SOFTWARE IS A RISKY BUSINESS

Linda Westfall
The Westfall Team
lwestfall@westfallteam.com
3000 Custer Road, Suite 270, PMB 101
Plano, TX 75075
972-867-1172 (voice)
972-943-1484 (fax)
www.westfallteam.com

There are many risks involved in creating high quality software on time and within budget. With ever-increasing software complexity and increasing demand for bigger, better, and faster product, the software industry is a high-risk business. When teams don't manage risk, they leave projects vulnerable to factors that can cause major rework, major cost or schedule over-runs, or complete project failure. Adopting software risk management processes is a step the can help effectively manage software development and maintenance initiatives. However, in order for it to be worthwhile to take on these risks, the organization must be compensated with a perceived reward. The greater the risk, the greater the reward must be to make it worthwhile to take the chance. In software development, the possibility of reward is high, but so is the potential for disaster. Risk exists whether it is acknowledged or not. People can stick their heads in the sand and ignore the risks but this can lead to unpleasant surprises when some of those risks turn into actual problems. The need for software risk management is illustrated in Gilb's risk principle. "If you don't actively attack the risks, they will actively attack you" [Gilb-88]. In order to successfully manage a software project and reap the rewards, software practitioners must learn to identify, analyze, and control these risks. This paper focuses on the basic concepts, processes, and techniques of software risk management.

RISK/OPPORTUNITY BALANCE

In the software industry the future seems to be coming at us at an ever-increasing rate. Effective software managers and practitioners proactively think about all the possibilities that the future may bring, but those possibilities have uncertain outcomes. We call those possibilities opportunities if we believe they may have positive outcomes. For example, we may have the opportunity to successfully complete a software project and make a substantial profit or we may have an opportunity to introduce a new product into the marketplace first and capture the lion's share of the market. We call the possibilities risks if we believe they may have negative outcomes. For example, we have the risk of not successfully completing that same software project and losing our investment or we may have the risk of our competition beating us to the marketplace with a new product and losing market share. To quote Tom DeMarco, "Moving aggressively after opportunity means running toward rather than away from risk." [Hall-98]

As illustrated in Figure 1, good risk management practices are a balancing act between the risk and the reward. While this paper focuses on risk management, the associated opportunity (reward) also needs to be identifying and managed. Not paying attention to opportunities and balancing opportunity management along with risk management can lead to the loss of important opportunities.

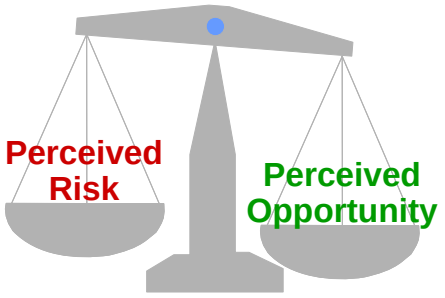

Figure 1: Risk / Opportunity Balance

Different people and different organizations have different risk tolerance levels. A person or organization's risk tolerance influences their perceived risk/reward balance point. For risk takers the sheer pleasure of taking the risk adds weight to the reward side of the risk/reward balance. Risk takers are more willing to take a risk even if the financial, economic or material gains are less than the loss associated with the potential problem. Risk avoiders are averse to taking risks and the mere presence of the risk adds weight to the risk side of the risk/reward balance. Risk avoiders need additional financial, economic or material incentives to take on the risk. People or organizations that are risk neutral look for at least a balance between the financial, economic or material risks and rewards. They have no emotional investment in either avoiding or taking the risk.

DEFINING RISK

So, what are risks? A risk is simply the possibility of a problem occurring some time in the future. According to the Project Management Institute [PMI-08], "Risk is an uncertain event or condition that, if it occurs, has an effect on at least one project objective. Objectives can include scope, schedules, cost and quality." "Risk, like status, is relative to a specific goal. Whereas, status is a measure of progress toward a goal, risk is a measure of the probability and consequences of not achieving the goal." [Hall-98] For example, every time a person crosses the street, that person runs the risk of being hit by a car. As illustrated in Figure 2, a risk starts when the commitment associated with that risk is made, the risk of getting hit by a car in the street does not exist until the person steps into the street. Of course there is the risk of getting hit by a car while standing on the sidewalk, but that is an entirely different risk. A project doesn't have the risks of ACME delivering a low reliability software component or of ACME being late in delivering that software component until the project selects ACME as its subcontractor for that component. If the project chooses to build the software component in-house, a different set of risks exist. A risk ends when one of two things happen:

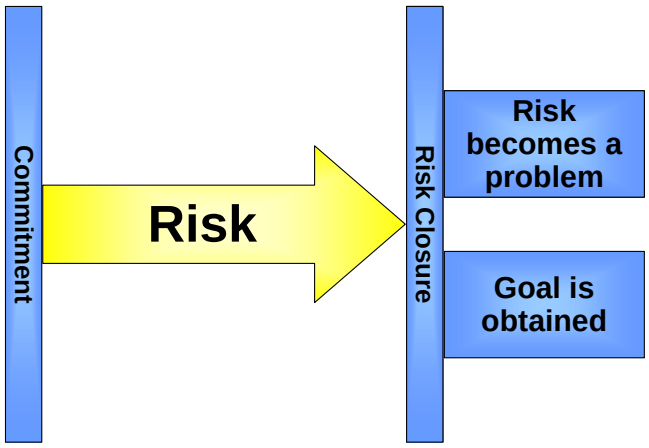

Figure 2: Risk Duration

1. Bang!! The person gets hit in the middle of the street by the car. The problem actually occurs. It is now a problem and is no longer a risk.
2. Or the person safely steps onto the sidewalk on the other side of the street. The risk disappears because there is no longer the possibility of a future problem because the goal has been obtained.

ACME either delivers a high quality product component on time (goal is reached) or they don't (problem occurs). Before it turns into an actual problem, "a risk is just an abstraction. It's something that may affect your project, but it also may not. There is a possibility that ignoring it will not come back to bite you." [DeMarco-03] If a person ignores the risk and runs into the street without looking, there can be dire consequences. There may not be a problem every time -- in fact that person can get away with it over and over again. But then it only takes one instance of the problem occurring for disaster to happen.

RISK MANAGEMENT PROCESS

Risk management is an ongoing process that is implemented as part of the initial project planning activities. Risks should be taken into consideration when estimates are made of the initial project effort, schedule and budget. The risk management process must also be an on-going part of managing the software development project. In fact, DeMarco and Lister [DeMarco-03] call risk management "project management for adults". Risk management is designed to be a continuous feedback loop where additional information, including risk status and project status, are utilized to refine the project's risk list and risk management plans.

The risk management process is illustrated in Figure 3. This process starts with the identification of a list of potential risks. Each of these risks is then analyzed and prioritized. A risk management plan is created to identify risk mitigation actions for high priority risks. Risk management plans can also include contingency actions that will be taken only if the associated risk triggers indicate that the risk is turning into a problem or the problem actually occurs. The mitigation part of the plan is then implemented and the planned risk reduction actions are taken. The tracking step involves monitoring the status of known risks, as well as the results of risk reduction actions and other project activities. As new status and information are obtained, additional risk analysis is done and/or the risk management plans are updated accordingly. Tracking may also result in the addition of newly identified risks or in the closure of known risks. If a trigger indicates that is risk is turning into a problem, the corresponding contingency plans are implemented.

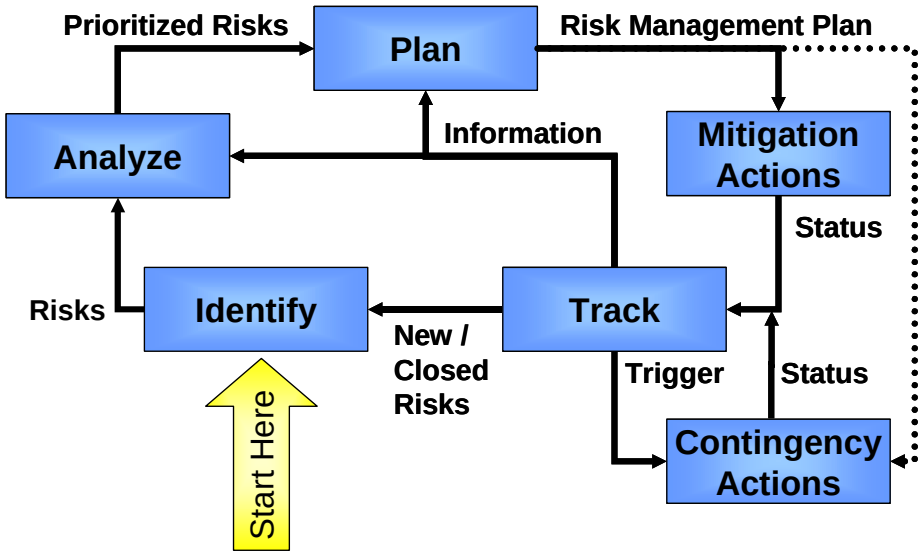

Figure 3: Risk Management Process

RISK IDENTIFICATION

During the first step in the software risk management process, risks are identified and added to the list of known risks. Risk identification requires a fear-free environment where risks can be identified and discussed openly. The output of this step is a list of project-specific risks that have the potential of compromising the project's success. The project team should be as thorough as possible on the first round of risk identification but not obsessive. It's probably impossible to identify all of the project risks on the initial pass through the risk management process. The team just doesn't have enough information yet. There are technical requirements yet to elicit, staffing issues yet to decide, design decisions yet to be made and all kinds of commitments yet to be made. The risk identification step will need to be revisited repeatedly throughout the project as more information is obtained from project execution, tracking and control.

There are many techniques for identifying risks, including interviewing/brainstorming, reporting, product decomposition, project decomposition, assumption analysis, and utilization of risk taxonomies.

Interviewing/Brainstorming: One technique for identifying risks is interviewing or brainstorming with project personnel, customers, users and suppliers. Since stakeholders at different levels inside and outside the development organization have different perspectives on the project, one goal of risk identification is to involve a variety of stakeholders in the risk process in order to obtain a broader more complete perspective of the project's risks. Using open-ended questions during interviewing/brainstorming, such as those in the following list, can help identify potential areas of risk.

- What problems do you see in the future for this project?

- Are there areas of this project that you feel are poorly defined?
- What interface issues still need to be defined?
- What requirements exist that the team isn't sure how to implement?
- What concerns does the team have about their ability to meet the required quality levels? Performance levels? Reliability levels? Security levels? Safety levels?
- What tools or techniques might this project require that it doesn't have?
- What new or improved technologies does this project require? Does the project team have the expertise to implement those technologies?
- What difficulties do you see in working with this customer? Sub-contractor? Partner?

Voluntary Reporting: Another risk identification technique is voluntary reporting, where any individual who identifies a risk is encouraged to bring that risk to management's attention. This requires the complete elimination of the "shoot the messenger" syndrome. It avoids the temptation to assign risk reduction actions to the person who identified the risk. Risks can also be identified through required reporting mechanisms such as status reports or project reviews.

Product Decomposition: As the product is being decomposed during the requirements and design activities, another opportunity exists for risk identifications. Every TBD ("To Be Done/Determined") is a potential risk. As Ould states, "The most important thing about planning is writing down what you ***don't know***, because what you don't know is what you must find out" [Ould-90]. Feasibility or lack of stability in areas of the requirements or design can signal areas of risk. A requirement or design element may also be risky if it requires the use of new and/or innovative technologies, techniques, languages and/or hardware. However, even if the technologies, techniques, languages and/or hardware have been around in the industry for a while, there still may be a risk if this is the first time this organization has attempted to utilize them. For example, JAVA has been around for a while, but if this is the first time this organization has used JAVA on a project, there may be risks because of lack of expertise that might affect the quality of the end product or impact the schedule because of a learning curve. Remember a risk starts when a commitment is made. As the software requirements are being defined (or being allocated from the system level requirements), the project is committing to what is going to be developed. As the software is being designed, the project is committing to choices about how the software is going to be developed. As the project makes these commitments the team needs to keep asking themselves – "What risks are associated with the commitment to meet this requirement or implement this design element?" -- in order the help identify the associated risks.

Project Decomposition: Decomposition can also come in the form of work breakdown structures during project planning, which can also help identify areas of uncertainty for specific sub-projects, tasks or activities that may need to be recorded as risks. Are there any feasibility, staffing, training or resource issues associated with each identified activity? When using this technique, the project team should be particularly alert for risks as critical path analysis is performed on the project schedule. Any possibility of schedule slippage on the critical path must be considered a risk because it directly impacts the ability to meet schedule.

Assumption Analysis: In this risk identification technique an analysis of process, product or planning assumptions is performed. Example assumptions might include the assumption that hardware will be available by the system test date or that three additional experienced C++ programmers will be hired by the time coding starts. If these assumptions prove to be false, what potential problems might occur? In other words, what are the risks associated with each assumption. If there are not any risks associated with an assumption, then it may not be a real assumption.

Risk Taxonomies: If any experienced software person on the project is asked, what will go wrong on this project, they will be able to answer with uncanny accuracy. Why? -- Because it is what went wrong on the last project and the one before that and the one before that. The problems from previous projects are one of the best indicators of the risks on new or current projects. This is why risk taxonomies are such a great tool. Risk taxonomies are lists of problems that have occurred on other projects and can be used

as checklists to help ensure all potential risks have been considered. Risk taxonomies can also be used during the interview process to help develop interview questions. Examples of risk taxonomies include:

- Software Engineering Institute's Taxonomy-Based Risk Identification report that covers 13 major risk areas with about 200 questions [SEI-93]
- Capers Jones's entire book, Assessment and Control of Software Risks, could be viewed as risk taxonomy. [Jones-94]
- Steve McConnell's book, Rapid Development: Taming Wild Software Schedule also includes an extensive list of what he labels as "Potential Schedule Risks." [McConnell-96]

When an organization is just starting their risk management efforts, they can start with taxonomies such as these from the literature. These industry taxonomies should then be evolved and tailored over time to match the actual problem types encountered by that organization. One last word of caution -- when using risk taxonomies, a delicate balance must be maintained between making sure that known issues are handled and focusing so much on the items from the list that new and novel risks are missed.

RISK STATEMENT

Once a risk is identified it should be communicated to everyone who needs to know about it. This includes management, people who could be affected if the risk became a problem, people who will analyze the risk, people doing risk planning and people who will take action to mitigate the risk. It may also need to be communicated to customers, suppliers or other stakeholders. Verbal communications allow discussion of the risk that can help clarify the understanding of the risk. The listener has a chance to ask questions and interact with the person communicating the risk. This two-way interaction may result in additional information about the risk, its sources and its consequences. Written communications result in historical records that can be referred to in the future. Everyone who received the written communication has the identical information about the risk. Written communications can also allow for easy dissemination of the risk information if the people who need the information are in multiple locations. The creation of an online risk database may provide a consistent and easily accessed mechanism to providing written risk information.

A written risk statement consists of the risk condition and its potential consequences for the project. The condition is a brief statement of the potential problem that "describes the key circumstances, situation, etc. causing concern, doubt, anxiety, or uncertainty." [Dorofee-96] The consequence is a brief statement that describes immediate loss or negative outcome if that condition turns into an actual problem for the project. Figure 4 includes two example risk statements.

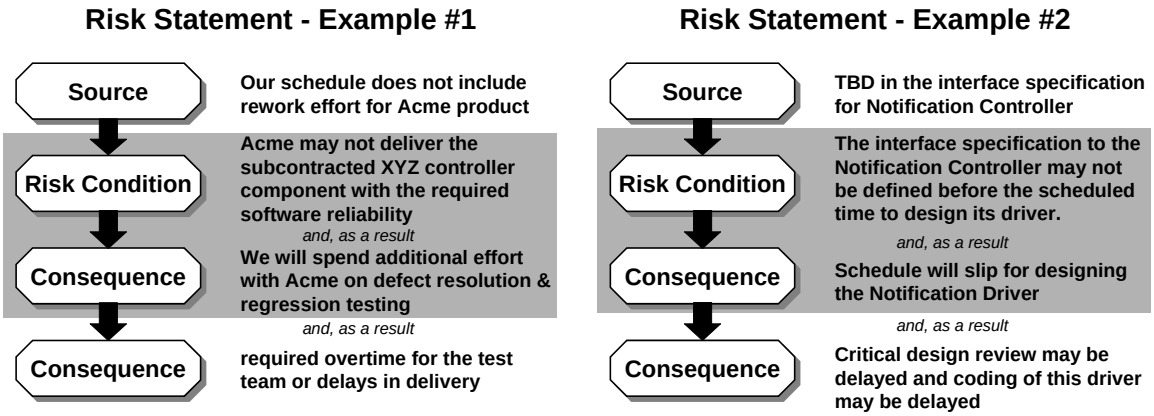

Figure 4: Risk Statement Examples

In example #1, a software quality engineer (SQE) reviewed the subcontractor portion of the project plan and performed an assumption analysis. The SQE noticed that the project was assuming that Acme would deliver software of the required reliability and that no provisions had been made in the schedule to deal with major defects found in the XYZ controller. For example #1 the risk statement (highlighted in

grey) would be: *Acme may not deliver the subcontracted XYZ controller component with the required software reliability and, as a result the project will spend additional effort with Acme on defect resolution and regression testing.*

For example #2, during the requirements inspection process the inspection team noticed a TDB in the interface specification requirements for the Notification Controller. This controller is an external piece of equipment that the system must interface with to send delinquent notices as part of the billing system. For example #2 the risk statement would be: *The interface specification to the Notification Controller may not be defined before the scheduled time to design its driver and, as a result the schedule will slip for designing the Notification Driver.* Note that in these examples, the source of the risk is not part of the risk statement and neither is any subsequent consequences to the initial consequences.

RISK ANALYSIS

The primary goal of the risk analysis step of the risk management process is to analyze the identified list of risks and prioritize those risks for further planning and action. During the risk analysis step, each risk is assessed to determine its context, estimated probability, estimated loss and timeframe.

A risk's context includes the events, conditions, constraints, assumptions, circumstances, contributing factors, project interrelationship and related issues that lead to the potential for a problem. The risk's context provides all of the additional information that surrounds and affects the risk and helps determine its probability and potential loss. Documenting the risk's context can be especially useful after time has passed and a risk is being reevaluated. Gaps between the original documented context and the current situation can help the project staff obtain a better understand how or if the risk has changed.

A risk's estimated probability is the likelihood that the risk will turn into a problem. A risk's potential loss is the estimated impact or consequences to the project if the risk does turn into a problem. Losses can come in the form of additional costs (dollars or effort), required changes to the schedule or technical effects on the product being produced (for example its functionality, performance or quality). Losses can also result from other types of impacts. For example, the organization could lose corporate goodwill, market share or employee satisfaction. Not only should each risk be assessed individually, but the interrelationships between risks must also be assessed to determine if compounding risk conditions exist that magnify losses.

A risk's timeframes are when the risk needs to be addressed and when the risk may turn into a problem. A risk associated with activities in the near future may have higher priority than similar risks associated with later activities even if it has a lower risk exposure.

The level of formal risk assessment needed for a project can range from the simple qualitative assignment of each risk to a category (for example, high, medium or low) to the use of quantitative mathematical modeling (for example, Monte Carlo modeling). The project manager and/or project team should use the simplest method available that allows them to make reliable risk planning decisions. Different risks may be assessed at different levels of formality. For example, a high impact risk with a high probability may require very formal and detailed analysis to determine the appropriate mitigation plans. However knowing that a risk is unlikely to turn into a problem and that it will have very little impact if it does may be all the team needs to know about that risk.

Boehm [Boehm-89] defines a risk exposure equation to help quantitatively establish risk priorities. Risk exposure measures the impact of a risk in terms of its expected value. Risk exposure (RE) is defined as the probability of an undesired outcome (the problem actual occurs) times the expected loss (cost of the impact or consequences) if that outcome occurs.

RE = Probability (UO) * Loss (UO), where UO = unexpected outcome

For example, if a risk is estimated to have a 10% chance of turning into a problem with an estimated impact of $100,000, then the risk exposure for that risk is 10% x $100,000 = $10,000. Comparing the risk exposure measurement for various risks can help identify those risks with the greatest probable negative impact to the project and thus help establish which risks are candidates for further action.

The analysis step in the risk management process is used to prioritize the list of risks. Risks can be prioritized using just their risk exposures or by using a combination of their risk exposures and

timeframes. When a risks need to be prioritized on multiple criteria (risk exposures), a prioritization matrix can be used such as the one in the first table below. In this example, a risk exposure score of 1 to 5 (5 being the highest) is used (as illustrated in the second table below).

	Criteria and Weights				Risk Exposure
	Technical Exposure (.25)	Cost Exposure (.15)	Schedule Exposure (.20)	Customer Satisfaction (.40)	
Risk 1	1	3	2	4	2.7
Risk 2	4	1	4	3	3.15
Risk 3	2	2	2	1	1.6
Risk 4	2	4	3	2	2.5

Exposure Score	Technical	Schedule	Cost	Customer Satisfactions
5	Unusable system	> 18 months slip	>10% project budget	Will replace purchase product with competitor's product
4	Unusable function or subsystem	12 – 18 months slip	7%-10% project budget	Unwilling to purchase
3	Major impact to functionality, performance or quality	6-12 months slip	5-7% project budget	Willing to purchase for limited use
2	Minor impact to functionality, performance or quality	3-6 months slip	1% - 5% project budget	Willing to purchase and use but will result in complaints
1	Minimal or no impact	<3 months slip	<1%	Willing to purchase and use (may not recommend)

RISK MANAGEMENT PLANNING

Since resource limitations rarely allow the consideration of all risks, the prioritized list of risks is used to identify the top risks for risk mitigation planning and action. Other risks may simply have tracking mechanisms put in place to monitor them closely. At the lowest priorities, other risks are simply documented for possible future consideration. This prioritized list of risks should be reviewed periodically. Based on changing conditions, additional information, the identification of new risk items or simply timing, the list of the prioritized risks may require periodic updates.

During the planning step of the software risk management process, the appropriate risk handling techniques are selected and alternative risk handling actions are evaluated. Whatever handling options are selected, the associated actions should be planned in advance to proactively manage the project's risks rather than waiting for problems and reacting in a firefighting mode. The resulting risk management plans should then be incorporated into the project plans with assigned staff and resources.

Taking the prioritized risk list as input, plans are developed for the handling actions chosen for each risk. As illustrated in Figure 5, specific questions can be asked to help focus the type of planning required.

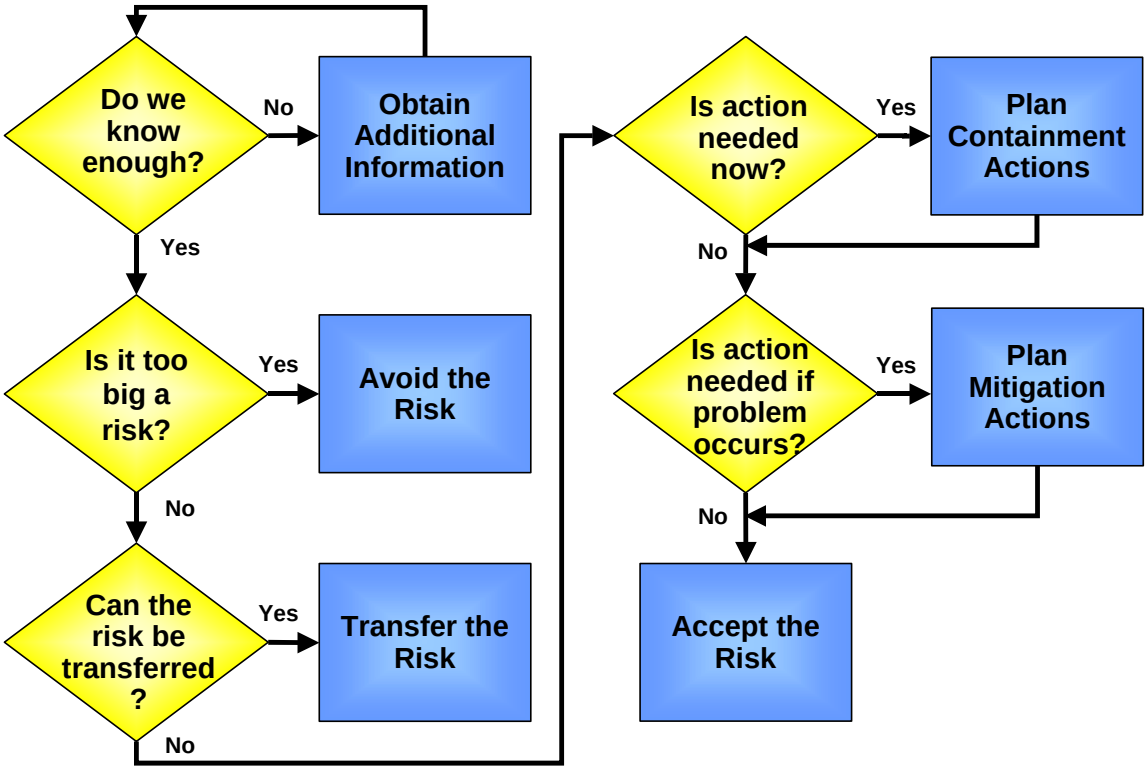

Figure 5: Techniques for Handling Risks

Do we know enough? If we don't know enough, we can plan to "buy" additional information through mechanisms such as prototyping, modeling, simulation, or conducting additional research. Once the additional information has been obtained, the planning step should be revisited. Based on the results of these activities and the information obtained, it may also be appropriate to repeat the risk analysis step for this risk, resulting in changes to its priority.

Is it too big a risk? If the risk is too big for us to be willing to accept, we can avoid the risk by changing our project strategies and tactics to choose a less risky alternate or we may decide not to do the project at all. For example, if our project has tight schedule constraints and includes state of the art technology, we may decide to wait until a future project to implement our newly purchased CASE tools. Once the risk has been successfully avoided, it can be closed. Things to remember about avoiding risks include:

- Avoiding risks may also mean avoiding opportunities
- Not all risks can be avoided
- Avoiding a risk in one part of the project may create risks in other parts of the project

Can we transfer the risk? If it is not this project's risk or if it is economically feasible to pay someone else to assume all or part of the risk, a plan can be developed to transfer the risk to another organization. For example we can contract with a disaster recovery firm to provide backup computer facilities that will allow continuation of the project in case a fire or other disaster destroys the project's work environment. Once the risk has been successfully transferred, it can be closed because it is no longer a project risk. In some cases the project may want to set up a monitoring mechanism to make sure that the people who assumed the risk are appropriately handling it.

Is action needed now? *Is action needed now?* Sometimes the risk cannot be avoided but it is still too big a risk to just accept. If the project decides to attack the risk directly, they typically start with creating a list of possible risk mitigation actions, also called risk containment actions, that can be taken to reduce the risk. Two approaches to risk mitigation plan actions should be considered:

- Actions that reduce the likelihood that the risk will occur
- Actions that reduce the impact of the risk should it occur

These may include actions such as establishing a liaison with the customer to insure adequate communications, conducting a performance simulation, or buying additional equipment for the test bed to duplicate the operational environment.

From the list of possible risk mitigation actions, the project team selects those that are actually going to be implemented. When considering which risk reduction activities to select, a cost/benefit analysis must be performed. Boehm [Boehm-89] defines the risk reduction leverage (RRL) equation to help quantitatively establish the cost/benefit of implementing a risk reduction action. RRL measures the return on investment of the available risk reduction techniques based on expected values. RRL is defined as the difference between the risk exposure (RE) before and after the reduction activity divided by the cost of that activity.

$RRL = (RE_{before} - RE_{after}) / \text{Risk Reduction Cost}$

If the RRL is less than one, it means that the cost of the risk reduction activity outweighs the probable gain from implementing the action.

Is action needed if the problem occurs? If risk mitigation actions are not taken or if those actions reduce but do not eliminate the risk, it may be appropriate to develop risk contingency plans. Contingency plans are plans that are implemented only if the risk actually turns into a problem. One or more risk triggers should be established for each risk with a contingency plan. A trigger is a time or event in the future that acts as an early warning system that the risk is turning into a problem. For example, if there is a risk that outsourced software will not be delivered on schedule, the trigger could be whether the critical design review was held on schedule. A trigger can also be a relative variance or threshold metric. For example, if the risk is the availability of key personnel for the coding phase, the trigger could be a relative variance of more than 10% between actual and planned staffing levels.

There are trade-offs in utilizing triggers in risk management. The trigger needs to be set as early as possible in order to ensure that there is plenty of time to implement risk contingency actions. It also needs to be set as late as possible because the longer the project waits, the more information they have to make a correct decision and not implement unnecessary actions.

Adjusting the project plan: Each selected action in the risk handling plans must include a description of the action and a list of tasks with assigned responsibilities and due dates. These actions must be integrated into the project plan with effort and cost estimations. Project schedules must be adjusted to include these new actions. For example, new tasks such as creating prototypes, doing research or conducting alpha testing at the customer's site must be included in the project plan.

A project is never able to remove all risk -- software is a risky business. A project will therefore typically choose to accept many of its identified risks. The key difference is that a conscious choice has been made from a position of information and analysis rather than an unconscious choice. Even if the project accepts a risk, they may want to put one or more risk triggers in place to warn them that the risk is turning into a problem. Risks that are assigned these triggers can then be set at a "monitor only" priority until the trigger occurs. At that time, the risk analysis step can be repeated to determine if risk reduction action is needed.

TAKING ACTION

During the taking action step, the project implements the risk management plans. Mitigation plans are executed. If risk triggers are activated, analysis is performed and contingency actions are implemented as appropriate. Note that with some luck and good risk mitigation plans, many of a project's contingency plans may never be implemented. Contingency plans are only implemented if the risks turn into problems.

Risk mitigation plans are considered effective if after they are implemented, the resulting risk exposure has been reduced to a level where the project can live with the possible impact if the risk turns into a problem.

TRACKING

Results and impacts of the implementation of risk mitigation and contingency plans must be tracked. The tracking step involves gathering risk data, compiling that data into information, and then reporting and analyzing that information. This includes measuring identified risks and monitoring triggers, as well as measuring the impacts of the implementation of risk management plans. In addition, the implementation of the project itself may alter the context, probabilities, expected losses or timeframes of identified risks. The results of the tracking can be:

- Identification of new risks that need to be added to the risk list
- Validation of known risk resolutions (e.g., risk turned into a problem or the associated goal was met) so risks can be removed from the risk list because they are no longer a threat to project success
- Information that dictates additional analysis or planning requirements
- Implementation of contingency plan

There are two primary mechanisms for tracking risks. The first is the reviews of the risk list and their status by project staff and management. The second is through the use of metrics.

Many of the reviews typically used to manage software projects can also be used to track risks. For example, tracking activities can be included in project team meetings, senior management meetings, and milestone and phase gate review meetings. At the beginning of a process, the entry criteria should be evaluated to determine if the process is truly ready to start. As part of that review, risks associated with that process and its tasks and products could also be evaluated. At the end of a process, the exit criteria should be evaluated to determine if the process is truly complete. This provides another opportunity to review the risks associated with that process, and its tasks and products.

Many of the software metrics typically used to manage software projects can also be used to track risks. For example, Gantt charts, earned value measures, and budget and resource metrics can help identify and track risks involving variances between plans and actual performance. Requirements churn, defect identification rates, and defect backlogs can be used to track other risks including rework risks, risks to the quality of the delivered product, and even schedule risks.

CONCLUSIONS

With ever-increasing complexity and increasing demand for bigger, better, and faster, the software industry is a high-risk business. When teams don't manage risk, they leave projects vulnerable to factors that can cause major rework, major cost or schedule over-runs, or complete project failure. Adopting a proactive software risk management process is a necessary step to more effectively manage software development initiatives. Risk management is an ongoing process that is implemented as part of the initial project planning activities and utilized throughout all of the phases of the software development lifecycle. Risk management requires a fear-free environment where risks can be identified and discussed openly. Based on a positive, proactive approach, risk management can greatly reduce or even eliminate the need for crisis management within our software projects.

REFERENCES

Boehm-89 Barry W. Boehm, *Tutorial: Software Risk Management,* Les Alamitos, CA, IEEE Computer Society, 1989.

DeMarco-03 Tom DeMarco and Timothy Lister, *Waltzing with Bears: Managing Risk on Software Projects*, Dorset House, New York, New York, 2003.

Dorofee-96 Audrey J. Dorofee, Julie A. Walker, Christopher J. Alberts, Ronald P. Higuera, Richard L. Murphy and Ray C. Williams. *Continuous Risk Management Guidebook*, Carnegie Mellon University, Software Engineering Institute, 1996.

Gilb-88 Tom Gilb, *Principles of Software Engineering Management,* Wokingham, England: Addison-Wesley, 1988.

Hall-98 Elaine M. Hall, *Managing Risk: Methods for Software Systems Development*, Addison-Wesley, Reading Massachusetts, 1998

Jones-94 Capers Jones, *Assessment and Control of Software Risks*, Yourdon Press, Prentice Hall, Upper Saddle River, NJ, 1994.

McConnell-96 Steve McConnell, *Rapid Development: Taming Wild Software Schedule*, Microsoft Press, Redmond, WA. 1996.

Ould-90 Martyn Ould, *Strategies For Software Engineering: The Management of Risk and Quality,* Chichester, England, John Wiley & Sons, 1990.

PMI-08 *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) Fourth Edition*, An American National Standard ANSI/PMI 99-001-2008, Project Management Institute (PMI), 2008.

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