Collaborative Risk Management

D. Greer[1] and D.W. Bustard[2]

[1]School of Computer Science
Queens University, Belfast
Belfast, BT7 1NN, UK
des.greer@qub.ac.uk

[2] School of Information and Software Engineering
University of Ulster, Coleraine
BT52 1SA, UK
dw.bustard@ulster.ac.uk

Abstract ***The SERUM methodology, developed by the authors, provides a framework for risk management based on a broad systems approach to software engineering. For its effective operation, all those affected by risk, or contributing to its management, need to be aware of the threats involved and how they are being addressed. Such awareness requires good ongoing communication among these stakeholders. This paper suggests how the Internet might support that communication. The approach, RA!SE (Risk Alert! for Software Engineering), assumes that software engineering data is held centrally in an organization, with stakeholders given access to relevant development and operational information through a Web-based communication tool. The paper clarifies the need for collaborative risk management and develops requirements for tool support. Details of a resulting prototype are presented, and illustrated with risk data from a previous study.***

Keywords**: Software Risk Management**

I. INTRODUCTION

Risk is inherent in the development and operation of every computing system. It is important, therefore, that risk is handled adequately to ensure that the potential for problems is minimized and that the necessary contingency plans are in place. Despite the strong arguments in favor of using risk management practices routinely, there remain major hindrances to their widespread adoption. There are perhaps three main inhibitors.

1. Software Engineers are optimists.

In general, humans have a natural aversion to risk because they tend to perceive what they expect (or hope) to see, rather than recognize what actually exists (termed Perceptual Denial). Experiments in the field of psychology show that people are generally reluctant to recognize future problems [5,19]. Other evidence from human psychology indicates that people tend to rate positive outcomes more likely than negative outcomes, when both have equal probability [2]. In software engineering, these traits manifest themselves as misplaced optimism about project and programming success [4]. The problem is made worse by a common perception that risk management is an 'extra activity', less important than development tasks and largely someone else's responsibility [16].

2. Slippage has become an accepted aspect of software development.

Many software engineering projects exceed their expected functionality, cost or schedule. This encourages a tacit acceptance of slippage and slows the response to problems that arise. Unchecked, this can lead to "runaway" projects [12] and even complete project failure. Such a possibility has increased recently because of pressure to develop software in "Internet Time" to meet commercial deadlines for e-business applications [9].

3. The scope of Risk Management is often too narrow.

For any piece of software there are risks in both its development and subsequent operation. Those risks may be directly associated with the software or be in the environment in which the software is either developed or used. The risks themselves are dynamic and will vary during the software development process, as environmental changes occur, and as the software evolves. This implies that a wide range of threats needs to be recognized, maintained and monitored in an integrated way, by a wide range of stakeholders. In practice, risk management is often applied much more selectively, with a focus on either development or operation, and relatively little attention is paid to monitoring and maintenance.

A greater visibility of risks would help alleviate all of these problems. In particular, software engineers cannot easily ignore risks if presented to them directly. Also, encouraging the engineers to identify risks, propose risk mitigation strategies and agree risk assessments will further reinforce such appreciation. Similarly, showing the effect of slippage on risk assessments can help bring out the threat that any deviation may have on a project or overall system. Finally, if all stakeholders affected by or handling risk can see the broad range of threats that are present they can better appreciate their impact, inter-dependence, and management.
Communication of risk data is central to improved visibility [10]. The emergence of Internet-based technology provides an opportunity to increase the level and quality of risk data being shared within an organization. There are several factors involved:

1. Risk data held centrally and accessible using a standard Web browser can be reached by anyone with very basic computing facilities.

2. The cost of distributing risk data and obtaining feedback on risks is greatly reduced.
3. Communication using the Web is potentially instant, so that reactions to risks are more likely to be timely.
4. Information can be tailored to suit each stakeholder.

The remainder of this paper examines the requirements for Internet support for collaborative risk management. The approach, RA!SE (Risk Alert! for Software Engineering), is illustrated by features from a prototype Web-based risk management tool.

II. Support for Collaborative Risk Management

There are many sources of risk in an organization. Of those relevant to software engineering, there are two main classes to consider: software development risks and software operational risks. Operational risks are associated with a software product behaving in an unexpected and detrimental way. Development risks include potential problems with the schedule or budget of a software project. This paper will concentrate on development risks although the RA!SE approach has also been applied to operational risks.

The management of risk is part of a larger development process. Figure 1, for example, summarises the nine-stage SERUM process (Software Engineering Risk: Understanding and Management) [13,14], which can be used to develop an incremental organisational change plan [1].

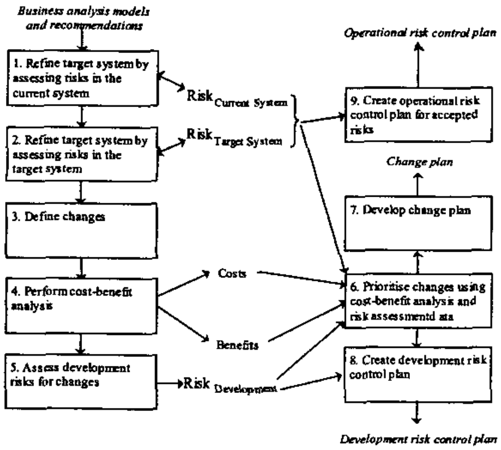

Business analysis models and recommendations
1. Refine target system by assessing risks in the current system
2. Refine target system by assessing risks in the target system
3. Define changes
4. Perform cost-benefit analysis
5. Assess development risks for changes
Risk Current System
Risk Target System
Costs
Benefits
Risk Development
Operational risk control plan
9. Create operational risk control plan for accepted risks
Change plan
7. Develop change plan
6. Prioritise changes using cost-benefit analysis and risk assessmentd ata
8. Create development risk control plan
Development risk control plan

Fig. 1. SERUM process summary

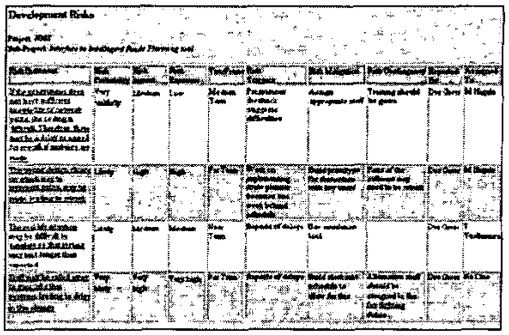
The first two stages of the process identify risk in the current and target systems and use that knowledge to make the target system proposal more robust. In the third stage, individual self-contained changes, which move the current system toward the defined target system, are identified. These include, for example, the introduction of specific IT systems, coupled with any consequential changes in related business activities (and vice versa). A cost-benefit analysis and a development risk assessment are made for each change in the fourth and fifth stages. These, together with the risk reduction achieved by the change, as calculated from stages one and two, feed into the prioritization of changes in the sixth stage. The individual changes are then grouped to form incremental steps in the seventh stage, resulting in the production of the change plan. All these activities use or extend the activity descriptions of the current and target systems. Control plans are also developed in stages eight and nine, to help manage the identified operational and development risks.

In general, risk management can be divided into six activities: Risk Identification, Risk Assessment, Risk Prioritisation, Risk Management Planning, Risk Resolution and Risk Monitoring [3]. Each of these is now examined to identify possible Web-based tool support.

III. Risk Identification

Questionnaires or checklists are the most common means of risk identification. For example, in the Software Engineering Institute's (SEI) Risk Program a questionnaire was developed to elicit risk within the framework of a taxonomy of software development activities [6]. However, there is an argument that such generalized checklists constrain project managers and do not cover all risks for all projects [21]. Alternatives to a questionnaire are described in [3,10,11,20]. These include:

1. Identification based on past experience. There is evidence that project managers can identify risks based on knowledge of problems encountered in previous projects [21].
2. Identification based on historical data, perhaps through the use of a project database.
3. Periodic risk reporting, where team members routinely report risks typically at project progress meetings.
4. Voluntary risk reporting, where risks can be reported formally at any time during a project, perhaps anonymously.



Development Risks

Project: [illegible]
Sub-Project: Interface to Intelligent [illegible] Planning tool

Risk Statement	Risk Probability	Risk Impact	Risk Exposure	Timeframe	Risk Triggers	Risk Mitigation	Risk Contingency	Reported By	Assigned To
[illegible]	Very unlikely	Medium	Low	Medium Term	[illegible]	Assign appropriate staff	Training should be given	[illegible]	[illegible]
[illegible]	Likely	High	High	Far Term	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	Likely	Medium	Medium	Near Term	Reports of delays	[illegible]		[illegible]	[illegible]
[illegible]	Very likely	Very high	Very high	Far Term	Reports of delays	[illegible]	[illegible]	[illegible]	[illegible]

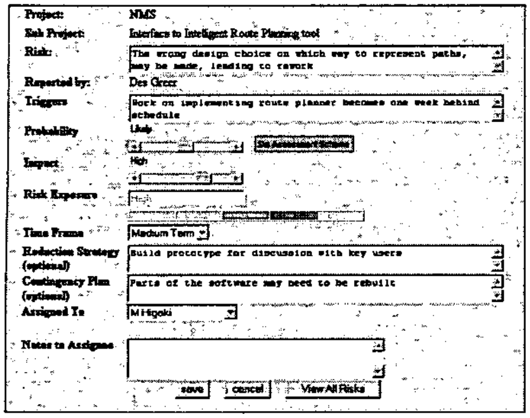
Fig. 2. Viewing a risk summary for a sub-project in RA!SE

Each of these approaches can benefit from being supported by a Web-based tool. This is especially true of identification based on historical data, since such data is best maintained in a central database. For periodic or voluntary risk reporting, the fact that these can be carried out at any time and from any workstation reduces the burden of reporting risks. Anonymous risk reporting is also accommodated easily. The prototype tool provides a screen for identifying risks associated with systems or projects. Projects are assumed to consist of sub-projects and systems of sub-systems.

For a given project and sub-project the tool displays a summary of the associated risk data as shown in Figure 2.

The sample data shown is for a Network Management System (NMS) at the NEC Corporation [15]. This provides (i) a statement describing each risk, together with an assessment of its (ii) probability and (iii) impact; (iv) the trigger identifying that a problem has occurred; suggestions for (v) risk mitigation, (vi) contingency, and details of (vii) who reported the risk and (viii) who is responsible for investigating and monitoring it.

From this window a new risk can be added or further details of a risk viewed or edited by selecting the desired hyperlink. The window for editing a risk is shown in Figure 3.



Project: NMS

Sub Project: Interface to Intelligent Route Planning tool

Risk: The wrong design choice on which way to represent paths, may be made, leading to rework

Reported by: Des Greer

Triggers: Work on implementing route planner becomes one week behind schedule

Probability: Likely

Impact: High

Risk Exposure: High

Time Frame: Medium Term

Reduction Strategy (optional): Build prototype for discussion with key users

Contingency Plan (optional): Parts of the software may need to be rebuilt

Assigned To: M Higaki

Notes to Assignee:

Fig. 3. Editing risk details in RA!SE

The main details recorded are a risk statement, the reporter, and any known triggers for the risk. The details presented here depend on the user's role. In this case the user is a Risk Manager with access to all functions. Others might be restricted to basic risk entry and their screen reduced accordingly.

IV. Risk Assessment

To measure risks quantitatively, the probability of the risk and its impact must be measured. Their product is known as the risk exposure [3].

Risk exposure = Probability (unwanted event) x Impact (unwanted event)

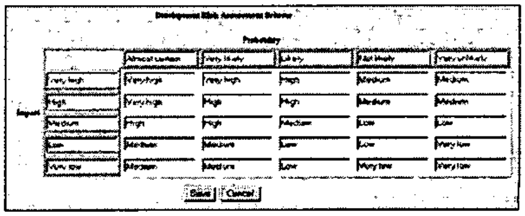
Quantitative measures are potentially more useful because they permit a more reliable means of comparison. However, there are a number of problems with assessing risks quantitatively. One is achieving a reliable measure. Due to their nature, risk measurements are estimates and so any measurement will be inaccurate. Further, not all the information required for the measurement may be available so the measure can be imprecise. Also, the information on which risk estimates are based is open to a biased interpretation and misunderstandings. Often, individuals believe that they are more accurate than they really are [22].

In general, we prefer a classification approach where probability can be assessed using symbolic terms such as "high", "low", and "medium". Similar terms can be adopted for a measure of loss. Assuming a consistent use of these terms, iti s then possible to convert them to numbers if desired [11], but qualitative measurements can also be compared directly. Classification means that a simple scale is used to assess risk exposure. For example, the SEI's Software Risk Evaluation (SRE) method [23] has a mechanism for classifying risks with one of three magnitudes using three classes of probability and three classes of loss. Table 1 illustrates this approach.

Table 1. SRE Classification of Risks

	Probability		
Loss	*Very Likely*	*Probable*	*Improbable*
Catastrophic	High	High	Medium
Critical	High	Medium	Low
Marginal	Medium	Low	Low

There are several variations of this mechanism. For example, the European RiskMan methodology [7] classifies Risk Exposure as "Unacceptable", "Critical", "Significant" or "Minor". These classes are obtained by considering both the loss due to failure and the probability of the failure occurring as "Low", "Medium" or "High". The IEC (International Electrotechnical Committee) gives four classes of risk [17] "Intolerable", "Undesirable", "Reasonable" and "Negligible". Figure 4 demonstrates how this approach is used in RA!SE using data from the NEC NMS project. The phrases used to represent probability and impact scores in this case were set by the analyst at NEC.



Development Risk Assessment Scheme

Impact \ Probability	Almost certain	Very likely	Likely	Not likely	Very unlikely
Very high	Very high	Very high	High	Medium	Medium
High	Very high	High	High	Medium	Medium
Medium	High	High	Medium	Low	Low
Low	Medium	Medium	Low	Low	Very low
Very low	Medium	Medium	Low	Very low	Very low

Fig. 4. Setting up an assessment scheme for risk exposure in RA!SE

V. Risk Prioritisation

In practice, risk identification may reveal so many risks that it is difficult to manage all of them in the same detail. Indeed, it is preferable anyway to concentrate effort on those that pose the greatest threat. There are a number of documented methods for prioritising risks. Two common approaches are:

1. Multivoting

Multivoting [24] is a prioritisation method carried out by a group. The group define criteria for the assessment and then group members rank the top N risks. The value of N is recommended to be one third of the total number of risks. In each individual's ranked list, the top ranked risk is given N points, the next N-1, and so on. The points are then totalled to obtain an overall ranking.

2. Pareto Top-N Ranking

This method is based on the Pareto principle where the "vital few" are separated from the "useful many" [18]. To achieve the ranking, risks are first sorted using a risk exposure calculation, then by timeframe of occurrence. The top 20% or so(typically) are then selected as the most important for management.

The second approach was taken in RA!SE simply because this type of risk expertise is more likely to be restricted to a few individuals. Use of tool support means that the tasks of prioritisation are easily automated. Selected users are given a Risk Manager Role and can obtain a display of the ranked risks; these can be adjusted manually, as required. This can be achieved through the window shown in Figure 5. The prioritization can be carried out at project or sub-project level, as in this case.

VI. Risk Planning

Risk planning is the process of deciding and documenting what, if anything, should be done about a risk. This may mean choosing an alternative route that does not involve the identified risk. Charette [8] lists reduction, transfer, protection and pecuniary as valid approaches to handling remaining identified risks.

The reduction approach involves reducing either the probability or consequence of a risk to an acceptable level. For example, if there is a risk of essential project personnel leaving the project causing a schedule delay, a risk reduction strategy might be to offer incentives for personnel to stay, thus reducing the probability associated with the risk.

Transferring the risk means letting someone else deal with it. This is similar to insuring against a risk and usually involves reducing the magnitude of the loss should a problem occur. Often, this is the only appropriate response when the expected loss is severe but of low probability. One example of risk transfer is when a task is subcontracted to another organization.

The protection approach is to try and reduce the likelihood that the occurrence of a risk will affect the total system operation. For example, use of redundant hardware will reduce the risk of a system failure affecting a system.

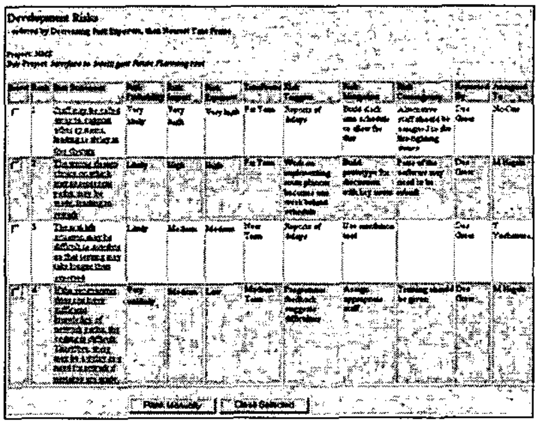
The pecuniary approach is where the risk is accepted and necessitates the monitoring of the identified risk and the creation of contingency plans should a failure occur. Thus, a fund of project resources is set aside for such contingencies.

The planning process should:

1. Determine who has responsibility for the risk.
2. Decide what, if anything, can/should be done to mitigate the risk.
3. Document the action that should be taken, if any [10].

If the analyst determines that the risk can be ignored, no mitigation action is planned. Where no mitigation action is possible, the strategy is to track the risk. The outcome of the risk planning process is a risk action plan that details:

1. Research plans if further information on a risk is required.
2. Acceptance rationale if a risk is accepted.
3. Tracking requirements, such as suitable indicators, for monitoring a risk.
4. Mitigation plans.



Development Risks

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
☐	1	[illegible]	Very likely	Very high	Very high	Far Term	Reports of delays	[illegible]	[illegible]	[illegible]	[illegible]
☐	2	[illegible]	Likely	High	High	Far Term	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
☐	3	[illegible]	Likely	Medium	Medium	Near Term	Reports of delays	[illegible]		[illegible]	[illegible]
☐	4	[illegible]	Very unlikely	Medium	Low	Medium Term	[illegible]	Assign appropriate staff	Training should be given	[illegible]	[illegible]

Fig. 5. Risk Prioritisation in RA!SE

Figure 5 shows that the Risk Manager may assign responsibility for a risk and document the mitigation action that should be taken. The Risk Manager may also record notes that may be useful. On submission, the information is recorded and an email message containing all of these details sent to the person given the responsibility.

A. *Risk Resolution*

The risk resolution techniques employed are specific to each risk. For example, this may involve subcontracting work, building a prototype or introducing certain project controls. In the SEI approach [10], four results are possible in the progress of the risk resolution stage.

1. The risk may be closed, in which case either its probability or its impact has been nullified.
2. The risk may have been reduced and therefore must be re-planned.
3. The mitigation process may have been unsuccessful and a contingency plan must be invoked.
4. The risk is still active and the resolution activity continues.

From the risk manager window (Figure 4), a risk may be closed by selecting it and clicking the appropriate button. In the case of (ii), the risk should be reassessed using RA!SE. In (iii), the risk has become a problem and is beyond risk management. Nonetheless its details are still held in RA!SE for future reference

B. *Risk Monitoring*

Risk monitoring involves measuring attributes of the risk management process, such as the probability of a risk, as a project progresses. These measures may be combined into indicators. For example, risk exposure combines probability and impact measures. Alternative indicators might include, for example, the amount of overtime worked on a project, indicating a certain schedule risk. Triggers may also be defined which relate to threshold values for these indicators. When these thresholds are reached, a defined action may be initiated. Data for risks is thus compiled and reported at appropriate times.

VII. Conclusions

This paper has described the need for an increased awareness of risk and its management. The strategy described is to raise the visibility of risk management by encouraging all stakeholders to identify and appreciate risks, estimate the probability and impact of risks, propose ways of alleviating risks; and contribute to contingency plans to handle the problems anticipated. An approach, RA!SE, implementing this strategy using a web-based system has been described. The approach distributes the risk identification and assessment effort but centralised the overall management of risk. Risks are assessed using a scoring mechanism where probability and impact of risks are separately estimated and combined to obtain an estimate of risk exposure. A Risk Manager can review identified risks and refine the information provided. The Risk Manager can also prioritise identified risks and assign them to individuals to assess and monitor.

Future work will concentrate on refining the RA!SE approach and tool through their experimental application across a range of organisations and circumstances. Fuller integration with the SERUM method, at both process and tool level, will also be investigated.

VIII. References

[1] Bustard, D.W., Greer, D., He, Z., Lundy, P., Oakes, R., and Wilkie, F.G., (2000), Developing a Co-Evolutionary Business-IT Change Plan, in Bustard, D.W., Kawalek, P., and Norris, M.T. (eds), Systems Modelling for Business Process Improvement, Artech House, , pp. 213-232.

[2] Blascovich, J., Ginsburg, G.P., Howe, R.C., (1975) "Blackjack and the Risky Shift, II: Monetary stakes" Journal of Experimental Social Psychology vol.11, pp 224-232.

[3] Boehm, B.W (1991), Software Risk Management: Principles and Practices, IEEE Software, Jan., pp 32-40.

[4] Brooks, F.P. (1975), "The Mythical Man-Month", Addison-Wesley, MA.

[5] Bruner and Postman, (1949) "On the Perception of Incongruity: A Paradigm", Journal of. Personality, vol. 18, pp 206-223..

[6] Carr, M.L., Konda, S.L., Monarch, I., Ulrich, F.C. and Walker, C.F. (1993), "Taxonomy-Based Risk Identification", Technical Report CMU/SEI-93-TR-6, Software Engineering Institute., Carnegie-Mellon University, Pittsburg, PA, 15213, USA.

[7] Carter, B., Hancock, T., Morin, J., Robins, N. (1994), "Introducing Riskman Methodology", NCC Blackwell.

[8] Charette, R. (1989), "Software Engineering Risk Analysis and Management", McGraw-Hill, New York.

[9] Cusumano, M., and Yoffie, D. (1999). "Software Development on Internet Time", IEEE Computer, (October), 60 - 69.

[10] Dorofee, A., Walker, J., Alberts, C.J., Higuera, R.P., Murphy, R.L. and Williams, R.C. (1996), "Continuous Risk Management Guidebook", Software Engineering Institute, Carnegie Mellon University.

[11] Down, A., Coleman, M., Absalon, P. (1994), "Risk Management for Software Projects" McGraw-Hill, London.

[12] Glass, R.L. (1998), "Software Runaways - Lessons Learned from Massive Software Project Failures", Prentice-Hall.

[13] Greer, D. and Bustard, D.W., (1997) "SERUM - Software Engineering Risk: Understanding and Management", The International Journal of Project & Business Risk, vol. 1, Issue 4, winter, pp. 373-388, Project Manager Today Publications.

[14] Greer, D., Bustard, D. and Sunazuka (1999a), T. "Prioritisation of System Changes using Cost-Benefit and Risk Assessments", Fourth IEEE International Symposium on Requirements Engineering, pp 180-187, June.

[15] Greer, D., Bustard, D. and Sunazuka (1999b), T. "Effecting and Measuring Risk Reduction in Software Development", NEC Journal of Research and Development, Vol.40, No.3, pp.378-38, July.

[16] Hall, E.M. (1998), "Managing Risk: Methods for Software Systems Development", Addison-Wesley, Reading, MA.

[17] IEC (1995), International Electrotechnical Committee, "IEC 300-3-9: Requirements and Guidelines for the analysis of Technological Risks", International Electrotechnical Committee, Geneva.

[18] Juran, J., M. (1989), "Juran on Leadership for Quality", The Free Press, New York.

[19] Kahneman, D. and Lovallo, D, Timid choices and bold forecasts – a cognitive perspective on risk-taking, Management Science, 39(1), 17-31, 1993.

[20] Karolak, D.W. (1996), "Software Engineering Risk Management", IEEE Computer Society Press, Los Alamitos.

[21] Moynihan, T. (1997), "How Experienced Project Managers Assess Risk", IEEE Software, May/June, pp. 35-42.

[22] Sage, A (1981), "Behavioural and Organization Considerations in the Design of Information Systems and Processes for Planning and Decision Support", IEEE Trans. On Man , Systems and Cybernetics, vol. SMC-11, No. 9, September/October.

[23] Sisti, F. and Joseph, S. (1994), "Software Risk Evaluation method v 1.0", Software Engineering Institute Technical Report, CMU/SEI-94-TR-19, SEI, Pittsburgh, PA., Dec.

[24] Xerox Corporation and Carnegie Mellon University (1992). "The University Challenge: Problem Solving Process User Manual", Xerox Corp., Stamford, CT.. Also described in [10].