

▶ **Gambling with Success: Software Risk Management**

by **Benjamin A. Lieberman, Ph.D.**

Senior Software Architect
Trip Network, Inc.

"Nothing ventured, nothing gained," goes the saying. And as many a visitor to Las Vegas knows, the ideas of Gain and Risk are highly intertwined. In software development, the costs associated with ill-defined project risks can be enormous. Without properly considering these risks, we are doing little more than throwing the dice and hoping for a favorable outcome.

A more mature organization realizes that risk is the price of opportunity,¹ and that risks can be well understood and mitigated. A true "risk" involves some possibility for loss, and "risk acceptance" is a decision to live with the resulting consequences for a given risk. It is the primary purpose of risk analysis to determine which risks have acceptable outcomes -- i.e., outcomes one can live with. For example, the loss of \$1,000 dollars poses less risk to a billionaire than to an impoverished family.

One additional component must be present for some occurrence to qualify as a "risk": the element of choice. If there is no choice about whether to mitigate or avoid the risk, then the possible occurrence is out of one's control and better understood as a "chance," or a so-called "Act of God." The ability to choose which risks are worthwhile (i.e., risks for which the gain justifies the possibility of loss) and which are foolhardy is core to the concept of risk management. The concept of choice indicates there is more than one possible approach available; further, the more choices that are available, the better the likelihood one of those choices will lead to a beneficial outcome.



- ▶ [subscribe](#)
- ▶ [contact us](#)
- ▶ [submit an article](#)
- ▶ [rational.com](#)
- ▶ [issue contents](#)
- ▶ [archives](#)
- ▶ [mission statement](#)
- ▶ [editorial staff](#)

The key to risk management is the identification and mitigation of all true risks or, failing all else, the development of a contingency plan in case the potential risk becomes a concrete reality. In this article I will explore the identification and consideration of risks common to software development, paying particular attention to the effect of a company's level of maturity and existing culture on perception of risk.

Management Maturity and Risk

Because all software development involves human beings, the *perception* of risk plays a great role in the approach for lessening the probability of risk occurrence². Often the "politics and perception" of risk involve ego and pride on the part of individuals -- some of whom are willing to take on a greater risk of loss than may be reasonable based on the level of anticipated gain. Thus, mitigation strategies will differ, based on the level of "corporate maturity."³ For a young, entrepreneurial company, survival is based on taking chances, and risk mitigation is mostly about preventing disastrous losses. For more established firms, what often appears to be most critical for survival includes maintaining the status quo, so that risk mitigation centers around eliminating "disruptive" elements or competition -- i.e., the classic "If it isn't broken, don't fix it" mentality. Finally, for firms that are highly inflexible to change (what Azides refers to as Late-Aristocratic or Bureaucratic⁴), risk management is based on returning to the status quo, which means eliminating or reducing the need for the project in the first place.

Thus, a risk that is considered too extreme for an established firm may be one that a young firm is willing to accept, because it has less to lose and more to gain. In terms of software development, this may include a greater acceptance by the executive management of "bleeding-edge" technologies that have not been firmly established, or a willingness to experiment with development methodologies or development tools. Often it is the development group that proposes these approaches, acting on the mistaken belief that they can meet an unreasonable timeline if only they apply the "right" process. Risk management in this case should focus on risk reduction and early elimination of technical risks; these are usually the least known and most likely to disrupt or derail the project.

In contrast, a well-established firm is primarily interested in maintaining existing customers and gradually adding new ones, and so will focus more on risk transfer strategies (such as outsourcing). This is because transfer approaches allow a company to reduce its overall risk exposure but still retain control over the risk.

Finally, for the hide-bound, bureaucratic firm, it may be a wonder that a project is ever started in the first place. Perhaps a new CEO, recognizing the need for change in order to avoid company failure down the road, initiates a project. If the company is typically hostile to any sort of change, then the primary risk management activity is to continually assess the attitude of executive management. The highest risk is that the change project will be canceled before there is a chance to show value.

Aside from the company's maturity level, there are many other pitfalls

that can affect risk management in any organization. Consider the traps in Table 1.

Table 1: Pitfalls for Risk Management

Pitfall	Description
Out of Sight, Out of Mind	Teams often don't pay sufficient attention to risks that are obvious but not necessarily very visible. They assume the risk is so obvious that it will be dealt with, when in fact the risk may be forgotten until it is too late.
Selective Bias	If the project carries a large number of risks and the development team has limited skills, the team might focus on a small subset of the risks rather than deal with all the risks equally.
Expertise Bias	Some development teams are overly confident; their attitude is, "We are so good, we can handle any risk, so why worry?"
Data Presentation Bias	As Mark Twain was fond of saying, ⁵ "There are three kinds of lies: lies, damned lies, and statistics." He derided conclusions based on statistical analysis because it is possible to alter the analysis to fit the proposition; in other words, you can find support for any position if you work the numbers hard enough. If a team uses a statistical approach to analyze risk, then a rigorous and objective analysis of the underlying assumptions is required to avoid biasing the perception of the risk.
Conservatism Bias (dogma)	Comparing current risks to those previously encountered can be an effective strategy because it takes prior experience into account; but it can lead to assumptions that mitigation should be done a certain way because it has <i>always</i> been done that way.
Law of Small Numbers (variability)	This refers to the false assumption that small sampling numbers have a large associated error (also see Data Presentation Bias).
Self-Fulfilling Prophecy	The risk is a true one; the team acts in a way that ensures it will materialize into a problem. For example, if the team is convinced that the new configuration management tool will lead to project failure, then they will not expend the effort to learn the proper way to apply the tool.
Gambler's Fallacy (probability)	Some people mistakenly think that future probability is altered by past events -- i.e., the chance that a seven will be rolled next is smaller because it has been rolled three times previously. The probability of a risk becoming an actual problem on a current project is not lessened because the same didn't materialize on the previous project.
Incorrect Associations	This refers to assuming a cause-and-effect relationship between two unrelated situations -- for example, assuming that a project quality problem is the fault of the programmers rather than of the poorly conceived and rapidly changing system requirements.
Sin of Omission	Leaving out critical data is almost as bad as including incorrect information.

The common theme of all the pitfalls in Table 1 as well as the different

business approaches we discussed earlier is that the people who are investigating and evaluating risks can alter the perception of risk. The single most effective strategy for avoiding these pitfalls is to be aware of their existence, so you can consciously identify and counter them. After all, success is based on understanding the situation as it truly is, rather than how we would wish it to be. As the late physicist Richard Feynman very eloquently pointed out,⁶ "For a successful technology, reality must take precedence over public relations, for nature cannot be fooled."

Corporate Culture and Risk

Experts in company behavior have demonstrated that, contrary to what we might assume, organizational culture is often independent of maturity; that is, culture is more closely aligned to the style of leadership encouraged by company executives. According to Geofee and Jones⁷ there are four basic cultural types, each with a positive and negative form: Fragmented, Networked, Mercenary, and Communal.

A Fragmented culture is typified by independent action, such as scientific research. A well-connected, in-group mentality typifies Networked cultures, with a focus on personal relationships. Mercenary cultures are strongly goal-oriented, even at the expense of team morale or well-being. Finally, Communal cultures are typified by a closely connected group with well-focused goals (i.e., a "we are family" approach).

As it is for different levels of organizational maturity, the perception of risk is dramatically different for each culture. A Fragmented culture might view risk through each individual's eyes, biasing the risk analysis toward personal gain. Networked cultures are typically more concerned with risks such as political issues that might affect the health of the team (even over the health of the company), potentially ignoring other risks to project success. Mercenary cultures focus on all risks that affect project success, but are often less concerned with risk to people, such as the risk of burnout or strained team relationships. Communal cultures, which are fairly rare, have well-developed interpersonal relationships and are least likely to bias the risk analysis. However, they are also the most likely to exhibit arrogance regarding their own success and therefore to assume risks without enough potential gain to sufficiently counter the potential loss.

Culture and Maturity Risk Profiles

Within each of the four cultures, it is important, during risk analysis, to understand the effect that the corporate culture will have on the interpretation and ranking of risk. The assignment of importance based on perception should always be considered, or at least reviewed, in this light. Although corporate culture is relatively independent of corporate maturity, there are several typical combinations.

The Entrepreneurial Profile. Entrepreneurial companies are often Mercenary or Communal in nature, and are most influenced by the company founder. If the founder has a relentless focus on business success, then the culture will tend to be more Mercenary. If a family

finds the company, then the culture is more likely to be Communal. In either case, the concern is that risks will be judged to be less consequential than they really are, given the high degree of arrogance that is typical of both cultures and fostered by the company style. To avoid marginalizing critical risks, teams should use objective measures (e.g., financial cost, lost business opportunity) to rate risk exposure rather than a more subjective approach (e.g., past experience, "instinct").

The Mature Profile. Mature companies (five or more years old with greater than \$20 million annual revenue⁸) are frequently Networked. This is due to the relatively long period of time that the company employees have had to work together and form relationships. This combination provides for stability and good interpersonal relations to judge and mitigate against risks. The major caveat is to be on the watch for cliques that seek to protect their own interests (i.e., the well-being of the group) in opposition to the entire company. One strategy to prevent this form of "empire building" is to include a senior executive in the project planning and risk discovery phase of the project, and to encourage cross-team interactions. The mere presence of a senior executive will discourage blatant inter-team rivalry, but other team conflicts can be addressed only by open communication between teams.

The Bureaucratic Profile. Finally, Late Aristocracy/Bureaucratic organizations have either a strongly Networked or Fragmented culture, depending on whether the people involved are co-located or distant from one another. The principal difficulty with this type of organization is its reluctance to accept the kind of change a new project represents. Conducting risk discovery and assessment requires obtaining continuous commitment from senior management and a very aggressive/persistent project leader who can overcome organizational roadblocks.

Risk Identification and Mitigation

Having considered some of the complex contexts in which risk assessment will occur, we can turn our attention to the mechanics of risk identification and ranking. Risks come in many forms, but software companies share a fairly large set of common risks, as shown in Table 2.⁹

Table 2: Common Forms of Risk

<ul style="list-style-type: none"> • Requirements volatility • Poorly defined requirements • Unrealistic schedule pressure • Low quality (error-prone modules) • Cost overruns • Corporate politics • Excessive paperwork • False productivity claims 	<ul style="list-style-type: none"> • Lack of a defined, repeatable process • Inadequate organizational structures • Overemphasis on short-range planning • Malpractice (incompetent management) • Staff deficiencies • Unrealistic budgeting
---	--

<ul style="list-style-type: none"> • High maintenance costs • Inaccurate cost estimating • Poor configuration controls (change management) • Inadequate understanding of risk • Poor customer relations (expectation management) 	<ul style="list-style-type: none"> • Over-engineering • Subcontractor deficiency • Shortfall in the execution of external tasks • Use of bleeding-edge technology • Inadequate system performance • Inadequate deployment planning
---	--

The list in Table 2 is in no way exhaustive, and it is critical to identify, analyze, and contain *any* situation that can significantly affect or impede the project.

Risk Description

There are four basic steps in describing risks, which can help lead to success in any software development project:

1. **Identification:** Discovery of potential loss.
2. **Assessment:** Determining the level of exposure to loss.
3. **Mitigation:** Creation of a risk containment or avoidance plan.
4. **Closure:** Successful avoidance or compensation.

These steps will lead to a complete description of all risks, which should be captured in a formal "Risk List."¹⁰ Each risk on the list should have the following details:

- **Definition:** Concise statement of the risk.
- **Consequence:** Expected impact if the risk is realized.
- **Likelihood:** Probability that the risk will occur.
- **Exposure:** Expected loss weighted by the probability of occurrence (Exposure = Likelihood * Consequences).
- **Risk Ranking:** Relative ranking-based consideration of Exposure.
- **Indicators:** Signs and symptoms to monitor the risk.
- **Mitigation Strategy:** Description of approach to avoid realization of the risk.
- **Contingency Plan:** Secondary plans to deal with consequences of risk realization.

Now that we have a framework for capturing risks, we can identify and

assess each risk in turn.

1. Identification of Risk

The first step in any risk management scheme is to identify all the factors that can lead to delay or cancellation of the project. Many of the risks in Table 2 are associated with the development *process*, the *product* under construction, or the management of the *project* itself. The project team should therefore consider these three areas while asking the following questions to determine the types of risk, the likelihood of occurrence, and the impact the risk would have on the project:

- What can go wrong?
- How likely is this to occur?
- What would be the cost or damage if this happened?
- How can we avoid this?

2. Assessment of Risk

The second step is to assess the level of exposure for each risk. If the actual dollar values can be determined, it is of benefit to provide this information as part of the risk description. This permits a non-biased ordering and ranking of the risk impact. Alternatively, you can use a simple scale to qualitatively rank the risks,¹¹ as shown in Figure 1:

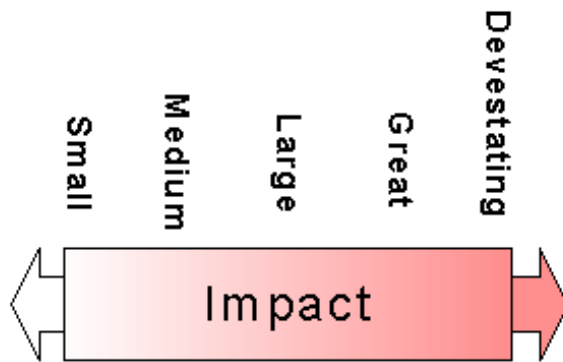


Figure 1: Qualitative Ranking Scale for Risks

A low-impact risk is one that incurs only negligible costs. Alternatively, a devastating impact risk is one that will lead to project cancellation, legal issues, and/or termination of the project team. When determining risk impact, it is helpful to take a backward-looking approach -- that is, to pretend as if the risk has already materialized and judge the resulting situation. As you do this, you should consider the effects on project personnel, customer satisfaction, and senior executive confidence, as well as the actual economic costs borne by the business.

Likelihood is a simple probability assessment from 1% (very unlikely) to 99% (all but unavoidable). Indicators are signs and symptoms that suggest the risk has either been mitigated or is occurring; for example, if

the risk is customer satisfaction, then a poor customer response to a release is an indicator that the risk is occurring or has already occurred.

Mitigation of Risk

Risk mitigation is an attempt to avoid or prevent the consequences associated with the risk. There are three primary mechanisms for mitigation:

1. **Risk reduction:** Reduce the probability of risk occurrence by forward planning.
2. **Risk avoidance:** Reorganize the project so that it cannot be affected by that risk.
3. **Risk transfer:** Reorganize the project so that someone or something else bears the risk (customer, outsource vendor, another company, another department, or the like).

Mitigation plans should be written and placed into effect as soon as possible. Since a key element of iterative software development is to take a risk-based approach to development and attempt the highest risk items early in the project, teams should address and retire the highest impact risks as early as possible in the development lifecycle. Addressing high-risk items early in the project is beneficial for several reasons:

- It leaves you plenty of time to deal with risk-generated problems.
- It reduces the impact of potential risks on the quality and timeliness of system delivery.
- The costs associated with a risk often increase over time.¹²

Contingency Plans contain steps that should be taken once a risk becomes a reality. For example, if the risk is to project schedule, than the contingency will be to maintain a time "buffer" to be used if the mitigation strategy of iterative development fails. For technological risks, the contingency may be to have a fallback plan to continue using the current solution while the new solution is made to work. Finally, for political risks, the contingency may be to petition the most senior executive in the company if a lower level executive becomes an impediment (typical of Bureaucratic firms).

As noted earlier, options must be associated with any endeavor that has enough potential for loss to qualify as a risk. In the event that a risk is realized, it is best to have a plan in place to deal with the costs and minimize impact to the project. Just as with a city disaster plan, the hope is that this plan will never have to be used, but it is best to define and practice it beforehand, just in case. A contingency plan might include additional schedule time "held in reserve," additional budget for emergency consultants, or other pre-disaster planning.

Conclusion

Risk management is critical for a project's success. An understanding of 1) how personal biases affect risk perception, and 2) the effect of corporate culture and maturity on risk planning and acceptance can better prepare the project team to manage major project risks. These influences must be carefully and objectively considered when creating a risk management plan. Risk management starts with the identification and documentation of situations or conditions that can lead to undesired consequences, including project cancellation. Risk mitigation consists of reducing the threat, avoiding undesired consequences, and/or transferring costs for identified project risks.

By assuming a risk-based approach to scheduling, including addressing the highest-risk items early in a project, the project team can increase the overall probability of success. It is important to avoid gambling with a project's success, which means accepting risks that are not justified by potential gains. By adopting a more rational approach to risk, the project development team will be able to prepare for all foreseeable circumstances and plan to meet them. Running successful projects, therefore, involves spending time to determine potential threats, understanding relative costs and benefits associated with those threats, devising mitigation plans to avoid realization of associated costs, and creating contingency plans to deal with possible undesirable outcomes.

References

E. Hall, *Managing Risk: Methods for Software Systems Development*. Addison-Wesley, 1998.

R.N. Charette, *Applications Strategies for Risk Analysis*. McGraw-Hill, 1990.

I. Azides, *Corporate Lifecycles: How and Why a Corporation Grows and Dies and What to Do About It*. Prentice-Hall, 1988.

Mark Twain (Charles Neider, Editor), *Mark Twain's Autobiography*. Harper Perennial, 2000.

R.P. Feynmen, *What Do You Care What Other People Think?* Bantam Books, 1988.

R. Goffee and G. Jones, *The Character of a Corporation*. Harper Collins, 1998.

E. Flamholtz, *Growing Pains - How to Make the Transition from an Entrepreneurship to a Professionally Managed Firm*. Jossey-Bass Publications, 1990, p.407.

B.W. Boehm, *Software Risk Management*. IEEE Press, 1989.

C. Jones, *Assessment and Control of Software Risk*. Prentice-Hall, 1994.

I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. Addison-Wesley, 1999.

Notes

¹ E. Hall, *Managing Risk: Methods for Software Systems Development*. Addison-Wesley, 1998.

² R.N. Charette, *Applications Strategies for Risk Analysis*. McGraw-Hill, 1990.

³ In this context, "maturity" relates to the natural growth of a company, and refers to the corporate business approach (e.g., entrepreneurial, balanced, change adverse, etc.). This should not be confused with the SEI's Capability Maturity Model (CMM), which measures the level of competence with respect to the software development process.

⁴ I. Azides, *Corporate Lifecycles: How and Why a Corporation Grows and Dies and What to Do About It*. Prentice-Hall, 1988.

⁵ Mark Twain (Charles Neider, Editor), *Mark Twain's Autobiography*. Harper Perennial, 2000.

⁶ R.P. Feynmen, *What Do You Care What Other People Think?* Bantam Books, 1988.

⁷ R. Goffee, and G. Jones, *The Character of a Corporation*. HarperCollins, 1998.

⁸ E. Flamholtz, *Growing Pains: How to Make the Transition from an Entrepreneurship to a Professionally Managed Firm*. Jossey-Bass Publications, 1990, p.407.

⁹ See B.W. Boehm, *Software Risk Management*. IEEE Press, 1989, and C. Jones, *Assessment and Control of Software Risk*. Prentice-Hall, 1994.

¹⁰ I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. Addison-Wesley, 1999.

¹¹ E. Hall, *Op.Cit.*

¹² See Jacobson, Booch, and Rumbaugh, *Op.Cit.*, and W. Royce, *Software Project Management*. Addison-Wesley, 1998.



For more information on the products or services discussed in this article, please click [here](#) and follow the instructions provided. Thank you!