

Imagining Managing Risk

D R A F T B

C. C. Shelley

June 2008

Oxford Software Engineering
9 Spinners Court, 53 West End, Witney, Oxfordshire, England, OX28 1NH
shelley@osel.netkonect.co.uk

ABSTRACT

Risk management is routinely performed in projects in the software industry, but its appropriateness and value is often questionable. This note describes conventional practice and identifies some common problems. Some simple revisions to conventional practice are proposed to improve projects' perceptions of risk, and their ability to establish a good risk profile.

1. Introduction

The author has participated in risk management activities within projects, taught risk management and defined risk management procedures – mostly involving a more or less conventional approach. This approach is described below.

(Before describing conventional project risk management, as practiced, it may be worth commenting on software projects. The term 'project' is very widely used to describe work requiring the development or support of software. In many cases the use of the word 'project' is unwarranted. The work is not large, complex, risky or in any meaningful way original and is not in need of 'project management' or a project manager. Never the less it is treated as a project and project managed simply because there is no other term in use to describe a package of development/support work – except perhaps 'programme' which is no better. As a consequence large amounts of development work are overmanaged and over administered, discrediting project management tools and techniques. There is a clear need for a vocabulary of terms to describe the range software development work with widely divergent characteristics.

Where the work can reasonably be described as a project the use of planning and management often takes little notice of the character of the work, or its context when invoking project management practices. The tailoring of management practices to particular project or organizational need is often rudimentary or inadequate, again leading to the discrediting of techniques and tools, when in fact it is their application that is at fault.)

2. Conventional Project Risk Management

The conventional approach to software project risk management appears to be well established as a de facto industry standard. The author has encountered it, and used it, in many organizations and numerous projects across Europe. It tends to be performed in the following way:

At the beginning of the project risks are listed - either at a dedicated risk workshop, or as part of a project start-up, or kick-off meeting. Typically threats and hazards to the project are brainstormed, insofar as they are collected without too much filtering or criticism. These may be analysed to identify duplicates or grouped into categories.

The threats (or groups, or categories) are then assigned a probability which can be described in any of a number of ways: high/medium/low, a range of 1 to 5 with 1 'low' and 5 'high', or percentage probability ranging from 1 to 100 percent, perhaps assessed within percentage ranges.

The impacts of the threats to the project are evaluated in a similar way: H/M/L. 1–5 etc.

The priority of the resultant risks are then calculated as a function of impact (i) and probability (p). Usually $f(i,p)$ is $(i * p)$ but can also (rarely) be given as $(i + p)$. Risks are then ranked in priority order with the high priority risks to be addressed first.

Mitigation plans are then developed (although these are often no more than a sentence or two, and then contingency (either effort or time) can be allocated. This may be done at a later date – or not at all. (Occasionally risk procedures show how a financial cost of the risk can be calculated – but this is not usual. Software projects tend not to be equipped to deal with financial matters.)

The risks and their attributes, mitigations and contingency may then be recorded on a risk management tool, often a spreadsheet, but increasing project we pages are being used.

Risk managers may be assigned to particular risks, although the default is the project manager or team lead. The expectation is that risk managers will proactively monitor risks and take necessary preventative measures, mitigating actions or have the required contingency should the threat manifest itself. The extend to which this is done will depend on both the credibility of the contents of the risk register and the diligence and professionalism of the risk manager.

The merit of this approach are its pervasiveness. Variants of this process are by many projects across the industry (in Europe). It will be found as a stand alone process, sometimes supported by procedures, and often by risk recording tools. It will also be found integrated into project management processes. Consequently it is well understood and used without much apparent confusion or dissent. And while it can work its primary demerit is that it is prone to being neglected. Identified risks can lack credibility and as the project proceeds the risk register can be ignored, the presumed risks can age and lose their relevance, and risk management activities are marginalized by the day to day realities of the project.

3. Technical Problems with this Approach

The approach described above has a number of technical problems:

1. It is usually not stated what is at risk. It is implicit that 'the project' is the focus, but the specific goals or objectives are often not stated and the impact of threats is not clear either. The formulation of risk statements that says how failure of the

project will affect the host organization or the customer is not easy, requiring care and thought. If the risk statement does not accurately capture the essence of the project this can discredit the risk management activity. Alternatively the formulation of a risk statement can cause difficulties and embarrassment by revealing that there is little agreement on project objectives, or that no one has a clear idea what the project's objectives really are. While this revelation can be valuable it is also potentially 'career limiting'.

2. There is rarely guidance available on the categories of threats, or generic lists of threats to be considered. Boehm and Capers Jones have analysed many projects and propose lists of generic threats, but these lists are rarely used. Consequently significant areas of risk can be neglected or missed, or duplicated. Some areas may receive too much attention with the threats being trivial or irrelevant.
3. Data to validate risk profiles is rarely available. It is therefore difficult to validate the project's risk profile., and if it is felt to be unrepresentative, i.e. the list of risks is perceived as incomplete, or overloaded or biased this can discredit the risk management process.
4. Threat probabilities are estimated by judgement, discussion and guessing. Statistical modelling is encountered occasionally but is rare. There is no evidence that threat probabilities are at all accurate.
5. The performance of risks and risk management is rarely reviewed as part of a post project review. There are two difficulties: for projects that conclude successfully it is difficult to discuss threats that did not manifest, and for projects that were unsuccessful it is unlikely that a review of risk management practice itself will be reviewed – although some allocation of blame and after the fact rationalization may happen.

4. Some 'Soft' Problems

In addition to the technical problem a number of people related 'soft' problems:

1. The identification and analysis of threats, being familiar, tends to be performed by

rote. And where groups perform these activities not all participants may engage, allowing one or two to lead the activity and populate the project risk register. This can be because the risk management activity is routine, and routinely applied to work that is not risky (i.e. not real projects), or because it is perceived as the job of managers or team leads to manage risks. Symptoms of risk management by rote is where risks are 'the usual suspects'. No real imagination or judgement given to the risks that could undermine this project.

2. Where real and significant risks are identified these can be suppressed by management and or customers because, if recorded, they could cause uncertainty or embarrassment. (The management of risks seems to be perceived differently in Europe and US with the US more sensitive to analysis of risks as critical, unsupportive or negative – or is this a case of being divided by a common language?)
3. Being objective is difficult and objectivity can suffer – and is difficult to validate. The identification and analysis of risks is difficult to do by those that may be affected by the risks – there is an element of 'it can't happen to me' present.
4. It is very difficult to discuss probabilities. Probabilities will be asserted, challenged and compared but very little in the way of analysis is (or can be) performed with current categorization techniques which tends to be numerical and derived by assertion and voting.
5. Frequent, ongoing or routine (re)evaluation of risks for a project, especially when the risk management practice (as well as the project) is already perceived as routine can lead to a 'normalization of risk', similar to the 'normalization of deviance' reported by Dianne Vaughn in 'The Challenger Launch Decision'. The project learns to live with the risk. The threats are normal – 'just the way things are'.
6. It is usually assumed (and occasionally documented in risk management procedures and guidelines) that risks will be either mitigated or contingency put in place to deal with the threat if it manifests; that the threat will be avoided by reduction, deferment or transfer. In essence pushing the risk away. It is rarely considered that embracing the risk; precipitating the threat,

early or at the project's convenience may be a sound tactic

5. Some Observations

Risk management, as practiced is biased towards a 'lowest common denominator' process. The procedure described earlier includes most common element of project risk management, but it is rare to find all the elements performed, or performed well. In general software project risk management practice is both routine and poor, with limited appreciation of risks and with a risk register populated by some risks data and red/yellow/green indicators for risks that may be revisited with declining interest as the real risks close in on the project unobserved.

When good risk management is found it is usually a matter of attitude. Those involved are interested and concerned. And although the procedure may not be 'state of the art' it can provide an adequate framework for those taking a proactive approach managing their risks.

When a project is seriously damaged by a risk it is often a surprise to the team (in the author's personal experience). The threat seems to sneak up on the project and pounce – but to outsiders it appeared to be inevitable. (So why did they not warn the project? Maybe because since it appeared obvious to them, they presumed it appeared obvious to the project, alternatively, it was none of their business.)

People like risks. Project 'firefighting' is fun and can be very well rewarded. (How often do management reward project staff for uneventful, unmemorable projects that delivered as they said they would, as opposed to those project staff that went 'above and beyond', in trying circumstances to bring the work back on track.)

6. And projects do need risks. Projects do not function in a steady state, but need some risk and uncertainty to provide some 'intellectual lubricant' and interest, to surface the good ideas, and to function. But care needs to be taken deciding which risks and to accommodate.

This need for risk may be a need for uncertainty. People may function more effectively and with more enthusiasm with some uncertainty. And from a project perspective some uncertainty can be a benefit: uncertainty is not strongly correlated with risk. In some case uncertainty is risky: when it is not clear what objectives are, what is wanted or by when. In some circumstance uncertainty is neutral, for example if an objective may be achieved between three to seven months, but is not needed

before twelve months, then the five months uncertainty is irrelevant. In some situations uncertainty is useful – a freedom from constraint. For example 'set based thinking', as described in lean approaches to software development. This approach proposes deferred decision making in order to keep development options open; in effect maintaining uncertainty.

6. Proposals

To make risk management more relevant to real software projects – rather than routine maintenance and support work and small tasks, - to engage those involved, and to manage risks to better effect it is proposed that:

1. *Encourage the use of picture to describe risks:* Get team members to sketch simple pictures to visualize and share perceptions of risks before attempts are made to quantify them. Informal hand drawn pictures that enable people to represent their perceptions of risk are produced on a whiteboard or flip chart, annotated, extended and discussed. Examples may include a risk timeline – showing *when* the project is most at risk, with analysis of why coming later. And informal diagramming of risk using shape and size to provide a representation of perceived threats and their character.
2. *Strive for objectivity:* Include an independent peer in risk management activities – not as a facilitator, although this may be desirable too, but simply to provide a degree of objectivity and to question any team assumptions and beliefs.
3. *Risk management is a team activity:* Risk workshops, where risks are identified and discussed, should be encouraged as a team activity where people work together. Risk identification, or analysis should not be a task for an individual or specialist working alone. Neither should it be a distributed activity with electronic communications – unless this is absolutely essential. Carefully managed risk management could become one of the 'catalytic processes' that deliver far more value than would be expected.
4. *Keep it relaxed and creative:* Risk workshops should be informal and introduce techniques to break conventional thinking – e.g. statement inversion. To

maintain informality consider risk workshops as a peer only process.

5. *Introduce Risk:* Consider retaining some identified risks, and even *introducing* risk with a view to developing a designed and balance risk portfolio. Team members will engage with a project that is interesting and exciting. This is done already, but usually not acknowledge, for example letting developers use a new technology, even though not strictly necessary. Some chaos injected into the project from time to time acts as an 'intellectual lubricant' – brainstorming can act as the agent for introducing doses of chaos. DeMarco suggests allowing every project to be a pilot: allow the project to relax certain standards to try something new. (One of the problems with pilots is that they are always successful because of Hawthorne effects. Every project a pilot simply exploits this.) This introducing, or perhaps more realistically, balancing of risk to suit the character of the project, and the host organization, encourages a positive, proactive approach, which may ultimately increase the project's chances of success, however that is evaluated.

7. Closing Remarks

Risk management as currently practiced often ignores the human element – both as a motivator and enabler of the risk management process itself, and the need for risk or uncertainty as a motivator for team members in software projects and as a mechanism for reducing design constraints.

Risk management can be recast to recognize and exploit the human element of software development by allowing team members to articulate or present their perceptions of risk and uncertainty in empirical, graphical ways before abstracting to quantitative data points. And also by engaging people more in risk management and exploiting their, often unrecognised, capabilities, and by balancing and managing project risk, not simply suppressing it or attempting to eliminate it.

But care should be taken to ensure that while some carefully considered unknowns and uncertainty that excite, motivate, free development work and increase the probability of project success are introduced, dangerous *unknown* unknowns are recognised and, so far as is possible, eliminated.