can be greatly reduced by having an experienced and proven high productivity software development team. A highly productive team can be 10 to 25 times more productive than an average team. Not being sure of the abilities of the team or their experience with the problem domain, necessitates a very conservative approach to the risk factors in this category.
12. Maintenance – this final category attempts to quantify software risk after the product is delivered. The project development team, many times, is responsible for maintenance of the software for some period of time after delivery. If this is not the case, the project risk increases due to having inexperienced people trying to fix bugs in the software. Tools used for development need to be available for maintenance. Vendor support after delivery is a risk issue if there has been no plan or budget for continued tool maintenance support.

Steps in developing a risk management plan

Step 1:

Using these categories, construct a risk categorization table. A project team might use this table to review the categories of risk for their project. This table provides the team a set of factors to consider and provides slots for them to decide which are relevant and what evidence they have. As the organization learns more about its performance, it may decide on ways to compare ratings on a given project with its prior history. It may determine a total rating count or number of risks or some combination of number and level of impact that predict project failure or success. This table is a starting point for identification of specific risks on each project.

Step 2:

Rank the risk to the project for each category:

- Risk Factors and Areas - Under each category, this column lists category risk factors.
- L - Low Risk Evidence - This column has characteristics of this factor when it can be considered low risk to the project.
- M - Medium Risk Evidence - This column has characteristics of this factor when it provides a medium risk.
- H - High Risk Evidence - This column has characteristics of this factor when it should be considered high risk.
- Rating - Select the level of risk (example: H, M, L or 3, 2, 1) applicable to this project.
- Comments - Provide information about project specifics that support the rating choice

Note that in some cases, evidence in one category for high risk may be evidence for low risk in another. For example, support for organization goals or use of new technologies may be taken either way, depending on the situation.

Table 4. Risk Categorization Table

Risk Factors and Categories	**L -Low Risk Evidence**	**M - Medium Risk Evidence**	**H - High Risk Evidence**	**Rating (HML)**	**Comments**
Mission and Goals Factors					
Project Fit	directly supports organization mission and/or goals	indirectly impacts one or more goals	does not support or relate to organization mission or goals		
Work Flow	little or no change to work flow	will change some aspect or have small affect on work flow	significantly changes the work flow or method of organization		
Organization Management Factors					
Organization Stability	little or no change in management or structure expected	some management change or reorganization expected	management or organization structure is continually or rapidly changing		
Development Team Stability	Team chosen, little or no change expected.	team chosen but members may change.	Team not chosen, no decision as to members		
Policies and Standards	development policies and standards are defined and carefully followed	development policies/ standards are in place, but are weak or not carefully followed	no policies or standards, or they are ill-defined and unused		
Management Support	strongly committed to success of project	some commitment, not total	little or no support		
Performance Objectives	verifiable performance objectives, reasonable requirements	some performance objectives, measures may be questionable	no established performance requirements or requirements are not measurable		
Executive Involvement	visible and strong support	occasional support, provides help on issues when asked	no visible support; no help on unresolved issues		
Customer Factors					

Risk Factors and Categories	L -Low Risk Evidence	M - Medium Risk Evidence	H - High Risk Evidence	Rating (HML)	Comments
Customer Involvement	end users highly involved with project team, provide significant input	end users play minor roles, moderate impact on system	minimal or no end user involvement; little end user input		
Customer Experience	end users highly experienced in similar projects; have specific ideas of how needs can be met	end users have experience with similar projects and have needs in mind	end users have no previous experience with similar projects; unsure of how needs can be met		
Customer Acceptance	end users accept concepts and details of system; process is in place for end user approvals	end users accept most of concepts and details of system; process in place for end user approvals	end users do not accept any concepts or design details of system		
Customer Training Needs	end user training needs considered; training in progress or plan in place	end user training needs considered; no training yet or training plan is in development	requirements not identified or not addressed		
Customer Justification	end user justification complete, accurate, sound	end user justification provided, complete with some questions about applicability	no satisfactory justification for system		
Contract Fit	contract with customer has good terms, communication with team is good	contract has some open issues which could interrupt team work efforts	contract has burdensome document requirements or causes extra work to comply		
Benefits Defined	benefits well-defined, with identified measures and baselines	some questions remain about benefits, or baseline is changing and measures doubtful	benefits not defined, no baseline established, unattainable or un-measurable		
Budget/Cost Factors					
Project Size	small, non-complex, or easily decomposed	medium, moderate complexity, decomposable	large, highly complex, or not decomposable		
Hardware Constraints	little or no hardware-imposed constraints or single platform	some hardware-imposed constraints; several platforms	significant hardware-imposed constraints; multiple platforms		
Technology	mature, existent, in-house experience	existent, some in-house experience	new technology or a new use or under development; little in-house experience		
Reusable Components	components available and compatible with approach	components promised, delivery dates unsure	components projected, but not available when needed		
Supplied Components	components available and directly usable	components work under most circumstances	components known to fail in certain cases, likely to be late, or		

Risk Factors and Categories	**L -Low Risk Evidence**	**M - Medium Risk Evidence**	**H - High Risk Evidence**	**Rating (HML)**	**Comments**
			incompatible w/ parts of approach		
Budget Size	sufficient budget allocated	questionable budget allocated	doubtful budget is sufficient		
Budget Constraints	funds allocated without constraints	some questions about availability of funds	allocation in doubt or subject to change without notice		
Economic Justification	completely justified and cost effectiveness proven	justification questionable or effectiveness not completely established	not justified or cost effectiveness demonstrated		
Cost Controls	well established, in place	system in place, weak in areas	system lacking or nonexistent		
Schedule Factors					
Delivery Commitment	stable commitment dates	some uncertain commitments	unstable, fluctuating commitments		
Development Schedule	team projects that schedule is acceptable and can be met	team finds one phase of the plan to have a schedule that is too aggressive	team projects that two or more phases of schedule are unlikely to be met		
Project Content					
Requirements Stability	little or no change expected to approved set (baseline)	some change expected against approved set	rapidly changing or no agreed-upon baseline		
Requirements Complete and Clear	all completely specified and clearly written	some requirements incomplete or unclear	some requirements only in the head of the customer		
System Testability	system requirements easy to test, plans underway	parts of system hard to test, or minimal planning being done	most of system hard to test, or no test plans being made		
Design Difficulty	well defined interfaces; design well understood	unclear how to design, or aspects of design yet to be decided	interfaces not well defined or controlled; subject to change		
Implementation Difficulty	algorithms and design are reasonable for this team to implement	algorithms and/or design have elements somewhat difficult for this team to implement	algorithms and/or design have components this team will find very difficult to implement		
System Dependencies	clearly defined dependencies of the software effort and other parts of system	some elements of the system are well understood and planned; others are not yet comprehended	no clear plan or schedule for how the whole system will come together		
Documents Stability	documents will be available on time and will contain few errors	some documents may be late and contain minor errors	little chance of getting documents on time, many corrections and changes		

Risk Factors and Categories	L -Low Risk Evidence	M - Medium Risk Evidence	H - High Risk Evidence	Rating (HML)	Comments
			expected		
Performance Factors					
Test Capability	modular design allows for easy coverage test planning and execution	modular design aids developing test harnesses for unit test	No modular design or ability to easily establish test coverage planning.		
Expected Test Effort	good estimate available, readily fits system acceptance process	rough estimate of test time, may be a bottleneck in the process	poor or no estimate of test times, definite chance of bottleneck		
Functionality	highly functional, meets all customer needs	good functionality, meets most customer needs	little functionality, many customer needs not met		
External Hardware or Software Interfaces	little or no integration or interfaces needed	some integration or interfaces needed	extensive interfaces required		
Project Management Factors					
Approach	product and process planning and monitoring in place	planning and monitoring need enhancement	weak or nonexistent planning and monitoring		
Communication	clearly communicates goals and status between the team and rest of organization	communicates some of the information some of the time	rarely communicates clearly to the team or to others who need to be informed of team status		
Project Manager Experience	project manager very experienced with similar projects	project manager has moderate experience or has experience with different types of projects	project manager has no experience with this type of project or is new to project management		
Project Manager Attitude	strongly committed to success	willing to do what it takes	cares very little about project		
Project Manager Authority/ Support	complete support of team and of management	support of most of team, with some reservations	no visible support; manager in name only		
Development Process Factors					
Alternatives Analysis	analysis of alternatives complete, all considered, assumptions verifiable	analysis of alternatives complete, some assumptions questionable or alternatives not fully considered	analysis not completed, not all alternatives considered, or assumptions faulty		
Quality Assurance Approach	QA system established, followed, effective	procedures established, but not well followed or effective	no QA process or established procedures		
Commitment Process	changes to commitments in scope, content, schedule are reviewed and approved by all	changes to commitments are communicated to all involved	changes to commitments are made without review or involvement of the team		

Risk Factors and Categories	**L -Low Risk Evidence**	**M - Medium Risk Evidence**	**H - High Risk Evidence**	**Rating (HML)**	**Comments**
	involved				
Development Documentation	correct and available	some deficiencies, available	nonexistent		
Use of Defined Engineering Process	development process in place, established, effective, followed by team	process established, but not followed or is ineffective	no formal process used		
Early Identification of Defects	peer reviews are incorporated throughout	peer reviews are used sporadically	team expects to find all defects with testing		
Change Control for Work Products	formal change control process in place, followed, effective	change control process in place, not followed or is ineffective	no change control process used		
Defect Tracking	defect tracking defined, consistent, effective	defect tracking process defined, but inconsistently used	no process in place to track defects		
Development Environment Factors					
Physical Facilities	little or no modification needed	some modifications needed; some existent	major modifications needed, or facilities nonexistent		
Hardware Platform	stable, no changes expected, capacity is sufficient	some changes under evolution, but controlled	platform under development along with software		
Tools Availability	in place, documented, validated	available, validated, some development needed (or minimal documentation)	invalidated, proprietary or major development needed; no documentation		
Configuration Management	fully controlled	some controls in place	no controls in place		
Security	all areas following security guidelines; data backed up; disaster recovery system in place; procedures followed	some security measures in place; backups done; disaster recovery considered, but procedures lacking or not followed	no security measures in place; backup lacking; disaster recovery not considered		
Vendor Support	complete support at reasonable price & in needed time frame	adequate support at contracted price, reasonable response time	little or no support, high cost, and/or poor response time		
Staff Factors					
Staff Availability	in place, little turnover expected; few interrupts for fire fighting	available, some turnover expected; some fire fighting	high turnover, not available; team spends most of time fighting fires		

Risk Factors and Categories	L -Low Risk Evidence	M - Medium Risk Evidence	H - High Risk Evidence	Rating (HML)	Comments
Mix of Staff Skills	good mix of disciplines	some disciplines inadequately represented	some disciplines not represented at all		
Product Knowledge	very experienced at testing this type of product	some experience in testing this type of product	no experience in testing this type of product		
Software Development Experience	extensive experience with this type of project	some experience with similar projects	little or no experience with similar projects		
Training of Team	training plan in place, training ongoing	training for some areas not available or training planned for future	no training plan or training not readily available		
Team Spirit and Attitude	strongly committed to success of project; cooperative	willing to do what it takes to get the job done	little or no commitment to the project; not a cohesive team		
Team Productivity	all milestones met, deliverables on time, productivity high	milestones met, some delays in deliverables, productivity acceptable	productivity low, milestones not met, delays in deliverables		
Maintenance Factors					
Complexity	structurally maintainable (low complexity measured or projected)	certain aspects difficult to maintain (medium complexity)	extremely difficult to maintain (high complexity)		
Change Implementation	team in place can be responsive to customer needs	team experiences delays, but acceptable to customer	team is unable to respond to customer needs		
Support Personnel	in place, experienced, sufficient in number	missing some areas of expertise	significant discipline or expertise missing		
Vendor Support	complete support at reasonable price and in needed time frame	adequate support at contracted price, reasonable response time	little or no support, high cost, and/or poor response time		

Step 3:

Sort the risk table in order of risk with high risk items first. For the top ten risks, and all risks rated high if more than ten, calculate the risk exposure. These are your key risks. Identify means of controlling each key risk, establish ownership of the action and date of completion. Integrate the key risks into project plan and determine the impacts on schedule and cost.

Step 4:

Establish a regular risk report format for weekly project status meetings. At a minimum, show status of top ten (Table 3), with ranking of each from previous week and number of weeks on the list. Show the risk response report (Table 2) and the risk change report. Table 5 shows this report with the changing in rankings and the resolution progress.

Risk Item	Rank this week	Last rank	# of weeks on list	Resolution Progress
Too few Engr experts	1	1	2	Contract under discussion
Design schedule tight	2	2	2	Enforcing Delphi estimates
Report function weak	3	5	3	On agenda with customer
Interface too different	4	4	3	On agenda with customer
New requirements	5	3	4	Review each new one for cost
"Goldplating" threat	6	6	4	Reviewing each phase
Unknown quality	7	8	3	No second supplier found yet
Wall unstable	8	new		Contract for braces in process
Timing problems	9	new		Plan to simulate in March
New technology risky	10	10	4	Reviewed requirements

Table 5. Weekly Risk Change Report

Step 5

The final step is insuring that risk management is an ongoing process within your project management. Monitoring and control must be done to the risk list on a regular basis. The project manager and team must be aware of the identified risks and the process for resolving them. New risks must be identified as soon as possible, prioritized and added on to the risk management plan. High priority risks must be worked with respect to the overall project plan.

Endnotes

---

[1] A Guide to The Project Management Body of Knowledge, The Project Management Institute, Inc. (PMI®), http://www.pmi.org