REVIEW OF RISK MANAGEMENT METHODS

Robert Stern (MBA), José Carlos Arias (PhD, DBA)

Abstract

Project development, especially in the software related field, due to its complex nature, could often encounter many unanticipated problems, resulting in projects falling behind on deadlines, exceeding budgets and result in sub-standard products. Although these problems cannot be totally eliminated, they can however be controlled by applying Risk Management methods. This can help to deal with problems before they occur. Organisations who implement risk management procedures and techniques will have greater control over the overall management of the project. By analysing five of the most commonly used methods of risk management, conclusions will be drawn regarding the effectiveness of each method. The origin of each method will be established, along with the typical areas of application, the framework of the methods, techniques used by each and the advantages and disadvantages of each of the methods. Each method will be summarised, then an overall comparison will be drawn. Suitable references will be included to highlight features, along with diagrams and charts to illustrate differences in each approach.

1. Introduction

There are various methods that have been developed to analyse the risk factors within any given project. For the purposes of this paper five methods are analysed in detail, they are as follows:

- **BOEHM (Report Section 2.1)**
- **RISKIT (Report Section 2.2)**
- **SEI-SRE (Report Section 2.3)**
- **SERUM (Report Section 2.4)**
- **SERIM (Report Section 2.5)**

Each of the above methods can be used as a very effective business tool in making sure that the risk element of a project is cut down to a minimum, different methods are effective on different types of project, this paper will establish the differences, similarities and effectiveness among the given methods.

Section 2.1 looks at the Boehm method, developed by Barry Boehm. He developed a set of principles and practices for managing the risk of developing software called the risk-analysis paradigm. Boehm's Software Risk Management model focuses on the concept of "risk exposure" as defined by the relationship where the probability of an unsatisfactory outcome and the loss due to the unsatisfactory outcome determine the valence of the risk event.

Section 2.2 analyses the RISKIT approach, which was developed in August 1996 at Maryland University, United States. The RISKIT method is applied mainly in large organisations, and used for IT projects.

Section 2.3 analyses the SEI-SRE method, which was originally developed by the Software Engineering Institute, which is funded by the US Air Force, a division of the American Department of Defence. The method was originally developed as a project management method and the element of risk management was later added to the equation.

Section 2.4 focuses on the SERUM method. It combines the use of the implicit risk management and explicit risk management. Implicit risk management is a method where by the software process is designed to reduce risk. Explicit risk management is a method where the risks are addresses directly by the process of identification, assessment, prioritisation, planning, resolution and monitoring.

Section 2.5 analyses the final method, SERIM. The purpose of SERIM is to enable assessment of risk factors in software development from several different perspectives, and developing focused action plans to manage risks before they become realities.

2. Risk Management Methods

2.1 BOEHM

i. Origin of the Boehm Method

Barry Boehm believes that "Risk Management helps people avoid disasters, avoid rework, avoid overkill, and stimulate win-win situations on software projects" Source: http://www.qaiindia.com/Training/softwarerisk.html Boehm's Software Risk Management model focuses on the concept of "risk exposure" as defined by the relationship where the probability of an unsatisfactory outcome and the loss due to the unsatisfactory outcome determine the valence of the risk event. The method developed by Boehm is the original Risk Management method.

ii. Typical Area of Application of Boehm Method

The Boehm method can be applied to almost any software related project.

iii. Framework of the Boehm Method

Boehm's top ten risk items.

Figure 1

1	**Personnel Shortfalls:** Staffing with top talent; job matching; team-building; morale-building; cross-training; prescheduling key people.
2	**Unrealistic Schedules and Budgets:** Detailed, multisource cost and schedule estimation; design to cost; incremental development; software reuse; requirements scrubbing.
3	**Developing the wrong software functions:** Organizational analysis; mission analysis; operational concept formulation; user surveys; prototyping; early users' manuals.
4	**Developing the wrong user interface:** Prototyping; scenarios; task analysis.
5	**Gold-plating. Requirements scrubbing:** prototyping; cost-benefit analysis; design to cost.
6	**Continuing stream of requirements changes:** High change threshold; information-hiding; incremental development (defer changes to later increments).
7	**Shortfalls in externally-performed tasks:** Reference-checking; pre-award audits; award-fee contracts; competitive design or prototyping; team-building.
8	**Shortfalls in externally-furnished components:** Benchmarking; inspections; reference checking; compatibility analysis.
9	**Real-time performance shortfalls:** Simulation; benchmarking; modelling; prototyping; instrumentation; tuning.
10	**Straining computer science capabilities:** Technical analysis; cost-benefit analysis; prototyping; reference checking.

Source: http://www.cs.concordia.ca/~teaching/comp554/Wint2000/AssessingProjectRisk.htm

iv. Techniques Used in the Boehm Method

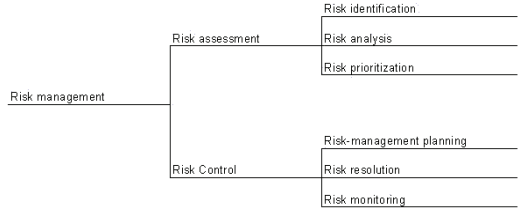
As figure 2 shows, the practice of risk management involves two primary steps, each with three subsidiary steps.

- Risk identification produces lists of the project-specific risk items likely to compromise a project's success.
- Risk analysis assesses the loss probability and loss magnitude for each identified risk item, and it assesses compound risks in risk-item interactions.
- Risk prioritisation produces a ranked ordering of the risk items identified and analysed.

The second primary step, risk control, involves risk-management planning, risk resolution, and risk monitoring:

- Risk-management planning helps prepare you to address each risk item, including the coordination of the individual risk-item plans with each other and with the overall project plan.
- Risk resolution produces a situation in which the risk items are eliminated or otherwise resolved.
- Risk monitoring involves tracking the project's progress toward resolving risk items and taking corrective action where appropriate.

Figure 2



Source: Software Risk Management Steps (Boehm 1991)

Risk Identification

Risk identification is the first step in a comprehensive and continuous risk management method. Most project risks are usually known by project personnel and as a consequence can be managed. The first step to successful risk management is to write down the risks and make them visible to all.

Risk Analysis

The risk exposure is described as a technique for risk analysis and outlines some techniques for estimating the probability and the size of a loss.

Risk Prioritisation

Deal with the most important risks first. There is often a good deal of uncertainty in estimating the probability or loss associated with a risk. The amount of uncertainty is itself a major source of risk, which needs to be reduced as early as possible.

Risk-Management Plans

The focus of risk-management planning is to develop a plan to handle each of the high -priority risks identified during the previous activities. The plan should be documented and oriented around answering the standard questions of why, what, when, who, where, and how.

Risk Resolution and Monitoring

After established a good set of risk-management plans, the risk-resolution process consists of implementing the risk-reduction techniques as identified in the plans. Risk monitoring ensures that this is a closed-loop process by tracking risk-reduction progress and applying whatever corrective action is necessary to keep the risk-resolution process on track.

v. The Effectiveness of the Boehm Method:

This is a method that can be used in all the phases of software development. The method is generally traditional and doesn't handle generic risk implicitly as SERUM does.

vi. Advantages of the Boehm Method

- Relatively simplistic
- Covers all phases of software development

vii. Disadvantages of the Boehm Method

- Doesn't handle generic risk implicitly

viii. Boehm Summary

Boehm developed a set of principles and practices for managing the risk of developing software called the risk-analysis paradigm. Boehm's Software Risk Management model focuses on the concept of "risk exposure" as defined by the relationship where the probability of an unsatisfactory outcome and the loss due to the unsatisfactory outcome determine the valence of the risk event (Boehm B.W. 1991). This model uses a decision tree method to identify the software risk items and top-ten risk identification checklist. The goal of the risk management is to reduce the "risk exposure" associated with the software.

2.2 RISKIT

i. Origin of the RISKIT Method

The RISKIT method was developed in

August 1996 by Jyrki Kontio, Helena Englund and Victor R. Basili at Maryland University, United States. The original reasons behind the development were because of lack of reliable methods that were currently in existence. When risk management methods are used, they are often simplistic and users have little confidence in the results of their risk analysis results. The developers of the RISKIT method believed that the following factors contribute to the low usage of risk management methods in practice:

- Risk is an abstract and fuzzy concept and users lack the necessary tools to define risk more accurately for deeper analysis.

- Many current risk management methods are based on quantification of risks for analysis and users are rarely able to provide accurate enough estimates for probability and loss for the analysis results to be reliable. On the other hand, the table based approaches are often biased and too coarse for risk prioritisation.

- Risks have different implications to different stakeholders. Few existing methods provide support for dealing with these different stakeholders and their expectations.

- Each risk may affect a project in more than one way. Most existing risk management approaches focus on cost, schedule or quality risks, yet their combinations or even other characteristics (such as future maintenance effort or company reputation) may be important factors that influence the real decision making process.

- Many current risk management methods are perceived as complex or too costly to use.

"A risk management method should be easy to use and require a limited amount of time to produce results, otherwise it will not be used. Given the increasing interest in risk management in the industry, we believe that for risk management methods to be applied more widely, they will need to address the above issues. Furthermore, risk management methods should also provide comprehensive support for risk management in projects, they should provide practical guidelines for application, they should support communications between participants, and they should be credible. The RISKIT method was developed to address the issues listed above." (Kontio J, 1997)

ii. Typical Areas of Application of the RISKIT Method

The RISKIT method is applied mainly in large organisations, and used for IT projects. However, the business analysis aspect of the technique could be applied to any project. One of the largest organisations who have made use of the method are Nokia Telecommunications. RISKIT was originally developed for software development projects, but it can be applied in many other areas, such as business planning, marketing and in technology related fields.

iii) Framework of the RISKIT Method

RISKIT follows seven different steps of implementation. Its main characteristics can be described by the following principles.

1. **The RISKIT method provides precise and unambiguous definitions for risks.**

RISKIT method defines risks more precisely and formally. Risk is defined as a possibility of loss, the loss itself, or any characteristic, object or action that is associated with that possibility.

2. **The RISKIT method results in explicit definition of objectives, constraints and other drivers that influence the project.**

When expectations are recognised and defined, we refer to them as goals. While some goals cannot be stated precisely, at least they should be identified and documented as well as the information available allows. RISKIT contains an explicit step and supporting templates to assist in the goal definition.

3. **The RISKIT method is aimed at modelling and documenting risks qualitatively.**

It provides conceptual and graphical tools to model different aspects of risks qualitatively, instead of requiring quantitative estimation of risk probability and impact to take place early in the project.

4. **The RISKIT method can use both ratio and ordinal scale risk ranking information to prioritise risks reliably.**

The estimation problem has been reduced in the RISKIT approach. Instead of forcing quantification of risks using ratio scale metrics often an unrealistic goal the RISKIT method only attempts to accomplish the necessary quantification of risks for risk management to take place.

5. **The RISKIT method uses the concept of utility loss to rank the loss associated with risk.**

Many current risk management approaches are based on ranking of risks based on the loss they cause to some specific attributes of the project, such as cost, time delay, or quality metrics. Often a single metric is used. This can be detrimental for two reasons. First, the use of a single metric, or a small number of metrics, can create strong bias away from secondary, yet influential goals that should be considered. Second, research in economics and management science has strongly indicated that decisions are made based on the changes in the expected utility of alternatives.

6. **Different stakeholder perspectives are explicitly modelled in the RISKIT method.**

All projects have more than one stakeholder, they may have different priorities and levels of expectations. Risk management should be based on the recognition of these stakeholder expectations and priorities.

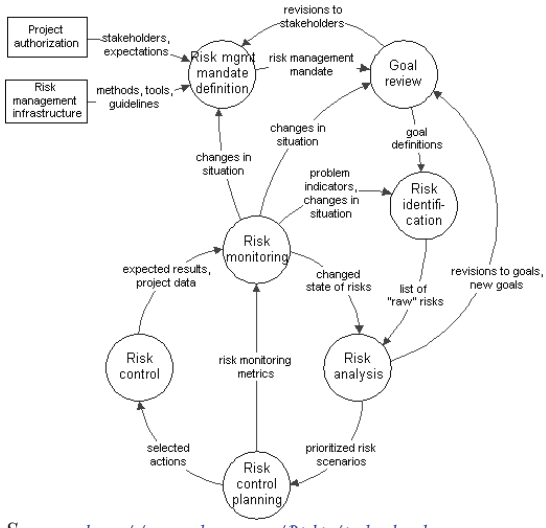
7. **The RISKIT method has an operational definition and training support.**

The RISKIT method has an operational definition so that it can be applied easily and consistently.

iv. Techniques Used in the RISKIT Method

The diagram below describes the techniques used in the methods.

Figure 3 - The Riskit risk management cycle



Source: http://www.rdware.com/Riskit/index.html

- RISKIT uses a graphical method the Risk Analysis Graph (RAG) to chart risk scenario development, which is part of Risk Analysis.

- Data flow diagram symbols are used in the RISKIT graphs.

- Brain storming techniques are used to develop the "raw risk list " which is part of Risk Identification.

- Extensive use of "templates" is made within RISKIT. Templates are structured sets of questionnaires

- The concept of utility loss is used to assess the impact of the risk, a formula is available to calculate this but is open to question as the formula is based on "estimates"

- RISKIT pareto ranking technique is used to prioritise risks according to probability of the risk and impact of the risk

- Five risk controlling options are suggested:-

1. No risk reducing actions
2. Contingency plans
3. Reduce loss
4. Risk avoidance
5. Reduce event probability

v. Effectiveness of the RISKIT Method:

The framework clearly shows that this method can be used for any software development process and gives more accurate results of risk by using the probability theory. However, it fails to cover Small to Medium sized organisations.

vi. Advantages of the RISKIT Method

- Flexibility - originally developed for software development projects, but it can be applied in many other areas, such as business planning, marketing and in technology related fields.

vii. Disadvantages of the RISKIT Method

- The method doesn't bridge the gap between risk estimation and risk metrics, this means it is very difficult to predict the potential risk reliably.

- The method doesn't offer a way to combine and harmonize the different stakeholder perspectives on the risk results.

viii. RISKIT Summary:

- Provides precise and unambiguous definitions for various aspects of risk.

- Results in explicit definition of objectives, constraints and other drivers that influence the project.
- Aimed at modelling and documenting risks qualitatively.
- Can use both ratio and ordinal scale risk ranking information to prioritise risks reliably.
- Used the concept of utility loss to rank the loss associated with the risk.
- Has an operational definition and training support.

2.3 SEI-SRE

i. Origin of the SEI-SRE Method

SEI-SRE (**S**oftware **E**ngineering **I**nstitute, **S**oftware **R**isk **E**valuation) method was originally developed by the Software Engineering Institute, which is funded by the US Air Force, a division of the American Department of Defence. The method was originally developed as a project management method and the element of risk management was later added to the equation. The SEI-SRE Risk Management Methodology provides a framework for evaluating risks, which could prevent project success.

ii. Typical Area of Application of the SEI-SRE Method

The area of application is mainly in the US Department of Defence IT projects, however it can also be applied to large organisations IT projects. Customers have included Xerox Corporation, State of Pennsylvania, Computer Sciences Corporation, US Army, NASA and US Air Force. Source: http://www.sei.cmu.edu/programs/sepm/risk/risk.faq.html

iii. Framework of the SEI-SRE Method

The SRE Method Provides a Framework for Evaluating Risks, in order to establish project success or failure. The Framework consists of mainly:

- **Methodological Concept (developed by SEI)**
  - Risk Management Paradigm
  - Risk Taxonomy
  - Risk Clinic
- **Hierarchical Holographic Modelling**
  - Temporal Discussion
  - Methodological Discussion
  - Human Dimension
- **Objectives of the Method**
  - Identify and Analyse Risks to the Project
  - Prepare High-Level, Strategic Mitigation Plans for Major Risks and Risk Areas
  - Create a Way to Further Define and Incorporate Tasks into the Overall Project Development Plan
  - Address Project Manager Expectations

iv. Techniques Used in the SEI-SRE Method

- Risk Management Paradigm.

- Risk Taxonomy.
- Risk Clinic.

The SEI Risk Management paradigm is depicted below. The paradigm illustrates a set of functions that are identified as continuous activities through the life cycle of a project.

Figure 4

Source: Carnegie Mellon's SEI software risk management model (Sisti 1994)

Figure 5

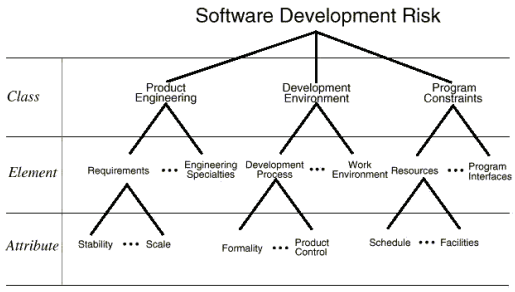
Function	Description
Identify	Search for and locate risks before they become problems.
Analyse	Transform risk data into decision-making information. Evaluate impact, probability, and timeframe, classify risks, and prioritise risks.
Plan	Translate risk information into decisions and mitigating actions (both present and future) and implement those actions.
Track	Monitor risk indicators and mitigation actions.
Control	Correct for deviations from the risk mitigation plans.
Communicate	Provide information and feedback internal and external to the project on the risk activities, current risks, and emerging risks. *Note*: Communication happens throughout all the functions of risk management.

Source: http://www.sei.cmu.edu/programs/sepm/risk/risk.mgmt.overview.html

The SEI-SRE Process

- In the identification activity, a taxonomy-based questionnaire (TBQ) is used to elicit as many risks as possible from the project team.
- The analysis activity is defined as the conversion of risk data into decision-making information. Effectively, this consists of building and evaluating a risk model.
- Planning is defined as the conversion of decision-making information into plans and actions; this includes planning both mitigating actions and also the acquisition of further information concerning a risk, where more information is needed to inform subsequent decisions.
- During tracking, suitable metrics of overall project risk are identified and monitored. Trigger events are identified and mitigating actions are monitored.
- Control consists of correcting for deviations from planned actions; this may involve all the key elements from one through to seven. An additional activity, communication, is identified and is seen as central to all the other activities.

Figure 6



Source: Software Development Risk Taxonomy (Carr 1993)

The Primary Functional Components of the SRE:

- Detection,
- Specification,
- Assessment,
- Consolidation and
- Mitigation.

Detection

Risks are detected using the TBQ.

Specification

The specification of a risk includes its conditions (identifying the circumstances under which the risk may occur), its consequences and its immediate source. This specification may be expressed in informal natural language or represented more formally using a structured syntax.

Assessment

Risk assessment consists of associating a qualitative "magnitude" (or exposure level) with each risk, which is one of "critical", "high", "medium" or "low" and represents the expected value of the risk, i.e. the product of probability and impact.

Consolidation

During consolidation, information obtained from multiple interview sessions or multiple evaluations on a single project is combined. Instances of multiple descriptions of a single risk are identified and any differences or inconsistencies between different parts of the risk information are reconciled.

Mitigation

Mitigation is facilitated by grouping similar risks into "risk mitigation areas". For each risk mitigation area, the current status is analysed and recorded, the desired status is recorded and mitigation strategies and activities are then developed and recorded. A risk map may be drawn up relating risks to mitigation areas and also relating risks to project goals.

v. The Effectiveness of the SEI-SRE Method:

The SEI-SRE method is known to be effective when used within the environments for which it was planned. However the method has not been applied to diverse commercial environments. The method is now available for commercial use but there are no real results available. The method is designed by the SEI, and therefore can be used in any software projects. It is typically used in Military projects in USA. Most of the commercial firms, which are using SRE, are based in the USA. This method is designed primarily for application in the USA but could easily be applied to the European Union.

vi. Advantages of the SEI-SRE Method

- The main strength of SEI-SRE is that it is not purely theoretical, many clients have used it successfully.
- Another important factor is that SRE was developed by the Software Engineering Institute, and can therefore be used in any IT projects.

vii. Disadvantages of the SEI-SRE Method

- In this method everything is defined as a template including the way the interviews are to be conducted. There is no space where human intelligence can be applied.
- Because the method is based on human experience, the results of the analysis may be inconclusive when assessed by the SRE team.

viii. SEI-SRE Summary

The SEI-SRE method of risk identification is more detailed than many other approaches. The method uses established, familiar, and well-tested tools. Therefore, this method is useful for software development organisations as it is very efficient.

2.4 SERUM

i. Origin of the SERUM Method

SERUM (Software Engineering Risk: Understanding and Management) is a risk management technique defined within a software process. It combines the use of the implicit risk management and explicit risk management. Implicit risk management is a method where by the software process is designed to reduce risk. Explicit risk management is a method where the risks are addresses directly by the process of identification, assessment, prioritisation, planning, resolution and monitoring. SERUM emphasis on implicit risk management, so that the generic common risk across all the projects are not to be dealt explicitly. Due to this reason, the number of risk to be dealt explicitly is reduced making risk management easier and handling most of the risks adequately.

ii. Typical Area of Application of the SERUM Method

Premise of the method: "Change is inevitable for all commercial software systems" For any application, there will be a many potential changes. The SERUM method is typically used within software projects, because it is suitable for projects which are contineous and have various release versions and updates it is a method which lends itself to software development.

iii. Framework of the SERUM Method

SERUM is based on two well-defined methods:

- Checkland and Wilson's Soft Systems Methodology (SSM)
- Gilbs Evolutionary Development.
- Soft Systems Methodology - used to feed into step 1 and through to step 5
- Evolutionary Development - used in steps 6-9 and afterwards (outcomes form steps)

SSM in SERUM

- Used for business analysis of the system
- Provides a set of recommendations for change (Changes: organisational and/or technological).
- Recommendations define the gap between the current position and the ideal model.
- May identify major changes to an existing way of working.
- Used for business analysis of the system

SERUM defines the gap between the current position and the ideal model:

SERUM Change Priority

Priority of change is expressed as a function of five variables.

- Risk exposure in the current system,
- Risk exposure in the proposed system,
- Risk exposure in the implementation of a change
- Cost of a defined change
- Benefit of a defined change

Evolutionary Development

- SERUM uses evolutionary development for planning the implementation of developmental changes
- Parts of a system are implemented and delivered in phases.
- The part is evaluated by the client and feedback is used in implementing subsequent phases.
- Crucial to this at the internal level is the development of measurable goals and attributes
- To develop these goals must understand the requirements and goals of the customer

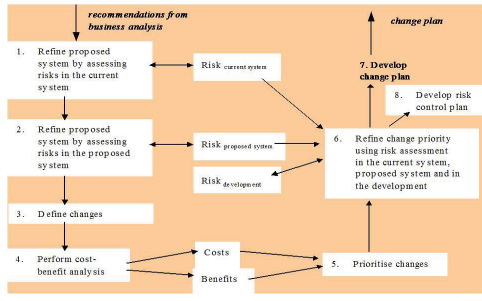
Evolutionary Development emphasises the need for

- Explicit goals,
- Measurable goals and
- Best case/worse case analyses.
- Interaction between developers and customers, users, budget controllers…

iv. Techniques Used in the SERUM Method

The approach is based on a combined cost-benefit and risk analysis of the current system, the proposed system and the change process. It is summarised in the diagram below (Greer D, 1998.)

Figure 7



Source: (Greer D, 1998.)

The process used by SERUM is as follows:

- Refine the proposed system by accessing risk in the current system
- Refine the proposed system by accessing risk in the proposed system
- Define the changes
- Perform the cost benefit analysis
- Prioritise change
- Refine the change priority using the risk assessment for the current system for the proposed system for the development process
- Develop change plan
- Produce risk control plan for accepted risks

v. The Effectiveness of the SERUM Method

Doesn't provide mechanism for ensuring requirements are "correct/ appropriate" other than by applying the "goals and measurements" idea "outwards". Development projects affected by vagueness, his approach gives mechanism by which developers can attack this and go back to stakeholders.

vi. Advantages of the SERUM Method

Particularly applicable to the planning of software, and software version releases

SERUM is a method of risk management that looks at the business aspect of system change and also the technical aspect. It even takes in to consideration the evolution part of the software. SERUM enables software to be released in versions.

Handles Implicit and Explicit risks.

SERUM is based on two well-defined methods mainly SSM and Evolutionary delivery. This is the main advantage of this method, not all of the risks have to be dealt with explicitly, like in other traditional methods.

Considers the Risk in current System as well as proposed system.

A lot of risk assessment methods do not take into account the risks involved in the current system, they just analyse the risk in the proposed system. Risk models should be able to take care of those risks that have been already there in the current system. It would be of no use, if the proposed system has the same amount of risk exposure in the same area as the current system.

Self-Learning

After every release is issued, the user comments are passed onto the developer. The developer then takes this into consideration and implements changes in the next release. This is a feature of SERUM that takes into consideration the importance of meeting the user requirements implicitly.

vii. Disadvantages of the SERUM Method

- SERUM doesn't take into account feedback of similar projects, even if they are in the same organisation. There are possibilities that everything could be documented again unnecessarily.

- SERUM, suggests that after every software release the user input should be taken and changes must be applied to the future release. It is however difficult to have a cost-benefit analysis for all the stages even before the development has begun.

- SERUM tries to cover as many of the risks as possible. This at times may not be manageable. The best way to overcome this would be to cover only the risks with high-risk exposure. This can cause low productivity, as the developers would be working on similar risks over and over again.

viii. SERUM Summary

The SERUM method is typically used within software projects, because it is suitable for projects which are contineous and have various release versions and updates it is a method which lends itself to software development. The SERUM approach is to tackle the changes that provide the best cost-benefit ratio. Priority of change is expressed as a function of five variables. The Method composition is divided up into two sections, Soft Systems Methodology and Evolutionary Development. SERUM is used in making sure projects are developed within the time frame and budget and avoid the risk of project failure. SERUM combines the use of Implicit Risk Management and Explicit Risk Management.

2.5 SERIM

i. Origin of the SERIM Method

SERIM (**S**oftware **E**ngineering **R**isk **I**ndex **M**anagement). The author of SERIM is Dale Karolak. He said, *"The software development process can truly be a jungle, filled with hazards that lie in wait to sabotage your projects. These hazards (risks) are numerous and often complex"*. The purpose of SERIM is to enable assessment of risk factors in software development from several different perspectives, and developing focused action plans to manage risks before they become realities. SERIM takes periodic "readings" on the status of software development projects so there can be a focus on high-priority risk areas. After risks are identified, SERIM helps to develop proactive plans for mitigating risk before they sabotage projects.

ii. Typical Area of Application of the SERIM Method

Because of its simplicity SERIM can be applied to a software project by anyone who is familiar with spreadsheet applications. SERIM uses risk questions to derive numerical probabilities for a set of risk factors. These numerical values are then analysed using spreadsheets that have been programmed with particular statistical equations.

iii. Framework of the SERIM Method

Karolak states that SERIM is based upon a "Just in Time" strategy where *"JIT software addresses the following concepts: risks (instead of inventory) and their contingencies built into the software process should be minimised; management of risks early in the development process will reduce cycle time; risk management will result in a product that has less cost associated with it, and has a better chance in meeting schedule commitments."* (Karolak, 1998). The identification of risks and its management throughout the software development life cycle is the cornerstone of JIT software.

The SERIM approach

The SERIM model recognises a three-step process of problem solving using models to help evaluate risks (Holloway, 1979).

- The first step is to analyse alternatives. Alternatives exist when deciding activities based on risks.
- The second step is to create a model that will evaluate alternatives. The model should help in the decision making process by assessing the alternatives.
- The third step is to make a choice. If a choice is not made, the passing of time will dictate the choices for you.

SERIM assumes that software development management alternatives are always present. The underlying models that are used to analyse alternatives are based on the use of probability trees addressing decision alternatives and the use of probabilities. Therefore the method, in practice, supports only the first two steps above, all decision making is left to the project manager using the software.

SERIM has been implemented as a software tool to support the first two steps, and to record the decisions made and action plans created by the project manager. There are three software modules:

- Project Description: Basic identifying information is entered about a project.
- Project Assessment: the user assigns ratings on a scale of 0 to 10 ("*" for not applicable) to a series of metric questions within ten categories of risk factors.
- Analytical Perspectives: the assessment ratings are used within the different formulae to assess a project's risk scores.

There are five Analytical Perspectives, that can be used to analyse a project, although there is one fundamental perspective (risk factors) and all others are based upon differing weighted profiles of these risk factors.

Risk Factors

Ten risk factors are identified in SERIM. These factors are believed by Karolak to encapsulate the risks associated with software development, based on his experience and the published literature. Each of these risk factors is assessed quantitatively, by the project manager, by assigning values between 0 and 10 to a set of individual questions related to each factor. The lower the score the more risky projects are likely to be.

The factors considered are:

- Organisation
- Estimation
- Development Methodology
- Tools
- Risk Culture
- Usability
- Correctness
- Reliability
- Personnel

iv. Techniques Used in the SERIM Method

The five analytical perspectives used in creating SERIM are:

- Risk Factors: this is the fundamental viewpoint.
- Risk Elements: This analyses how the risk factors impact on Technical Risks, Cost Risks and Schedule Risks.
- Risk Categories: This analyses how the risk factors impact on software Process and Product.
- Risk Activities: This analyses how the risk factors impact on: Risk Identification, Risk Strategy and Planning, Risk Assessment, Risk Mitigation and Avoidance, Risk Reporting, and Risk Prediction. (Within each activity, risks are considered from the operational, strategic, technical, business, industry, and practitioner points of view.) This most closely matches the typical risk assessment and management process.
- Development Phases: This analyses how the risk factors impact on each phase of software development cycle: Pre-Requirements, Requirements, Design, Coding, Testing, and Delivery and Maintenance.

Relationship between Risk Factors and Risk Elements

Software risk factors identify relationships between the risk elements previously identified in terms of items more closely related to software issues. A risk factor can relate to more than one risk element.

Relationship between Risk Factors and Risk Categories

Another way that SERIM slices the data is to categorise it in terms of software product risk and software profess risk (these two aspects are termed categories). The influence of each risk factor is considered to be either major or minor.

Relationship between Risk Factors and Risk Activities

The risk activities that are identified in SERIM map on to the usual risk assessment and management model of:

- Risk Identification.
- Risk Strategy and Planning
- Risk Assessment
- Risk Mitigation / Avoidance
- Risk Reporting
- Risk Prediction

However, the method again evaluates each activity as a simple number using different combination of risk factors' question sets. For instance, to evaluate how difficult risk identification is likely to be the software tool simple calculate the mean of the value of the eighty-one questions in the full question set. A high number would indicate a well-organised, low risk project, a low number would indicate a potentially risky project.

Relationship between Risk Factors and Development Phases

This analyses how the risk factors impact on each phase of software development cycle:

- Pre-Requirements,
- Requirements,
- Design,
- Coding,
- Testing
- Delivery and Maintenance

Again, as for risk activities, the method evaluates the riskiness of the project in any particular phase of the lifecycle as simple number using different combinations of risk factors' question sets. For instance, to evaluate how difficult testing is likely to be the mean is calculated for a set of 42 specific questions. A high number would indicate a well-organised, low risk project that should complete testing without any major problems, whereas, a low number would indicate a potentially troublesome time in testing.

v. The Effectiveness of the SERIM Method:

It seems that despite the claims made for the SERIM approach by its author, the use of SERIM is severely limited. A "one-off" use of the software tool would provide the user with the opportunity to record their opinions about the state of a software development project.

vi. Advantages of the SERIM Method

- Within the SERIM model there is simple use made of basic statistics (mainly through assignment of subjective probabilities and use of means and weighted means). Therefore sets of formulae are used to assess the risks in the complex software development environment
- The author claims that SERIM can be used to monitor risks at any point in the development cycle. SERIM assessments can be performed against a regular schedule for each project, or as seems appropriate to the individual project manager.

vii. Disadvantages of the SERIM Method

- The intention with SERIM is that the perspectives are used by the project/ risk manager to focus on problem areas, and create action plans to improve the project's chance of success. However, SERIM does not provide guidance and advice on how to create and implement these action plans.

- The major drawback of the SERIM method is the lack of explicit guidelines on how to use the information and how to identify the risks that may be lurking in the project.

viii. SERIM Summary

The SERIM method is a simple and flexible way to perform software risk management. It is particularly well suited for small manufacturers that may not be able to use more expensive and complex processes. SERIM overview is as follows:

- Identifies different risks for technical implementation, cost, and schedule

- Predicts risks by software development phases

- Provides a means for corrective action to reduce risks

- Identifies the effectiveness of your software risk management activities

- Measures the risk associated with your software product and process

- Handles multiple projects for analysing software risks

3. Conclusions

After analysing the five methods of risk management it has become clear that the most suitable method depends entirely upon the specific criteria of any given project. The method that may be most suitable to one project may well be the most inefficient in another project. Provided that the best method is selected and implemented by all parties concerned there should be every chance that the element of risk in a project can substantially be reduced.

According to a paper by Dr. Linda H. Rosenberg, Theodore Hammer and Albert Gallo from the NASA Risk Management website, http://satc.gsfc.nasa.gov/support/ASM_FEB99/crm_at_nasa.html, "Most project managers agree that risk management works, but the difficulty lies in actually implementing it, even when required to do so. The risk management plan is often hastily written and then thrown in a corner to gather dust." (Rosenberg, 1999)

The Boehm method uses a Software Risk Management model, which focuses on the concept of "risk exposure" as defined by the relationship where the probability of an unsatisfactory outcome and the loss due to the unsatisfactory outcome determine the valence of the risk event. The main advantages of the Boehm Method are its relatively simplistic and its ability to cover all phases of software development, however the disadvantages include not being able to handle generic risk implicitly. The RISKIT method is applied mainly in large organisations, as it fails to cover Small to Medium sized organisations adequately. The advantages of the RISKIT Method include its flexibility, it was originally developed for software development projects, but it can be applied in many other areas, such as business planning, marketing and in technology related fields. The disadvantages

of the RISKIT Method include the inability to bridge the gap between risk estimation and risk metrics, this means it is very difficult to predict the potential risk reliably.

The SEI-SRE method was originally developed as a project management method and the element of risk management was later added to the equation, the SEI-SRE method of risk identification is more detailed than many other approaches. The method uses established, familiar, and well-tested tools. Therefore, this method is useful for software development organisations as it is very efficient. SERUM emphasis on implicit risk management, so that the generic common risk across all the projects are not to be dealt explicitly. Due to this reason, the number of risk to be dealt explicitly is reduced making risk management easier and handling most of the risks adequately. The SERUM method is typically used within software projects, because it is suitable for projects which are contineous and have various release versions and updates it is a method which lends itself to software development. SERUM has many advantages over other methods, such as BOEHM, it handles Implicit and Explicit risks, as well as considering the Risk in the current system and also the proposed system.

The final method, SERIM enables assessment of risk factors in software development from several different perspectives, and developing focused action plans to manage risks before they become realities. SERIM takes periodic "readings" on the status of software development projects so there can be a focus on high-priority risk areas. The major drawback of the SERIM method is the lack of explicit guidelines on how to use the information and how to identify the risks that may be lurking in the project.

4. References

Boehm B. W, (1991) "Software Risk Management: Principles and Practices." IEEE Software, January 1991, pp. 32-42.

Carr M, (1993) Taxonomy-Based Risk Identification. Software Engineering Institute, CMU/SEI-93-TR-6, June 1993. Available Online http://www.ik.bme.hu/~mohacsi/sqm/documents/risk/risk_management.htm Accessed 17th June 2001

Greer D, (1998) Report on SERUM trial at NEC Corp., University of Ulster.

Holloway, C. A. (1979) Decision Making Under Uncertainty: Models And Choices. Englewood Cliffs: Prentice-Hall.

Karolak, D.W (1998) Software Engineering Risk Management: Finding Your Path through the Jungle. Prentice-Hall

Kontio J, (1997) Available Online http://mordor.cs.hut.fi/~jkontio/riskittr.pdf Accessed 10th June 2001

Rosenberg, 1999 "Continuous Risk Management at NASA" Applied Software Measurement / Software Management Conference, February 1999, San Jose, California, Available Online http://satc.gsfc.nasa.gov/support/ASM_FEB99/crm_at_nasa.html Accessed 17th June 2001

Sisti, F. J. and Joseph, S (1994) Software Risk Evaluation Method, Technical Report CMU/SEI-94-TR-19, Software Engineering Institute, Carnegie Mellon Uni, Pennsylvania.

Other Web References:

http://www.qaiindia.com/Training/softwarerisk.html Accessed 12th June 2001

http://www.cs.concordia.ca/~teaching/comp554/Wint2000/AssessingProjectRisk.html Accessed 15th June 2001

http://www.sei.cmu.edu/programs/sepm/risk/risk.faq.html Accessed 13th June 2001

http://www.sei.cmu.edu/programs/sepm/risk/risk.mgmt.overview.html Accessed 15th June 2001

http://www.rdware.com/Riskit/index.html Accessed 17th June 2001