Value Based Requirements Risk Management (VRRM) Model Simulation

A Thesis Presented to

Department of Software Engineering
Faculty of Basic & Applied Sciences

In Partial Fulfillment

of the requirement for the degree

Of

Master of Sciences (Software Engineering)

By

Shafqat Hussain Majoka

(71-FAS-MSSE/F05)

Department of Software Engineering
Faculty of Basic & Applied Sciences
International Islamic University Islamabad

1- Risk management

D.E
Ip
5.2.11

International Islamic University, Islamabad
Faculty of Basic & Applied Sciences
Department of Software Engineering

FINAL APPROVAL

It is certified that we have read the thesis, entitled **"Value Base Requirement Risk Management (VRRM) Simulation's Model"**, submitted by Shafqat Hussain Majoka Reg. No. 71-FAS/MSSE/F05 .It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for MS Degree in Software Engineering.

ROJECT EVALUATION COMMITTEE

External Examiner:

Dr. Awais Majeed
Assistant Professor, Computer Science Department,
Military College of Signals (MCS),
National University of Sciences & Technology,
Rawalpindi, Pakistan.

Internal Examiner:
Mr. Shahbaz Ahmed
Assistant Professor, Software Engineering Department,
International Islamic University,
Islamabad, Pakistan.

Supervisor:
Dr.Naveed Ikram
Associate Professor, Software Engineering Department,
Faculty of Basic and Applied Sciences,
International Islamic University,
Islamabad, Pakistan

Abstract

Simulation is one of the interesting and powerful tools for peeping into the after effects of any activity or process, before actually running that process. The importance of simulation especially augments, due to the increase in complexity and involvement of more technicalities in process models.

In simulation one develops an actual model of reality by using different notions and actions. This enables us to examine alternatives without actually really undertaking them. The number of researcher and scientists engaged in the simulation and gaming of social systems, it grown tremendously the last twenty years.

This research is intended to simulate latest researches in field of Requirement Risk Management, a process model called Value based requirements risk management (VRRM) was simulated. VRRM introduces the concepts of "value" in risk management process. As being the latest research, model has been implemented only once in the industry on small scale project and the claims made by VRRM still need to be validated.

A systematic approach was followed to attain the desire objective. Simulation techniques were studied and an appropriate technique was selected according to the nature of VRRM. At next step different simulation tools were analyzed and an appropriate tool was selected based on the different factors required for simulation of VRRM. According to the protocols defined by the selected tool a reference simulation model was proposed to bridge the gap between actual VRRM and GoldSim, after that a practical simulation model was developed by mapping all the activities of reference VRRM to a simulation model. Then simulation model was run under different scenarios to collect data for analysis to verify the claims made by authors of VRRM.

Acknowledgment

I would like to thank **Dr.Naveed Ikram** for his continued support and guidance for the completion of this work. He was always available to help and resolve the research issues. His leadership skills helped me to complete the work which was pending since long.

I express my gratitude to the colleagues in both companies whose unprecedented support helped me understanding the complex metrics and generating comparable results. I am especially very thankful to friend Summair Raza for his continuous moral support for the completion of my thesis work.

I am very thankful to my research group member, Abdul Basit, although he is working on separate topics but his moral support was commendable.

Last but not least my family whose continued moral and financial support helped me in completing this long journey.

Declaration

I hereby declare and affirm that this thesis neither as a whole nor as part thereof has been copied out from any source. It is further declared that I have completed this thesis entirely on the basis of my personal effort, made under the sincere guidance of our supervisor. If any part of this report is proven to be copied out or found to be a reproduction of some other, we shall stand by the consequences. No portion of the work presented in this thesis has been submitted in support of an application for other degree or qualification of this or any other University or Institute of learning.

Shafqat Hussain Majoka
71-FAS/MSSE/F-05

Dedication

I would like to dedicate my work to

***ALMIGHTY ALLAH*,**

Who has always showered His endless blessings upon me;
I also dedicate this work to my

FAMILY AND FRIENDS

Whose sincere prayers and love were a source of strength for me
and made this project successful.

Table of Contents

Table of Figures

List of Tables

Chapter- 1

Introduction of Simulation

1 - Introduction of Simulation

Software processes simulation modeling (SPSM) gaining growing interest in the software engineering community, social science and economics and academia after great success in industries like aerospace (Pagendarm & Posts, 1994) (Tezduyar, Aliabadi, Behr, Johnson, Kalro, Litke, 1996) and power generation (Christie, Brown, Joe, 1980) (Weber, Overbye, Sauer, DeMarco, 1998). It has been used to address a variety of issues from the strategic management of software development to supporting process improvements, to software project management, training (Marc, Raymond & David, 1999).

During last decade simulation got importance in academia and software engineering research community. At present it's widely accepted scientific research method to explore complex problems when empirical method is expensive. In software engineering there are large number of variables with different degrees of importance depending on the process lifecycle and organizational maturity (Eickelmann, 2001). It is much difficult and costly to experiment and study all these factors. The "*process model is a formal abstraction of the real world software engineering process which emulates the essential relationships among activities, inputs and outputs of the sub processes*" (Eickelmann, 2001) and to study key variables in software engineering domain. (Marc, Raymond & David, 1999) categorize the use of process modeling into following purpose.

- Strategic management;
- Planning;
- Control and operational management;
- Process improvement and technology adoption;
- Understanding; an d Training and learning.

1.1 What is Simulation

Simulation is used to creating a model of an existing or proposed system in order to identify and understand those factors which control the system or to forecast the future behavior of the system.

There are number of definition of simulation model existing in literature but most popular definition by Madachy, R. A. (1994).

"Simulation is the numerical evaluation of a model, where a model consists of mathematical relationships describing a system of interest".

1.2 Types of Simulation

Following section describes different types of simulation which can be used to simulation any process model.

1.2.1 Stochastic Simulation

Simulation models that contain probabilistic components are called stochastic. *"Stochastic modeling recognizes the inherent uncertainty in many parameters and relationships and stochastic variables are random numbers drawn from a specified probability distribution*" Eliza, C. (2003). In stochastic model output parameter values may vary due the variation in the value of input parameters or intermediate (internal) model variables (M. Müller., Dietmar. Pfahl. 2008).

1.2.2 Deterministic Simulation

In deterministic simulation, Model's input variables and output variables are related (Madachy, 1996). Deterministic simulation models consist of interlocking systems of differential or difference equations, initial condition and output parameters. It has lot of usage in environmental, engineering and policy-making disciplines. Deterministic models are useful approach of reality that is easier to develop and interpret than a stochastic model (D. Poole., AE. Raftery. 2000).

1.2.3 Dynamic Simulation

System dynamic is simulation technique introduced by Forrester during mid-1950s. The purpose of dynamic simulation was to model complex continuous systems for improving management policies and organizational structures (Madachy 1996). Models are devised using continuous quantities interconnected in loops of information feedback and circular causality (Madachy 1996). In system dynamics problem are defined in terms of graphs over time, and determined for an endogenous, behavioral view of the significant dynamics of a system such that the model can reproduce the dynamic problem of concern by itself. Ultimately, understandings and applicable policy insights are derived from the resulting model and corresponding changes are implemented (George, 1991). Abdel-Hamid has developed a generic system dynamics model of software development (Abdel-Hamid & Madnick, 1991). Others have modified it to support project and process management in particular organizations (Lin, Abdel-Hamid & Sherif 1992). Details of the different dynamic models and their uses are provided in (Madachy, 1994).

1.2.4 Discrete Event Simulation

Discrete Event Simulation (DES) basically considers behavior of a system as a discrete set of tasks that are normally performed in series. Time advances in a simulation of the model when a discrete event occurs. In DES models, objects go by process-blocks while simulation is running. Every process has a unique input and creates unique output products. The main benefit of using DES for software process modeling is that all items can have unique attributes and unique values for some common attributes if any. Normally, all software work products have some unique characteristics like complexity, size and quality, among others. Moreover, software projects normally define their processes in discrete terms like analysis, design, code, and test; so typical software process looks like a set of discrete events. So to model a project for management and estimation, the modeling approach should consider these discrete event capabilities. However in discrete event simulation, time clock increase in case of event occurrence.

1.2.5 Hybrid Simulation

The software process consists of a number of activities (e.g., the development of the specification document, the change management, designs modification on the basis of early discovered defects, etc.). Activities may be parallel or sequential. Activities coordinate through the exchange of artifacts to give the final product. All these activities need resources (e.g. personnel, computer time, etc.) to accomplish the desired job and may cause conflict on the use of same resources. The software process therefore exhibits both discrete system aspects along with continuous ones (resource usage by an activity, milestones achieved, quantity of discovered defects). The hybrid two-level approach (Madachy, 1996) elaborates the two levels of abstraction, a higher abstraction level and a lower abstraction level, to represent the software process by a combination of three methods: the analytical, the continuous and the discrete-event method. At the higher abstraction level, the process is modeled by a discrete-event queuing network, consisting of the components, activities, their interactions, and the exchange of artifacts. The queuing model is a synonym of the software process, where service stations represent activities, and circulating customers represent artifacts that move among activities. Each activity in turn can be described by a set of service-stations to represent the component sub-activities. Some activities/sub-activities may require resources (e.g. personnel, time), whereas others may be intended to perform co-ordination tasks only, simulation of managerial policies (e.g. whether to release or not to release a process artifact to the next stage based on its quality level). At lower abstraction level, the implementation details of the service stations (i.e. activities) that were introduced at the higher abstraction level are provided. Here both the analytical and continuous methods are used. Particularly each activity is modeled either by using an analytical average-type function, or by using a continuous type time-varying function, or combination of both. Such type of functions expresses the amount of resources, or time, or effort used by various service stations to simulate the corresponding activities or sub-activities.

1.3 Research Motivation & Justification

One of the primary advantages of simulation is that it provides users with practical feedback when designing real world systems. This helps users improve the efficiency and accuracy of design before constructing it. Users can also explore the merits of alternate system without physically building the system.

Simulation helps designers to study a problem at different level of abstractions. At higher level of abstraction the designer can study the behavior and interactions of the components within the system.

Simulation can also be used for teaching / demonstrating concepts to students, we can show the dynamic behavior of all the system components, though the simulation in this research was basically used as tool to demonstrate the behavior of VRRM (Samad, Ikram, &

Muhammad, 2008). VRRM process was basically designed to manage requirements related risks in a value based manner.

An effective risk management is critical while developing software systems, where requirements' risks are proven to have highest contribution as every requirement contributes towards achievement of some value for some stakeholder and so it's important to manage requirement risks effectively and efficiently. VRRM bears the designed following the guidelines provided by IEEE standard for Risk Management (IEEE Std. 1540-2001) and conforms almost fully to the standard's requirements. Similarly its covers on most part the requirements of CMMi model. The claims of VRRM were verified on the basis of case study by (Basit, Murtaza & Ikram, 2009) on very short duration projects. However the practical implementation of VRRM on large scale project was still missing to verify the claims. So Inspired by the above mentioned benefits of simulation, I developed a simulation model to validate claims made by VRRM's authors on large scale projects.

1.4 Research Question

In software and information business, theory of value is comparatively new and getting considerable interest in research community in recent years. There is a shortage of research literature on this concept due to its very new nature. The Value Based Requirement's Risk Management (VRRM) Process Model is presented during the year 2008. It introduces a new concept of value in the process model of Risk Management that is based upon the IEEE standards for risk management (IEEE Std. 1540-2001). Also, it comprises of almost all activities considered mandatory by IEEE and CMMI Model (Basit, Murtaza & Ikram, 2009).

The primary focus of our research is to validate claims made in VRRM process proposed in (Basit, Murtaza & Ikram, 2009) using simulation modeling technique. Software process simulation plays a significant role in achieving higher level of process maturity and improvement. (Christie, 1999) argues that CMM-based process improvement can benefit from process simulation because it provides insights into complex process behavior; moreover simulation can help to tackle different questions on CMM levels. "*Unfortunately traditional process definition does not shed much light on these behavioral issues and the usual way to resolve them is to run the actual process and observe the consequence that is not cost effective to perform process improvement*" (Christie, A.M. 1999).

Problem statement of this experimental study is given as under:

- Do claims made in VRRM process model hold?
- What further improvements can be made in VRRM?

1.5 Expected Outcome

- Simulation Model of Value Base Requirement Risk Management (VRRM) process.
- Analysis of data collected in different simulation run and recommendation to improve VRRM.

Chapter- 2

Literature Survey

2 - Literature Survey

2.1 Simulation in Software Engineering

Software engineering, right from its emergence requires sound analysis and strong planning for developing software; the need becomes more critical in case of complex and large applications where a small mistake can lead to failures and derailment of projects. The main reason of project disaster is poor planning (Metzger, Boddie, 1996). Over the last few decades the software engineering has badly suffered several schedule and cost overruns as well as poor product quality (Marc, Raymond, David, 1999). Although the high level languages, different advance tools, techniques and development methodologies have helped software developers to overcome the problem but how these tools, techniques, methodologies and languages can be effectively merge into a successful software requires a exact, thorough and detailed planning and with deep insight of future software development activities and contingency plans that need to be adapted in case of any derailment in schedule, cost and expected plan. One solution in this situation that has also gained some success is software simulation.

2.1.1 Project Management

Software project management encompasses the knowledge, techniques, and tools necessary to man the development of software products (James & Harvey, 1989). The ever increasing size and complexity of software development requires the managers to have extra skills to manage critical situations e.g. in case a risk becomes evident, in case projects runs out of budget etc. such critical situations are ever difficult to handle because of their "being unpredictable" nature. Simulation helps project managers have a dry run of the project, consider "What-if" scenarios for accessing the effects of different management policies (Bengee, James, 2004) , so that an initial concept and pre-pattern of "healing activities" and their alternatives can be devised. The overall effect of these activities can be analyzed and their effectiveness can be studied. Having such a run the managers can be at ease to smoothly implement any of such activity in case it is so that in case it is needed.

In multi-project environment, where organizations have multiple projects running in parallel, simulation comes as helping tool. For example, '*what will happen to project A's completion date if project B is delayed? What are the effects of resource contention on individual* *projects? And what are the consequences of the high turnover rate of resource C on the projects that utilize it?"* (Bengee, James, 2004). Simulation helps to analyze and make pre-decisions regarding such questions before actually running the real facts.

2.1.2 Risk Management

Software development involves high risk profile and hence requires risk management processes to jump into cater the potential risks in case they occur and effect product delivery, cost and quality. The important of software process simulation is the management of software development risks (Marc, Raymond & David, 1999) usually discussed within the category of project planning/management. Simulation can help software project managers analyze the impact of risks on project and make preemptive measures whenever needed during the development phase of project.

2.1.3 Process Engineering

Process engineering focuses on the design, inputs, operation and outputs of a certain software process, and Simulation is becoming indispensable tool for analysis and design of a software process (N.T. Koussoulas., S. Daskalaki. 1993). It is directly/indirectly associated with software process improvement (David, 1999). Organizations can use software process simulation at different levels of maturity to obtain significant benefits (David, 1999). Simulation helps software engineers to understand, plan, control and improve software processes at all levels. Regarding a process, simulation helps organizations:

- To improve organizational decision making
- Provides Justification for process improvement initiatives (Raffo, 1996)
- Supporting quantitative prediction of project level performance in terms of cost, quality, and schedule (Paulk et al, 1993 , Raffo et al , 1999)
- To forecast process change impact/behavior before implementation (Raffo, D.1996, Saurabh. K., Krishnan. B. 2009).

Organizations can get help to improve cost, time and schedule. (Zhang, Kitchenham, & Jeffery, 2007) proposed a framework to adopt simulation for assistance of achieving higher level of process maturity.

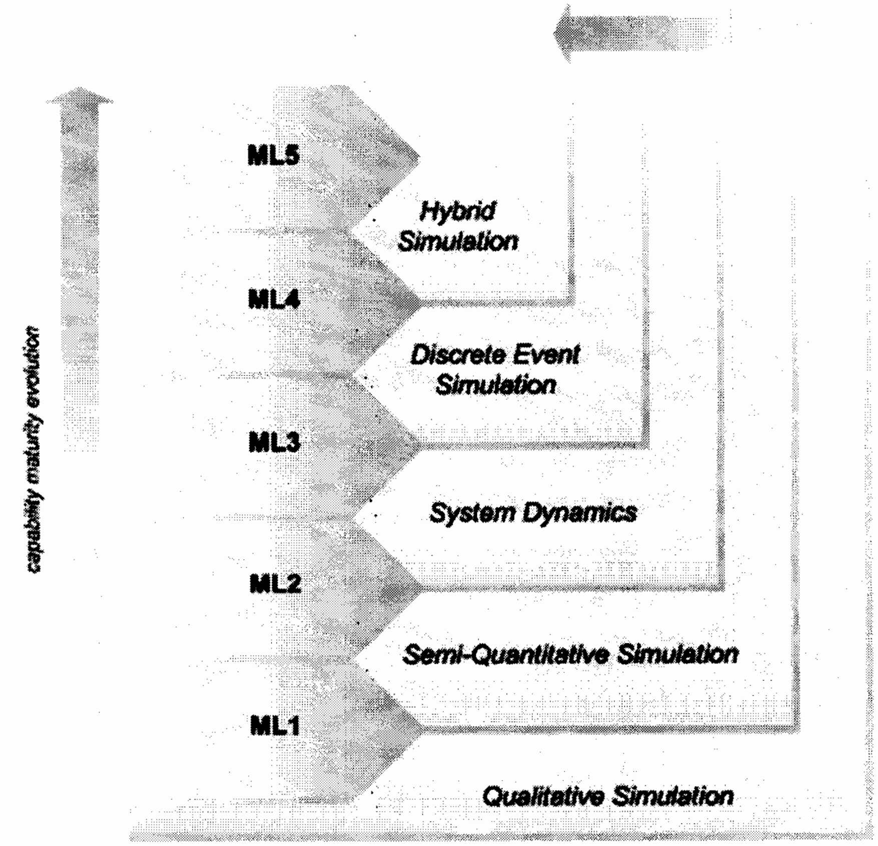


Figure 1: Framework of adopting process simulation modeling in CMMI organizations (Zhang, H. et al. 2007)

All the discussed benefits are not only for higher maturity level organizations but low maturity organizations can also equally benefit if simulation is properly implemented.

2.1.4 Requirement Engineering

Simulation serves as a major helping hand in for highlighting software system requirements early in the software development lifecycle, particularly for examining temporal behavior (Christie, 1999). Simulation can mimic the performance characteristics of software components and their interactions, the effects of time delays and feedbacks, and of finite capacities and resource bottlenecks (Zhang, Kitchenham, & Jeffery, 2007) (Bengee, James, 2004). Thus simulation is an important tool for not only defining the requirements at initial stage but also but can also be used for testing the alternative modifications, before going for their actual implementation. Finally, a system simulation can be used an integral part of requirements which provides the quantitative measures that the software must abide by.

Chapter- 3

Research Methodology

3 - Research Methodology

The main objective of this research is to analyze behavior and validation of claims made in value base requirement risk management (VRRM). Simulation research is an ideal methodology where an in depth analysis is required to understand behavior of process model in different scenarios prior to implementation in production environment. In my study I want to develop simulation model of VRRM to validate its claims.

3.1 Research Methodology Design

This experimental study was conduct into three phases. Unit of analysis was the data collected from different simulation runs.

3.1.1 Phase I – Analysis of VRRM

The aim of this phase was to get an understanding of conceptual value based requirement risk management model and analysis of:

a. The information flow in "Value Base Requirement Risk Management" Model

b. Input/Out parameters all activities and sub-activities of model.

c. Case Study Analysis of VRRM (Basit, Murtaza & Ikram , 2009).

3.1.2 Phase II – Simulation Tools Analysis

The purpose of phase-II was to conduct:

a. Analysis and selection of appropriate technique to simulate VRRM.

b. A criteria was prepared to select the simulation tool.

c. Designed a reference model to bridge gap between VRRM conceptual model and simulation tool.

3.1.3 Phase III – VRRM Simulation Model

Simulation model of VRRM was developed in tool selected in phase - II and validation of model was secondary portion of this phase.

3.1.4 Conclusion and Recommendations

Conclusion and recommendations list were proposed on the basis of data collected during different Run/Scenario of Model.

Chapter- 4

VRRM Analysis

4 – VRRM Analysis

VRRM Process Model was jointly proposed by (Samad,Ikram, & Muhammad,2008). It is a Risk Management Process for software requirements mainly based upon IEEE Std. 1540-2001 Risk Management Process and fully conforms to the CMMI and covers almost all he activities that are important, useful and purposeful (Samad,Ikram, & Muhammad,2008). VRRM Process Model functions by considering the concept of "value" in the Software Requirements owned by different stakeholders. A term 'Success Critical Stakeholders' (SCS) is introduced in this regard to value the requirements and their related risks in order to ensure that success driven stakeholders participate effectively in the software development projects.

VRRM Process Model is represented at two levels of abstraction. The first level is called as Abstraction Level-1 and the second level is referred as Abstraction Level-2. At the higher level, the Abstraction Level-1 comprises of Management and Assessment & Mitigation of risks. The Management part contains the Planning and Monitoring and Control. The Assessment and Mitigation part comprises of Identification, Analysis and Treatment of Risks for a software development project.

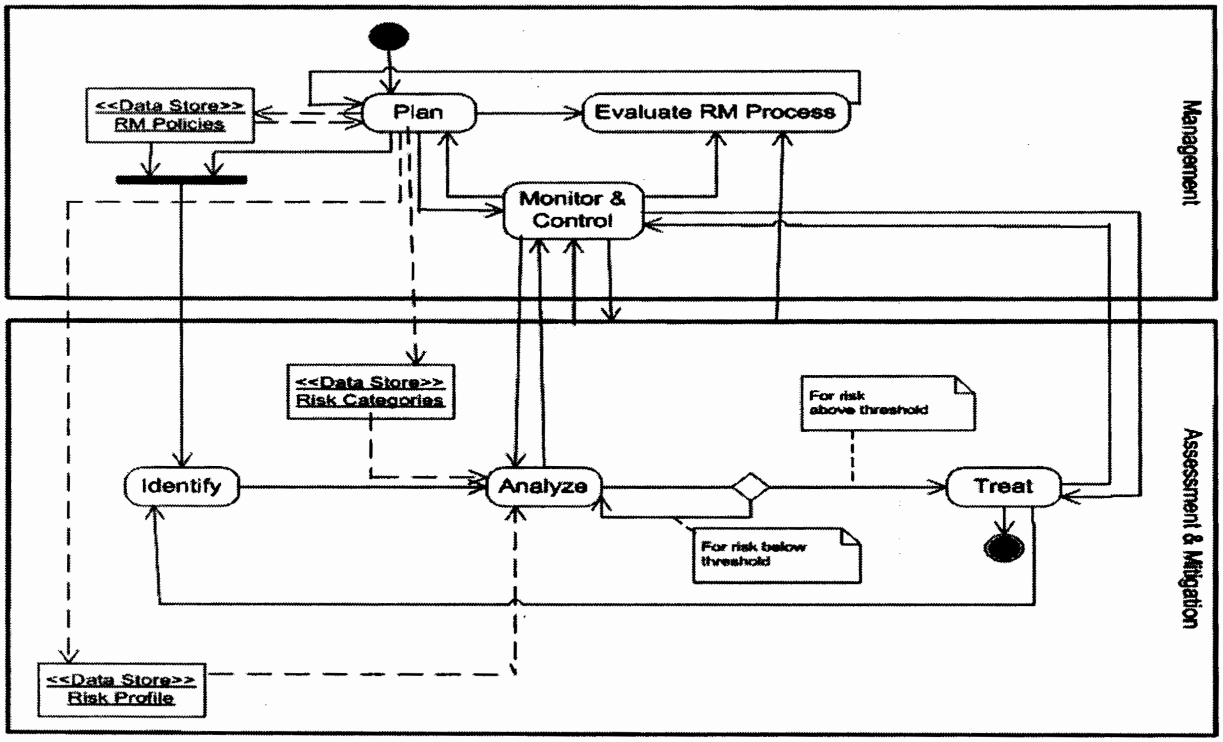


Figure 2: VRRM abstract level 1 (Samad et al. 2008)

Abstraction Level-2 illustrates the complete set of activities divided into six major categories.

Plan (Samad et al. 2008):

1) The process starts with planning. The input to Plan RM Process is the organizational risk management policies and the process evaluation results from past. The output of this activity is Risk Management Plan which is then further used as input to many succeeding activities.

2) Using this plan, roles & responsible parties are defined and resources are planned. The criteria for risk attributes, scales and measures, thresholds and risk categories is also planned and defined here. All this information is stored in their respective data stores. The only artifact produced here at this stage is risk management plan.

3) Process evaluation is done in Evaluate RM Process activity. This evaluation is performed according to the strategy planned in Define Evaluation Process activity. All the evaluation results will be captured in data store Evaluation Results. This data store will have the evaluation results and the lessons learned from previous projects too for reference for current projects.

Identify Risk (Samad et al. 2008):

1) The risks are identified in Identify Risk activity. This activity is put under continuous Monitor and Control as risk identification is a continuous process and risks can occur at any time. Also some entity can be a risk at some times but cannot be the risk at some other time.

2) The identified risks are then categorized according to the criteria set during planning activity. The categorization of risk activity can also point out some other risks that have not been identified yet.

Analysis (Samad et al. 2008):

1) The likelihood and consequences of each risk item is then calculated.

2) As VRRM Process is focused on requirement risks, the respective requirement is linked with the business objective that it is assumed to fulfill.

3) Afterwards the success-critical-stakeholders SCSs are identified.

4) These SCSs then assess the value of the subject requirement and the value of risk identified w.r.t. that requirement. The SCSs assess the value in all three perspectives TOP

individually. The net value for that particular risk item is than calculated by taking aggregate of values of all stakeholders in TOP perspectives.

5) This net value plus the thresholds and consequences of the risk item are then used to evaluate the risk against thresholds defined during planning.

Treatment (Samad et al. 2008):

1) For risks that are above threshold, the contingency plan is developed.

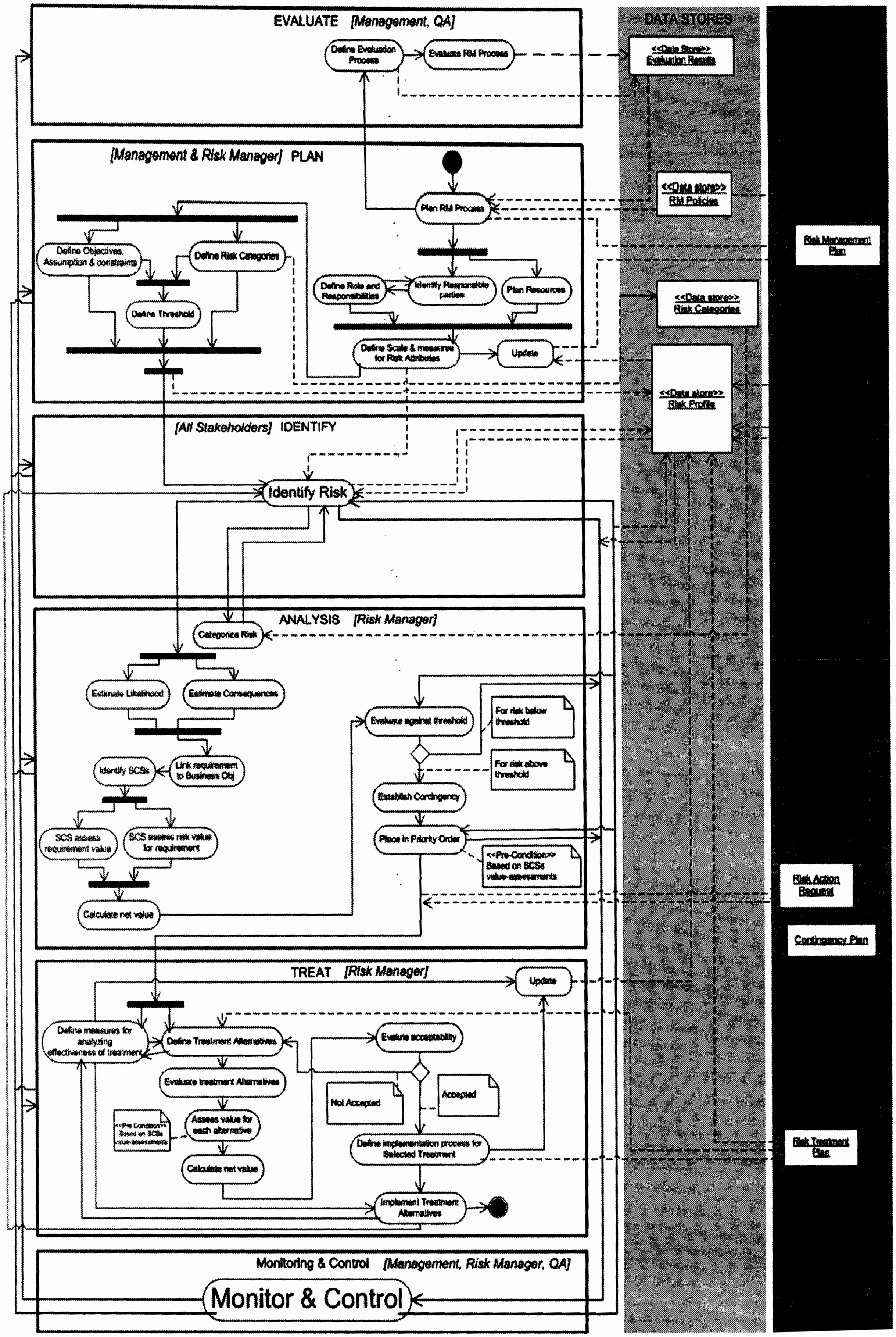
2) These risks are then put in priority list for treatment. The priority ordering of the risk items is done on basis of SCSs value-assessments; done in same manner as for risk acceptance. This value assessment for priority ordering is also done in TOP perspectives and net value is calculated as aggregate of values of all SCSs in all three perspectives

3) The top-prioritized risk item is then passed on for treatment.

4) The treatment alternatives for top-priority risk item are defined and made available to the relevant parties in Define Treatment Alternatives activity. These treatment alternatives are then evaluated for each risk item and the value for each alternative is calculated in same way as for risk acceptance and prioritization in TOP perspectives. The final selection is made on the basis of net-value of SCSs evaluated with the help of negotiations, either against some pre-defined or on-spot criteria.

Monitoring & Control Samad et al. 2008):

1) Risks that are below threshold are continuously Monitored and Controlled as some risk item that is below threshold at some time can exceed the threshold at some other moment.


EVALUATE [Management, QA]
Define Evaluation Process
Evaluate RM Process
DATA STORES
<<Data Store>> Evaluation Results
[Management & Risk Manager] PLAN
Plan RM Process
<<Data store>> RM Policies
Risk Management Plan
Define Objectives, Assumption & constraints
Define Risk Categories
Define Threshold
Define Role and Responsibilities
Identify Responsible parties
Plan Resources
<<Data store>> Risk Categories
Define Scale & measures for Risk Attributes
Update
<<Data store>> Risk Profile
[All Stakeholders] IDENTIFY
Identify Risk
ANALYSIS [Risk Manager]
Categorize Risk
Estimate Likelihood
Estimate Consequences
Evaluate against threshold
For risk below threshold
For risk above threshold
Link requirement to Business Obj
Identify SCSs
Establish Contingency
SCS assess requirement value
SCS assess risk value for requirement
Place in Priority Order
<<Pre-Condition>> Based on SCSs value-assessments
Risk Action Request
Calculate net value
Contingency Plan
TREAT [Risk Manager]
Update
Define measures for analyzing effectiveness of treatment
Define Treatment Alternatives
Evalute acceptability
Evaluate treatment Alternatives
Not Accepted
Accepted
<<Pre Condition>> Based on SCSs value-assessments
Asses value for each alternative
Define implementation process for Selected Treatment
Risk Treatment Plan
Calculate net value
Implement Treatment Alternatives
Monitoring & Control [Management, Risk Manager, QA]
Monitor & Control

4.1 VRRM Case Study Analysis

A case study was presented by (Basit, Murtaza & Ikram, 2009) in practical to highlight the issues and pitfalls and to verify the conclusions made by author of VRRM. Case study implements all the phases of VRRM Model in two individual organizations. Each phase of case study is analyzed and presented here with the help of business process modeling notations along with the inputs, processing and outputs of each activity. Figure 4 is high level diagram of the case study which shows the flow of information in different activities (Planning, Identification, and Analysis, Treatment and Monitoring and control). This diagram shows that:

1. Process starts from planning activity.

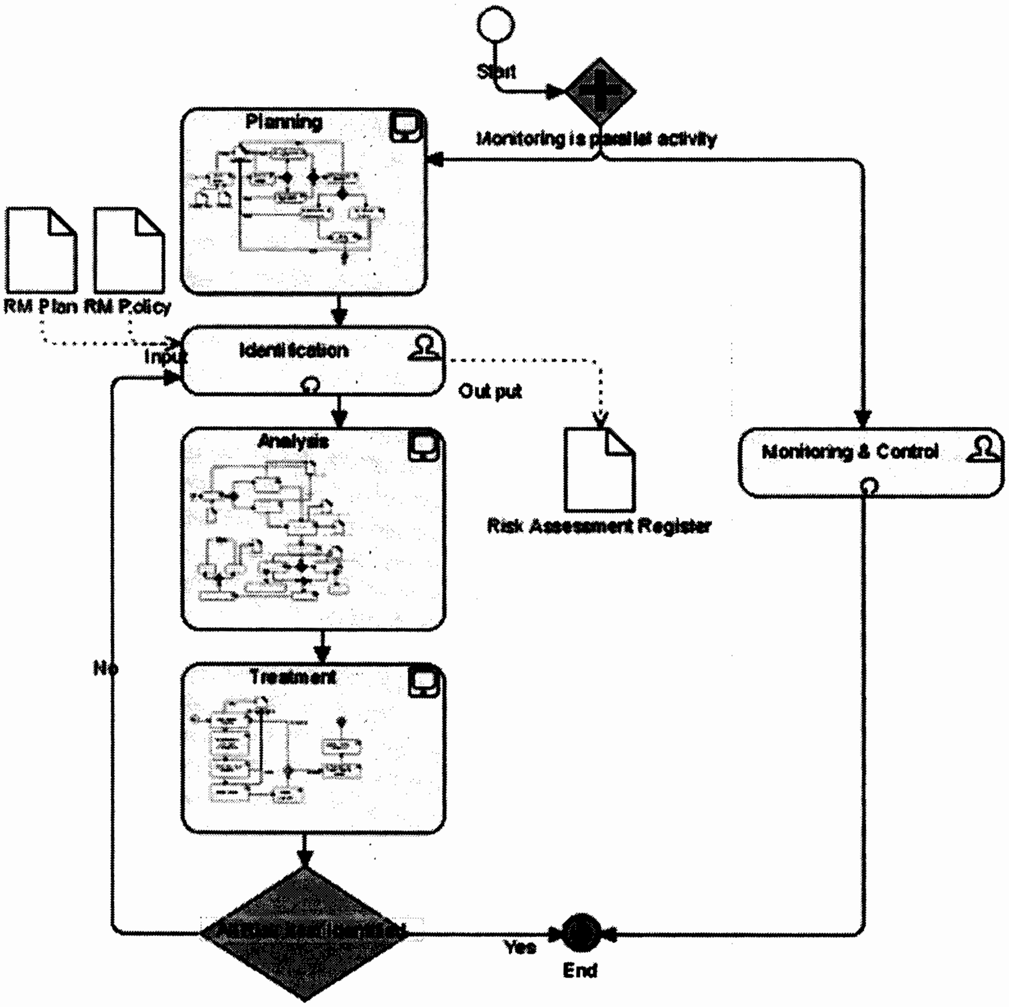


Figure 4: Case study high level diagram

2. Monitoring & Control is parallel and loop/continuous activity.

3. Identification starts after planning and it is also loop/continuous activity.

4. Analysis activity starts after the identification activity to analyze the risk item

5. Treatment activity is carried out once the analysis of risk item completed.

Planning:

VRRM process starts from planning activity. The risk management plan is part of the project plan and it gives the overall risk management process overview that how it was implemented, how the activities was be carried out, who was responsible for which activities and how was be the process evaluated for improvement purposes. The Figure 5 is detail diagram of planning activity which shows flow of information into different eight sub activities. Business process modeling notation are used to draw high level diagram and Plus sing (+) in diagram is used to depict the parallel flow of information into different activities.

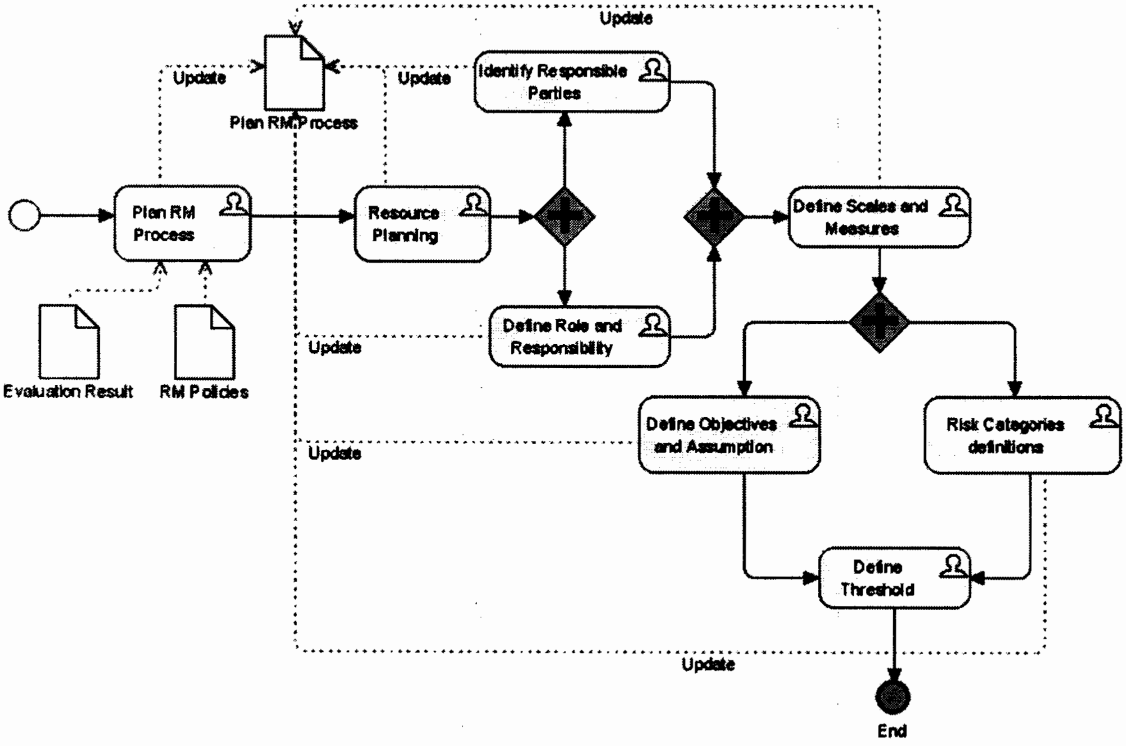


Figure 5: Planning phase diagram of VRRM case study

The Table-1 shows the details of input /output parameters of each sub activity which is derived from (Basit et al. 2009) case study along with the their actors.

Activity			
Plan RM Process	Risk Management Polices , Evaluation Results	Risk Management Plan" (Annexure B)	Project Team

Resource Planning	Resource Pool Information	Resource Management Plan; Resource Matrix (Annexure B)	Project Management Team
Identifying Responsible Parties	Stakeholders Analysis	Responsible Parties	Project Manager , Risk Management Team
Defining Role and Responsibility	Resource Management Plan	Roles and Responsibilities	Risk Management Team, project management team , Project manager
Defining Scale and Measure	Input Data	Scale and Measures	NiL
Define Objective and Assumption	Policies and Guidelines	Objectives and Assumptions	NIL
Risk Category Definition	Risk Register	Definition of Risk Categories	NiL
Define Threshold	Historical Records; Project Information	Threshold Values	NIL

Table 1: Participant in planning phase of case study (Basit et al. 2009)

Identification:

The risks are identified in Identify Risk activity which uses the organizational RM policies and risk management plan from planning phase as input. The detail of input / out and resource is as in table 3 derived (Basit et al. 2009) case study.

Activity	[illegible]	[illegible]	[illegible]
Identification	RM Polices , RM Plan, Project Documentation	Risk Register	Risk Management team

Table 2: Participant in identification phase of case study (Basit et al. 2009)

Analysis phase:

The identified risks are than passed on to Analyze activity. Here these risks are first categorized according to the pre-defined categories and afterwards the likelihoods and consequences of the identified risk items are calculated. See figure 6 and table 4 for detail derived from (Basit, Murtaza & Ikram, 2009).

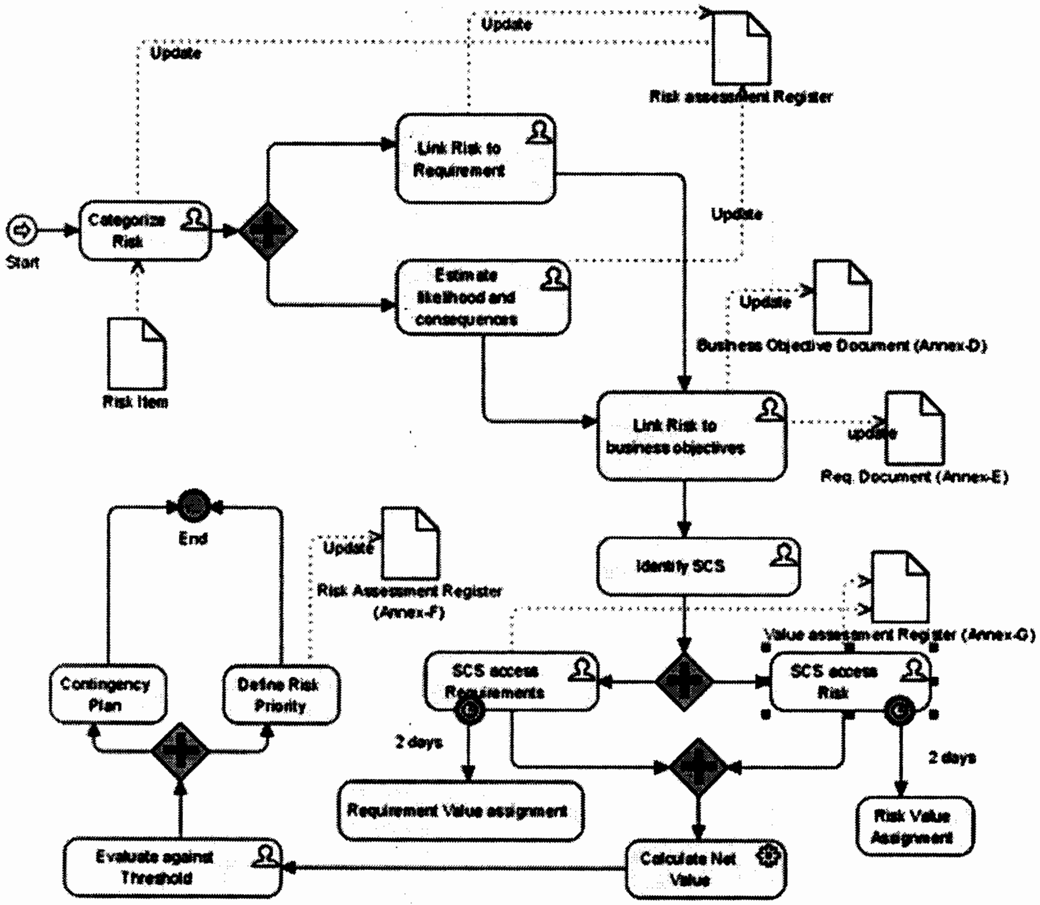


Figure 6: Analysis diagram of VRRM case study (Basit et al. 2009)

Activity			
Categorize risk	Identified risk	Risk assessment register (Annex-F) updated	Risk management team
Estimate likelihood and consequences	Identified risk (after specifying its category)	Risk assessment register (Annex-F) updated	Risk management team, project management team
Link risk to requirements	Identified risk	Risk assessment register (Annex-F) updated	Risk management team, success critical stakeholders
Link requirements to business objectives	Identified risk, associated requirement(s)	Requirement document (Annex-E), business objective document (Annex-D)	Risk management team,
Identify success critical stakeholder	N/A	N/A	Project management team, risk management team, author
Success Critical Stakeholders Assess Requirements' and Risks' Value	Identified risk, associated requirement	Value assessment register	Success critical stockholders, risk management team
Calculate net value	Req. risk values	Value assessment register	risk management team
Evaluate against threshold	Net req. risk value, threshold table	List of risks qualified for treatment. List of risks not qualified for treatment.	Risk management team
Establish contingency	risks qualified for treatment	Contingency plan	Risk management team, project management team or members of organization
Place in priority order	Risks qualified for treatment, Req. risk net value	Risk assessment register (Annex-F)	Risk management team, author

Table 3: Participant in Analysis phase of case study (Basit et al. 2009)

Treatment phase:

After analysis the risk items are then passed on for treatment. The Treat activity is also made value based. See figure 7 and table 5 (Basit, Murtaza & Ikram, 2009) for detail flows:

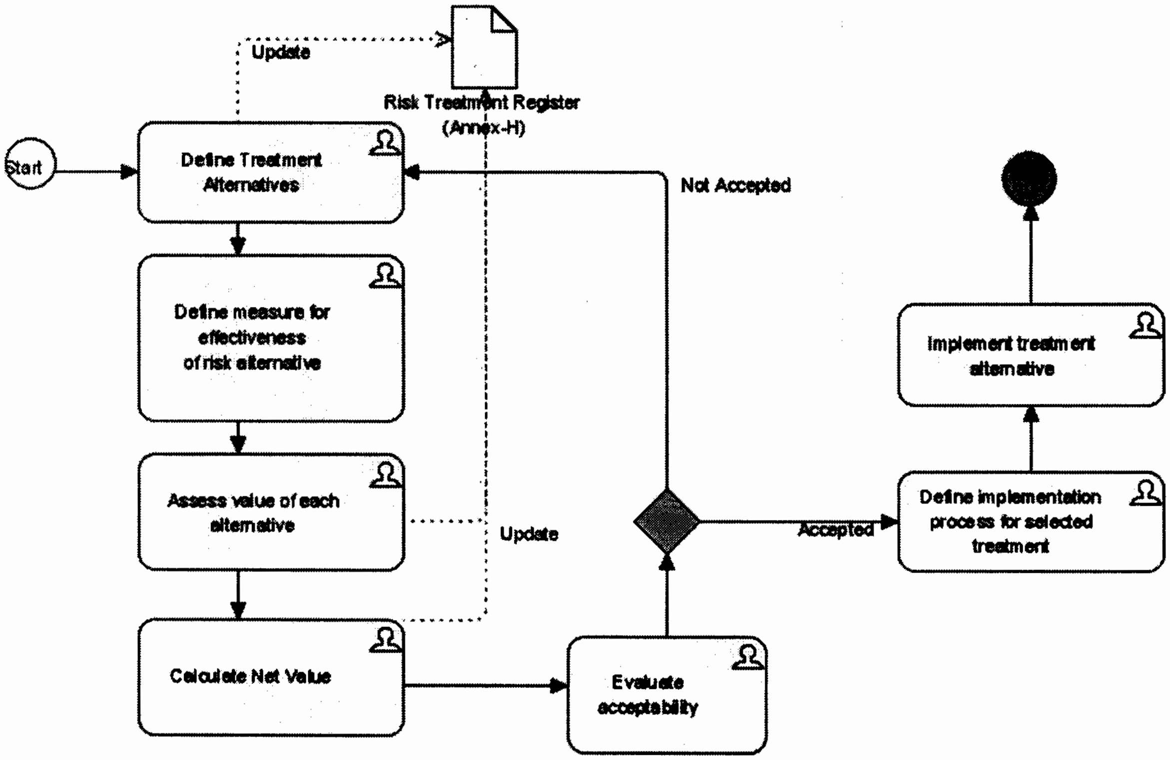


Figure 7: Treatment Diagram of VRRM case study (Basit et al. 2009)

[illegible]			
Define Treatment Alternative	Risk Register, Risk Management Policies	"Risk Treatment Register" (Annexure H)	Risk Management Team , Author , Success Critical Stockholder, Project management Team
Define Measure for effectiveness of Alternative	Risk Register, Risk Alternate Treatments	List of Measures	Risk manager
Access Value of Each Alternative	Alternate Risk Treatments	"Risk Treatment Register" (Annexure H)	Success Critical Stakeholder
Calculate Net Value	Alternate Risk Treatments	"Risk Treatment Register" (Annexure H)	Risk Manager
Evaluate Acceptability	"Risk Treatment Register" (Annexure H)	"Risk Treatment Register" (Annexure H)	Project Manager , Success Critical Stakeholder
Define Implementation process for Selected Treatment	Alternate Risk Treatments	Execution Results and Verifications	Risk Management Team, Project Manager
Implementation Treatment Alternative	Alternate Risk Treatments	Execution Results and Treatments	Project Management Team

Table 4: Participants in treatment phase of case study (Basit et al. 2009)

Monitoring:

Monitor & Control is a continuous activity and is performed parallel with all other activities continuously.

Activity			
Monitoring	Risk Register	Mitigation Plan	Risk Management Team , Project Management Team

Table 5: Participant in monitoring phase of case study (Basit et al. 2009)

The case study actually implemented VRRM and flagged certain concerns. However, implementation of just one case study is not sufficient. So keeping resources in view an idea was devised to simulate VRRM using some simulation tool. A thorough research was implemented in this regard, simulation techniques were studied, tools were analyzed, reference model was developed and finally scenarios were built and run using the selected simulation tool.

Chapter- 5

Simulation Technique & Tool Selection

5 - Simulation Technique & Tool Selection

In this chapter different simulation techniques were studied and appropriate tool selection was conducted in order to simulate VRRM.

5.1 Simulation Technique

A thorough analysis of simulation techniques was conducted in order to select appropriate one. This section gives brief overview of all the possible techniques and the reasons for their being appropriate and inappropriate for VRRM model simulation:

Stochastic: stochastic was not appropriate because in VRRM probabilistic factor is not involved i.e. each and every variable has a concrete value and nothing like "random sampling" is involved.

Deterministic: Deterministic was not qualified because deterministic simulation model deviate the way VRRM works. In VRRM the same set of input variables may not result the same output values e.g. the same set of risks identified may not qualify for treatment for each run, depending on the threshold values and weight age assigned by SCS.

Static: Static disqualified as In VRRM, it is not possible to provide all the input parameters at single point in time e.g. the risk qualification from treatment is evaluated (by SCSs) only when the risk has identified (at any stage in model run)

Continuous: Continuous was not qualified because, In VRRM update of model variable values at equidistance time steps is not required e.g. the criticality of risk can only be entered / evaluated when it occurs i.e. it is impossible to predict about, at what instance of time risk was occur and when was SCS evaluate its weight age and priority for treatment.

Similarly, once the risk has found, it is impossible to predict about when it will qualify for treatment and when will the SCS prioritize/ select the final treatment from list of alternative treatments.

Event-Driven: Event-Driven simulation conforms to all the aspects of VRRM as VRRM is also based on group/network of activities with systematic data flow among them. Similarly VRRM also requires event driven approach due to its "input required at runtime" and even driven nature (occurrence of risk at any instance, qualification of risk for treatment at any instance).

Quantitative: is not qualified because In VRRM although model parameters are specified as real or inter numbers but I cannot ensure, for any organization, the availability of empirical data and experts willing to make the estimates for model parameters

From the above discussion and analyzed it clear that "Event Driven" Simulation technique is good candidate for VRRM simulation.

5.2 Simulation Tools Survey

Various methods and techniques can be used for modeling business processes in order to obtain an understanding of possible scenarios for improvement. Many researchers have suggested the tool which integrates components for system dynamic, hybrid, qualitative and quantitative to measure the effectiveness and performance of their business processes. A large number of process simulation modeling tools are available in market but as per VRRM process requirement we are bound to select a tool which supports "Discrete-Event" simulation. In order to achieve our research objective and to select appropriate simulation tool, we have divided tool selection criteria into four categories:

5.2.1 Technique Support

Simulation Techniques (ST) matrixes are divided into nine sub categories. The main objective of these sub categories is to evaluate the simulation technique support related capabilities of a tool. The table 6 depicts simulation techniques along with their possible values.

Simulation Techniques	Possible Value(s)
System Dynamics	Yes OR No
Continuous-Time	Yes OR No
Hybrid	Yes OR No
Discrete-Event	Yes OR No
Quantitative	Yes OR No
Static	Yes OR No
Qualitative	Yes OR No
Object-Oriented & Hierarchical	Yes OR No
Multi-Method	Yes OR No

Table 6: Simulation technique matrix for tools analysis

5.2.2 Functional Feature

Functional feature support matrixes are categorized into seven sub categories to evaluate functional characteristics of tool which are required during model execution. Table 7 illustrates functional features along with their possible values.

Functional Features	Possible Value(s)
Scenarios Support	Yes OR No
Animation	Low OR Medium OR High
Output Statistic	Low OR Medium OR High
Graphic	Low OR Medium OR High
Parallel Activity Support	Yes OR No
Activity Base Costing	Yes OR No
Database Connectivity	Yes OR No

Table 7: Functional Features matrix for tools analysis

5.2.3 Non-Functional Feature

These matrixes are used to evaluate the non-functional characteristic of tool and divided into five categories in table 8.

Non-Functional Features	Possible Value(s)
Free	Yes OR No
Compatibility with other tools	Yes OR No
Operating System Support	Name(s)
User Friendliness	Low OR Medium OR High

Table 8: Non-Functional features matrix for tools analysis

5.2.4 Experimentation Feature

The experimentation related matrixes are subdivided into eight categories which can be used to evaluate simulation model results. The Table 9 demonstrates the features along with their possible values.

Experimentation Feature	Possible Value(s)
Simulation Experiment	Yes OR No
Optimization Experiment	Yes OR No
Parameter Variation Experiment	Yes OR No
Compare Run Experiment	Yes OR No
Monte Corel Experiment	Yes OR No
Sensitivity Analysis Experiment	Yes OR No
Calibration Experiment	Yes OR No
Custom	Yes OR No

Table 9: Experimentation feature matrix for tools analysis

5.2.5 Tool Selection

A large number of simulation tools and packages are available in market. Initially we listed round about twenty tools but shorted only nine tools for further analysis after basic judgment of the author .In this activity vendor of tools was contacted and a tool evaluation performa was sent to them to evaluate their respective tool according to criteria define in sections 5.2. The table 10 gave overview of capabilities of all tools include in survey. Finally GoldSim was selected as candidate tool for VRRM simulation because it support almost all simulation techniques including "Discrete Event", moreover it support parallel activity run, parameter variation and high support to run different scenarios and free for academic purpose.

Technique Support Tools	System Dynamics	Continuous Time	Hybrid	Discrete Event	Quantitive	Static	Qualitative	Multi-Method	Object-Oriented & Hierarchical
AnyLogic	Y	Y	Y	Y	Y	Y	Y	X	X
GoldSim	Y	Y	Y	Y	Y	Y	X	Y	Y
Process Model	X	X	X	Y	Y	X	X	X	Y
Faber	Y	Y	Y	Y	Y	Y	Y	Y	Y
VenSim	Y	Y	X	X	Y	X	Y	X	X
Simio	Y	Y	Y	Y	Y	X	Y	X	X
Simprocess	X	X	X	Y	Y	Y	Y	X	Y
PowerSim	Y	Y	X	Y	Y	Y	Y	Y	Y
SIMUL 8	Y	Y	Y	Y	Y	Y	X	X	Y

Experiment Features Tools	Simulation Experiment	Optimization Experiment	Parameter Variation Experiment	Compare Run Experiment	Monte Corel Experiment	Sensitivity Analysis Experiment	Calibration Experiment	Custom
AnyLogic	Y	Y	Y	Y	Y	Y	Y	Y
GoldSim	Y	Y	Y	X	Y	Y	Y	X
Process Model	Y	Y	Y	Y	X	Y	X	X
Faber	Y	Y	Y	X	X	X	X	Y
VenSim	Y	Y	Y	Y	Y	Y	Y	Y
Simio	Y	Y	Y	X	Y	X	X	Y
Simprocess	Y	Y	Y	Y	Y	Y	Y	Y
PowerSim	Y	Y	Y	Y	Y	Y	Y	X
SIMUL 8	Y	Y	Y	Y	N	Y	N	Y

Functional Features Tools	Scenarios Support	Animation	Output Statistic	Graphs	Parallel-Activity Support	Activity Base Costing	Database Connectivity
AnyLogic	Y	H	H	H	Y	Y	X
GoldSim	Y	L	H	M	Y	X	Y
Process Model	Y	H	H	H	N	Y	X
Faber	Y	M	M	H	Y	X	X
VenSim	Y	L	H	H	N	X	Y
Simio	Y	H	H	M	Y	X	Y
Simprocess	Y	M	H	H	Y	Y	Y
PowerSim	Y	H	H	H	H	N	Y
SIMUL 8	Y	Y	H	H	Y	Y	Y

Non-Functional Features Tools	User Friendly Environmen	Documentation Availability	Free	Operating System Support	Compatibility with Other Tool
AnyLogic	H	Y	N	Window , linux	M
Goldsim	H	Y	Y (ACADEMIC)	Window	M
Process Model	H	Y	N	Windows 7	H
Faber	M	Y	N	Windows, Linux,	L
VenSim	H	Y	N	Window	M
Simio	H	Y	N	Window	Y
Simprocess	H	Y	N	Windows, Linux	H
PowerSim	H	Y	N	Windows 7, Vista, XP or 2000	M
SIMUL 8	H	Y	N	Window	H

Table 10: Detail analysis of simulation tools

Chapter- 6

Reference Model Implementation

6 - Reference Model Implementation

In this chapter a reference model (RM) was designed to bridge gap between the actual VRRM and GoldSim, Moreover model validation was also conducted in parallel to the simulation model development.

6.1 Reference Model

All process modeling tools has modeling elements / constructs to develop a simulation model of any process. Every modeling tool is composition of different modeling elements and every element need some pre-requisite to reflect the actual behavior of a process. So we need to design a reference model to reduce the gap between conceptual model and simulation tools. A reference model behaves like a bridge between the actual VRRM and GoldSim. Figure 8 show graphical representation of model and flow of information.

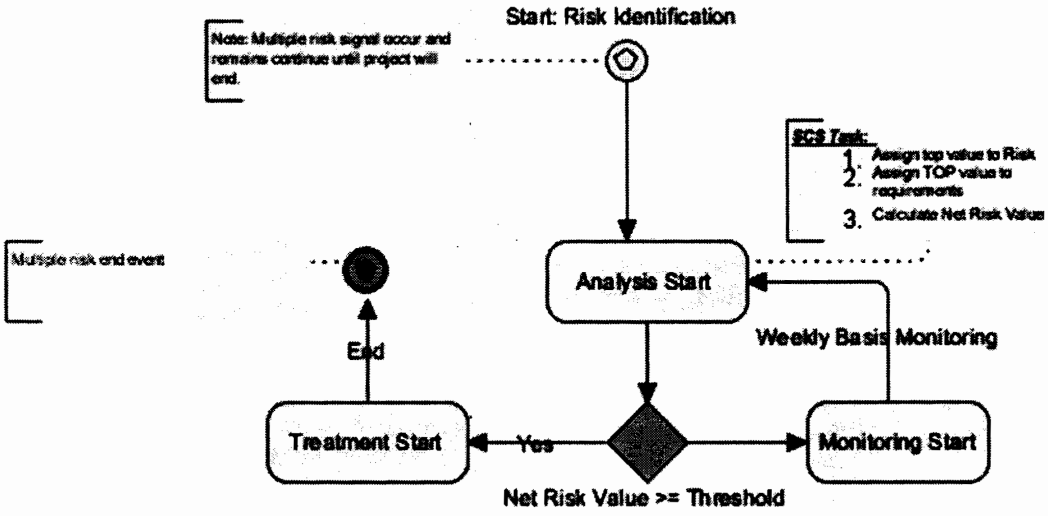


Figure 8: VRRM reference simulation model

In our reference model (RM), after the risk identification success critical stakeholder assigned TOP values to risk and all those requirements having any direct/indirect relation with the risk using equation presented in section 6.1.1. Once TOP values assigned, risk net value was calculated using the equation presented in section 6.1.1. Moreover risk was considered for treatment if and only if net risk value was greater than threshold otherwise it was put in monitoring list. All the risks was monitor regularly on weekly basis and was consider for treatment when their net risk value was cross the threshold value. Treatment is also value based, SCS assigned TOP value to every treatment alternative. Once TOP values are assigned to every alternative then equations given in section 6.1.1 was used to select suitable alternative for treatment.

6.1.1 RM Equations

All the equations presented in this section were derived from results obtained in (Basit, Murtaza & Ikram, 2009) case study. However the equations were also slightly modified and implemented in simulation model as a result of endeavor to overcome the shortcomings of original VRRM model on which (Samad, Ikram, & Muhammad, 2008) was based.

<u>Equations to calculate Risk Value:</u>

1. Each SCS assigns value to each risk from Technical, Organizational and People point of view. The average of all the three values is calculated for each risk using following equation where n represent no. of SCS, w1 represent weight for technical value, w2 represents weight for organization value and w3 represents weights for people values.

$$J = \text{TopValue/Risk} = \sum_{i=1}^{n} SCS_i(W1 \times T + W2 \times O + W3 \times P)/3$$

2. After each SCS assigns TOP value to each risk, the Net Risk Value is calculated by taking average of all the TOP values assigned by n Stack holders using the blew equation.

$$K = \text{Risk value} = \frac{J}{n} \qquad n = \text{total number of SCSs}$$

3. Similarly, Each SCS assigns value to each requirement from Technical, Organizational and People point of view. The average of all the three values is calculated for each requirement using blew equation where n represent no. of SCS, w1 represent weight for technical value.

$$L = \text{TopValue/Req} = \sum_{i=1}^{n} SCS_i(W1 \times T + W2 \times O + W3 \times P)/3$$

4. After each SCS assigns TOP value to each requirement, the Net Req Value is calculated by taking average of all the TOP values assigned by n Stack holders to each requirement.

$$M = \text{Req value} = \frac{L}{n} \qquad n = \text{total number of SCSs}$$

5. For each risk the n requirements are linked with, so here the average of all the requirement values is calculated to get a net requirement value for that risk where Mi represent the average value of a single requirement.

$$N = \text{NetReqsValue/Risk} = \sum_{i=1}^{n} M_i \,/\, n \qquad n = \text{total no. of reqs linked per risk}$$

6. The net requirement risk value is calculated by taking the average of net risk value and net requirement value

$$O = \text{NetReqRiskValue} = (K + N)/2$$

Equations to evaluate treatment alternative:

1. Each SCS assigns value to each alternative (expected for a particular risk) from Technical, Organizational and People point of view. The average of all the three values is calculated for each alternative where T, O and P represent technical, Organizational and People values respectively.

$$P = \text{TopValue/Alternative} = \sum_{i=1}^{n} SCS_i (T + O + P)/3 \qquad n = \text{total number of SCSs}$$

2. For each alternative n SCSs assign the TOP value, so here we calculate the Net Alternative Value by dividing the TOP value by total number of SCSs (participating in value assignment)

$$Q = \text{Net Alternate value} = \frac{P}{n} \qquad n = \text{total number of SCSs}$$

3. Finally a particular alternative is selected on the basis of the maximum net alternate value i.e. the alternate having maximum Net Alternative Value is selected for treatment of risk under consideration.

$$R = \text{Selected Alternative} = \text{Max}\,(Q_1 + Q_2 + Q_3 + \cdots + Q_n)$$

6.2 GoldSim Elements Used in RM

Reference model was developed in GoldSim using version 10.11. In RSM seven type of GoldSim elements were used. Please the next subsequent sections for detail.

6.2.1 Container Element

GoldSim allows user to creating hierarchical "Top-down" model. It provides six types of container elements. But in RSM case, only two of the containers were used.

Conditionality Container: GoldSim provide conditionality containers to allow users to activate or deactivate a specific model portion dynamically. User has the choice to make local or global conditionality containers. Generally, a local container can only be referenced by other elements inside the container. In RSM local conditionality container was used to divide RSM into three subsystems.

a. Identification

b. Analysis

c. Treatment

Global Container: The contents of a global Container can be (referenced) by any other elements in the model. In RSM I used global container to assign TOP values to "Risk", "Requirement" and "Alternative"

6.2.2 Event Elements

In order to start or stop any activity in RSM to reflect the actual behavior of VRRM process, signals need to be generated. GoldSim allows its certain elements to give out or receive information in form of signal. Discrete signals are special type of output which emits information discretely rather than continuously. In RSM four types of event elements were used all subsystems (Identification, Analysis and Treatment). Details of these elements are given below:

Timed Event: RSM was initiated when any risk item was found. In order to start a process, a signal needs to be generated for other elements of the process to inform them about the new risk item being found and consequently, enabling other activities to start with further processing. GoldSim provides Timed Event to replicate this behavior. Timed Event elements produce discrete event signals based on a specific rate of occurrence.

Discrete Change Event: In RSM every risk item should be tracked and appropriate actions should be taken if any risk was found (e.g. whenever a risk was found, a unique ID was assigned to it and SCS was intimated to assign TOP value to risk, requirements, alternates etc).

To achieve the required behavior, GoldSim provides a very powerful element 'Discrete Change'. Whenever a discrete element was triggered by an event, it emits a discrete change signal in response. This signal contains information about the response to an event and consists of two pieces of information, a value (e.g. Rs. 10) and an instruction (e.g. Add). The discrete change element was used widely for keeping track of every change in RSM.

Decision and Status Element: In RSM whenever some activity was completed, a decision needs to be taken regarding further processing. To achieve this task, GoldSim provides different types of elements (Decision or status). We have used both these elements in our RSM. Decision elements allows for better decision representation or branching logic in a model. Decision elements have two to three outputs which represent discrete event signals. When an element was triggered, it emits a discrete event signal from only one of its outputs, which depends on a set of user-specified conditions or expressions. Status elements allow the user to observe and monitor the status of anything triggered by the events. The output of a status element was a Boolean output (i.e. true or false). The element also has an initial condition (true or false).

6.2.3 Input Elements

Input elements are used to input data to different activities in GoldSim. There was three types of input elements (Data, Time series and Stochastic) in GoldSim. In RSM two of inputs element (Data and Stochastic) was used.

Data: Data elements are intended to represent the constant, variable inputs in RSM. A Data element can represent both values and conditions (i.e., True/False), and can represent a single scalar datum, or an array of data. Data element was excessively used in RSM to input purpose.

Stochastic: GoldSim provide stochastic element to represent uncertainty in the input data. This element was used in Identification phase because there was probability whether risk was found or not, moreover stochastic element support large number of distribution type but in RSM "Long Normal Distribution" for simplicity.

6.2.4 Output Elements

Time history: Time History results show the "history" of a particular output as a function of time. It was used in RSM to show output.

Result Distribution: It is often useful to plot the distributions of multiple outputs on a single plot. It was used in RSM to capture the output in multiple realizations.

6.2.5 Stock Elements & Functions

Reservoir: Reservoir elements are provided by GoldSim to store data which was produced during simulation. In RSM it was used as registers for storing result. (i.e. Risk Register, Risk treatment, Register).

External DLL: At certain occasions, there may be a need for defining a complex function which can not be implemented as it is using the expression editing features which was provided by GoldSim. To cater these problems, GoldSim was designed in such a way as to develop separate programming modules (written in C, C++, Pascal, FORTRAN & other compatible programming languages) which can then be directly linked with the GoldSim algorithms. These user defined modules was referred as external functions and the elements through which it was linked in GoldSim are called External (DLL) elements.

A DLL was developed in C++ which generates TOP (Technical, Organization & People) values which was later used by success critical stockholders for assigning TOP value to risk item, requirement and treatment alternatives. Please see the code of DLL in appendix A.

Selector: GoldSim introduced selector element to deal with such situations where the user want to use IF-THEN logic in the model to specify how a certain parameter changes as a function of some state of the system (e.g. time). Selector element was used in treatment subsystem to select the treatment alternatives.

6.3 Abstract level 1:

RSM has two levels of details just like that of conceptual Value Based Requirement Risk management (VRRM) (Samad,Ikram, & Muhammad,2008). Simulation model consist of three subsystems and a result container. Every subsystem in simulation model has some pre-conditions for becoming active or inactive with the exception of results and control variables. Result container contains the results of simulation and control variable contain the control variable to run simulation according to given scenario.

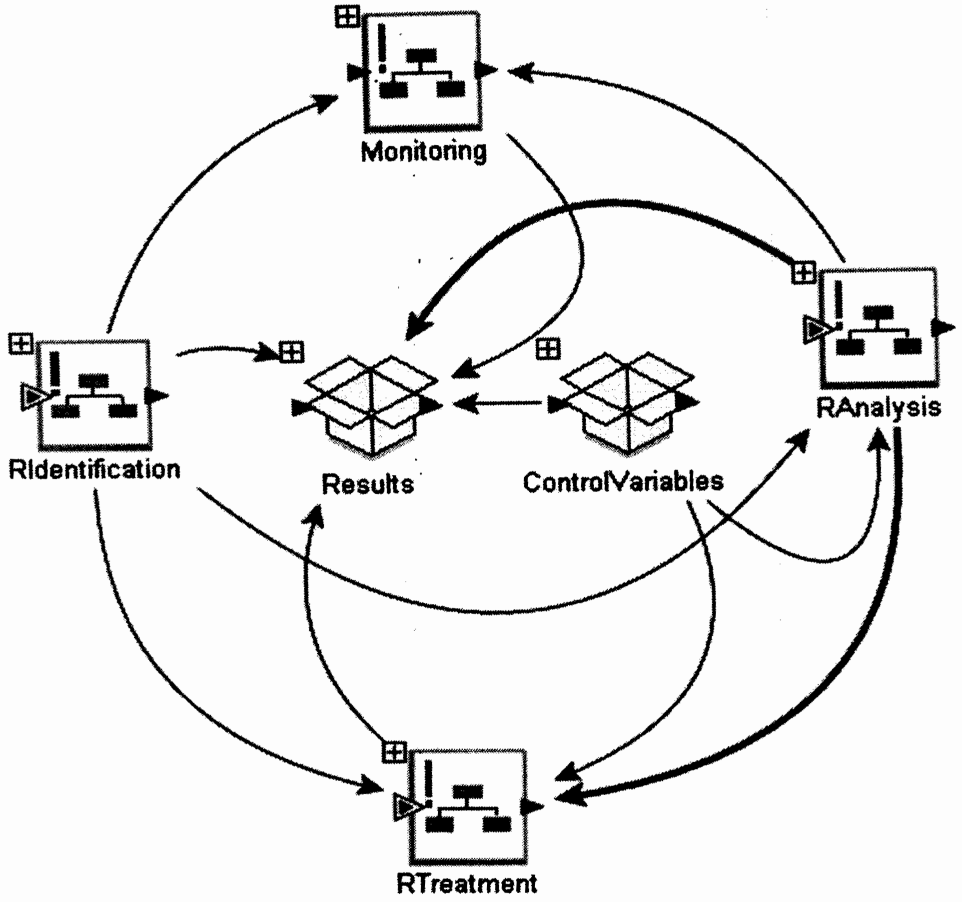


Figure 9: High level implementation of VRRM simulation model

The main objective of RIdentification subsystem was to find risk items. It automatically becomes active when the project starts and become deactivated on the completion of the project. The working of RAnalysis and RTreatment subsystems was different than RIdentification. On detection of any risk item, RIdentification subsystem generates a signal to RAnalysis and RTreatment subsystem concurrently.

RAnalysis subsystem became active on getting a signal from RIdentification however RTreatment waits for the analysis of risk. When RAnalysis was done with the risk item, RAnalysis generates a signal to RTreatment subsystem if there was a need for treatment alternative and goes to inactive state. Upon receiving the signal, RTreatment got activated and executed treatment alternative, moreover once the selected treatment alternative executed it also become inactive.

6.4 Abstract Level 2:

mentation of every subsystem and result container

6.4.1 RIdentification

In RIdentification subsystem six types of GoldSim elements was used to find risk item. In this subsystem a timed event was used to generate a signal to Likelihood element to search for risk item. Likelihood was stochastic data element of GoldSim element used to analyze the probability of risk to occur. Long-normal distribution was used in the 'Probability Stochastic' data elements to calculate the probability of risk. Likelihood hunts for risk items on a daily basis.

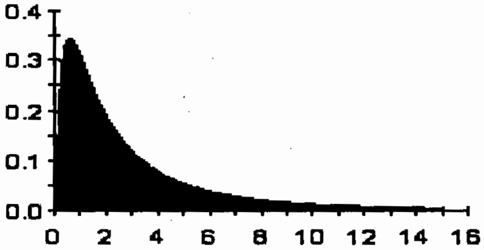


Figure 10: Long-normal distribution of risk occurrence

RiskSignal is a decision element used for decision making regarding sending of risk item for analysis. Identification subsystem was presented in figure 11.

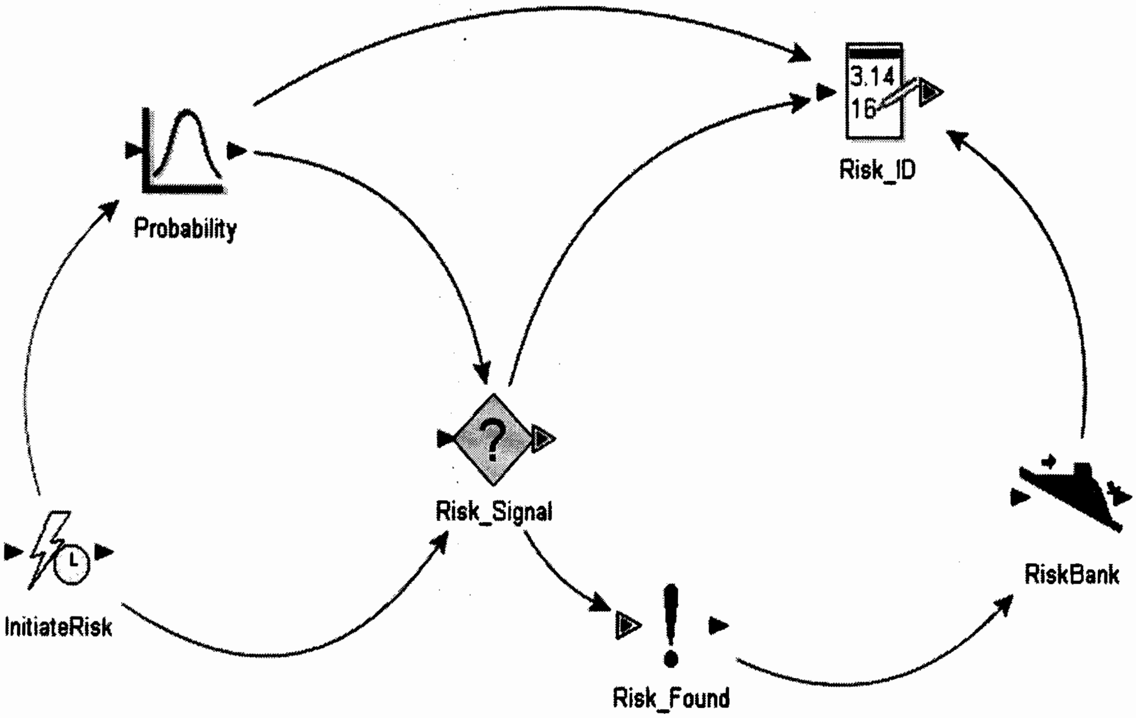


Figure 11: Risk identification subsystem diagram

RiskFound was a discrete change element for keeping track of changes which occur in the system. RiskFound updates RiskBank upon receiving a green signal from RiskSignal element. RiskBank was a data storage element and works in the same way as risk registers in VRRM Case study (Basit, Murtaza & Ikram, 2009).

6.4.2 RAnalysis: -

Like RIdentification, the RAnalysis phase also behaved like subsystem. In RAnalysis subsystem, there were seven element of GoldSim used to analyze a risk item. It got activated when it received any risk signal from RIdentification subsystem and got deactivated when analysis of the risk was completed.

As the main objective of RAnalysis subsystem was to analysis risk item, to achieve the required behavior. Its working includes:

1. Success Critical Stockholder (SCS) assigns TOP values to risk item.
2. Only the relevant SCS assigns TOP value to the Risk item as per requirement of VRRM. Once the TOP values are assigned to risk, System calculates their aggregate value.
3. In parallel to the above mention activities, requirements are dynamically linked to risk items by the simulation model and relevant SCS was assigned TOP Value to the requirements.
4. Aggregate of requirements and risk was be compared with the threshold to decide whether the risk item needs any treatment or not. If the aggregate is greater than threshold then risk was be sent to RTreatment subsystem otherwise it was monitor regularly. See figure 12 for detail.

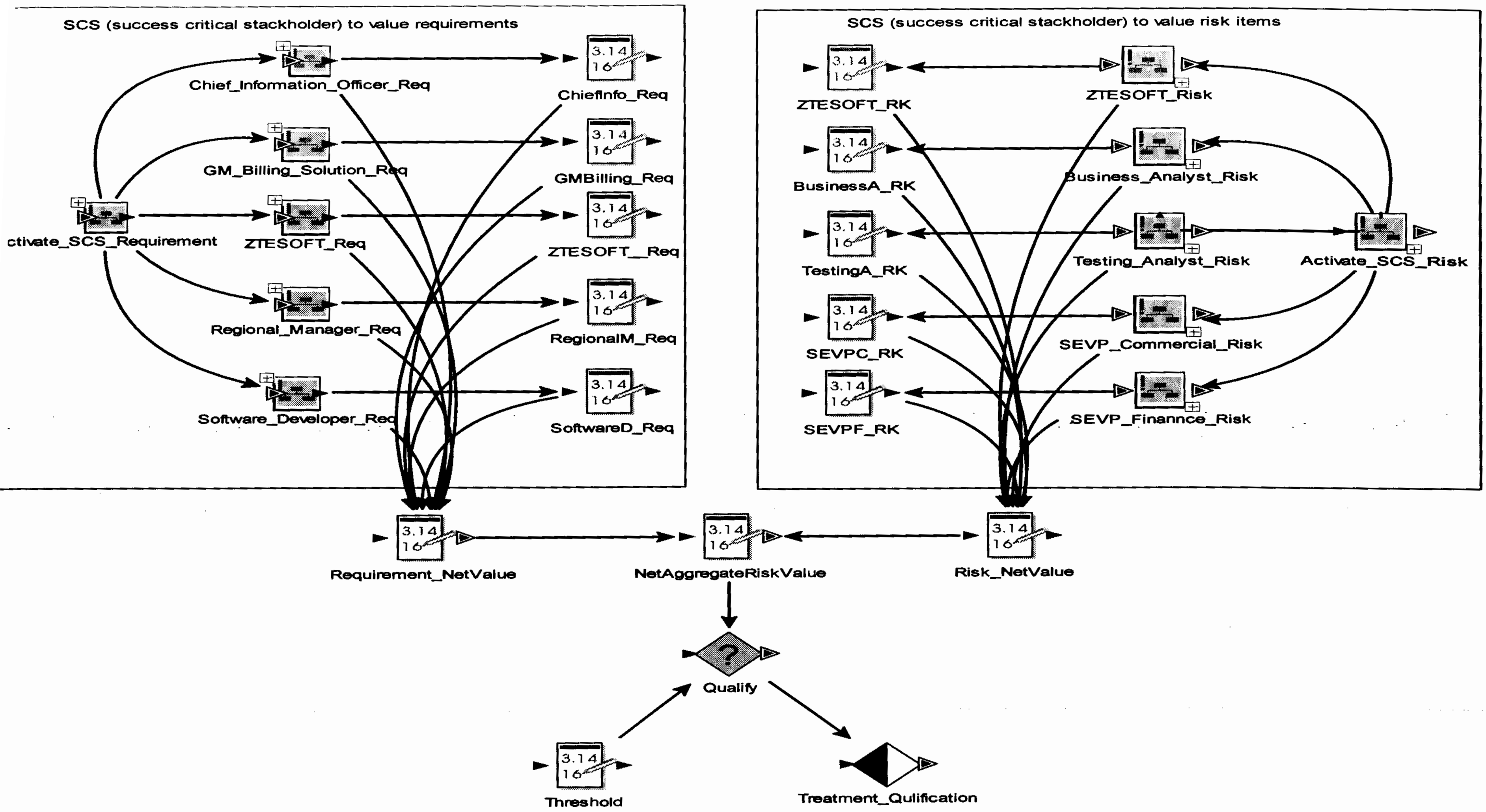


Figure 12: Risk Analysis subsystem diagram

Risk TOP Values: - GoldSim provided container element to implement layer architecture and it was used in RSM excessively. In figure 13 Success Critical Stakeholders (ZTESOFT) assigned TOP value to risk item. In RSM GoldSim data element used to calculate the risk TOP and aggregate values.

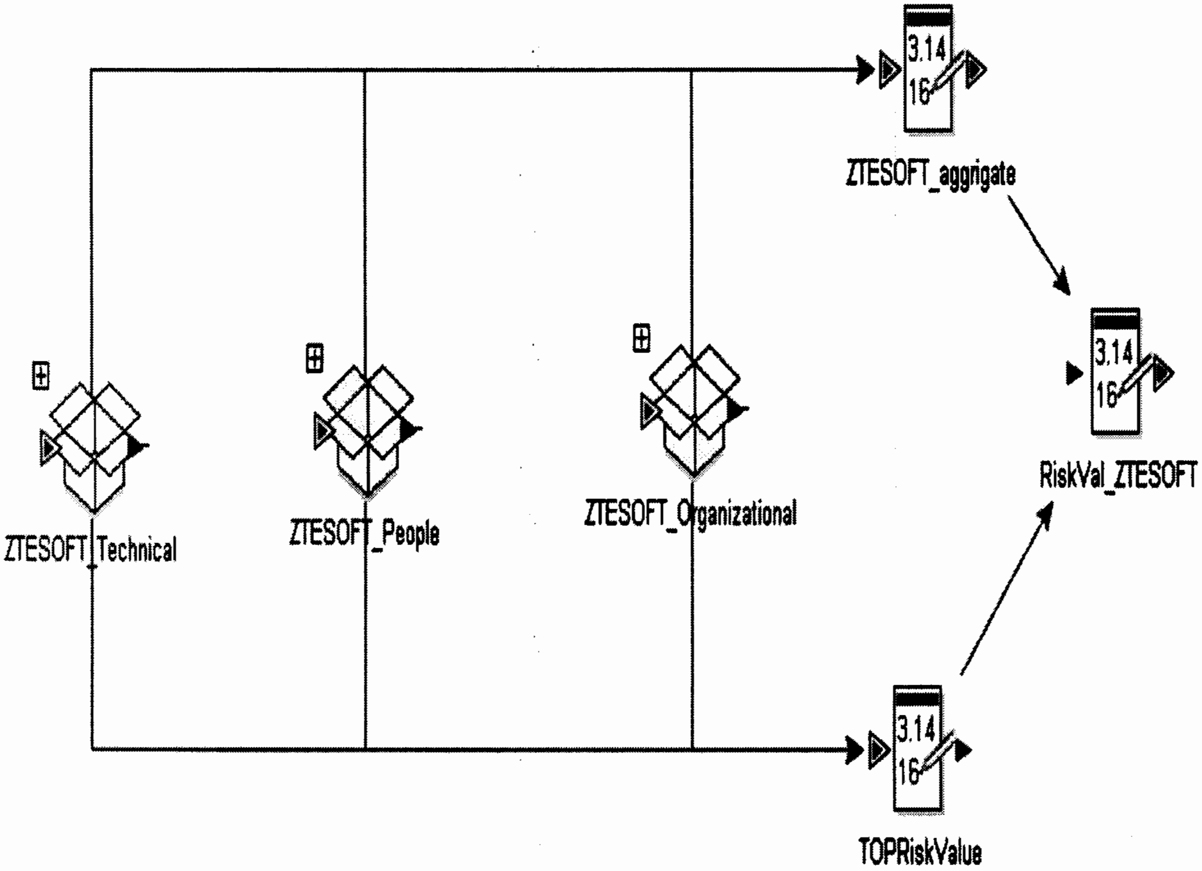


Figure 13: TOP value assignment to risk item

Requirement TOP Values:

In figure 14 Success Critical Stakeholders (Chief Information Officer) assigned TOP value to requirement which was linked to a risk item. The GoldSim data element was used to calculate the risk TOP and aggregate values of requirement.

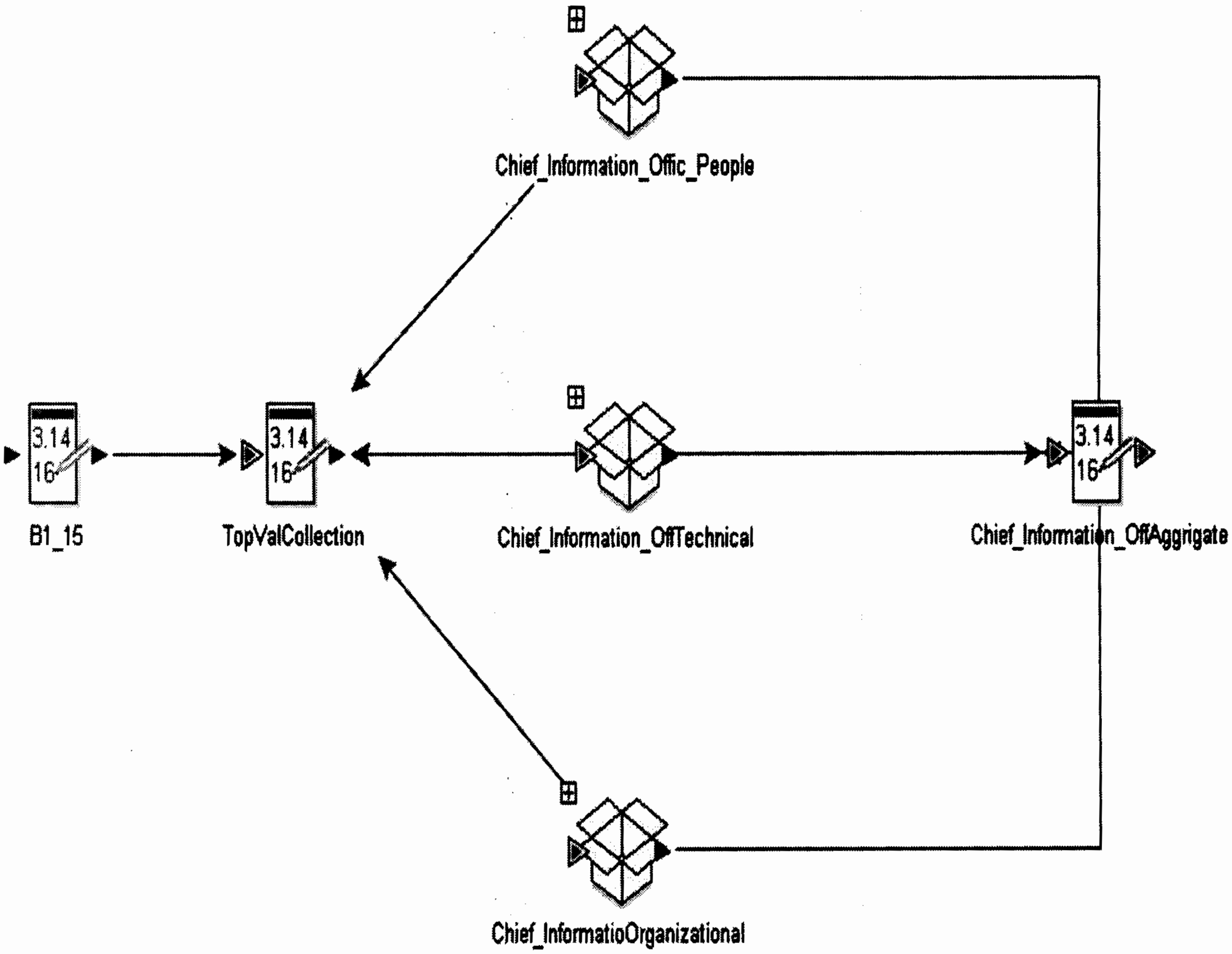


Figure 14: TOP value assignment to linked requirements

6.4.3 RTreatment:

The main objective of treatment subsystem was to evaluate and apply the best alternative to a risk item. In RTreatment subsystem seven different elements of GoldSim was used to replicate the desire behavior. Treatment subsystem working includes:

1. Treatment subsystem was dependent on both identification and Analysis subsystems. It got activated only when it received any signal from both identification and Analysis subsystem concurrently.
2. Once it was activated, then SCS (Success Critical Stakeholder) assigns TOP value to those treatment alternatives which are appropriate to treat a risk item.

After the TOP value assignment activity was completed, the aggregate of all alternative was compared with each other. The alternative having highest aggregate value was selected for the treatment of risk item. Figure 15 show the implementation of RTreatment subsystem in RSM.

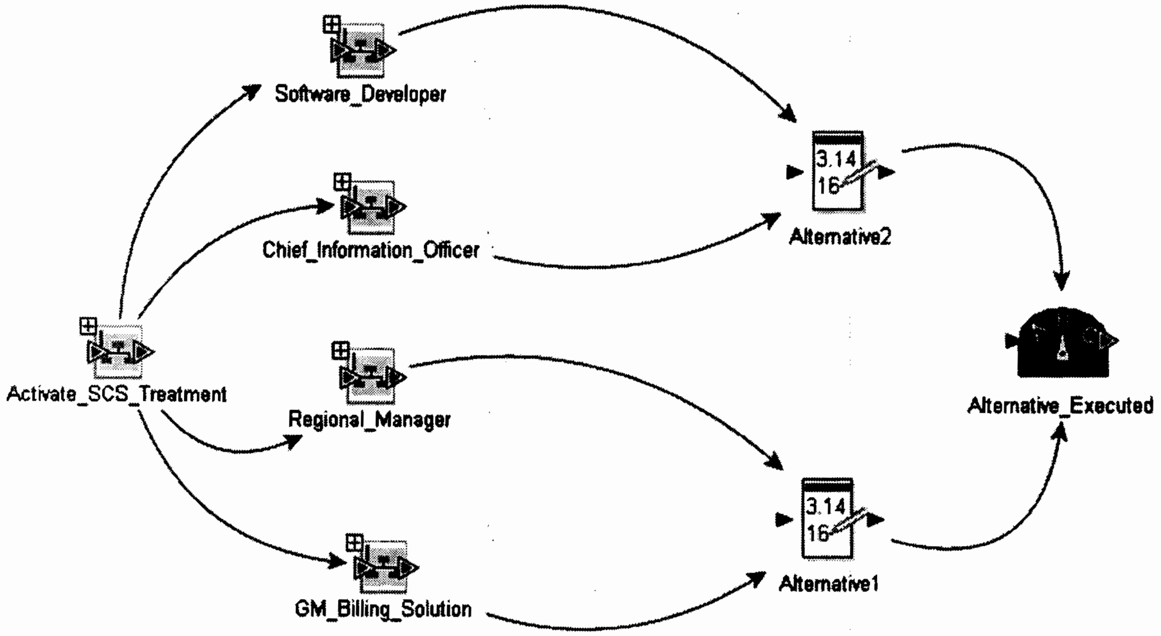


Figure 15: Treatment subsystem diagram

Alternative Top Value: -

In figure 16 a Success Critical Stakeholders was shown whileassigning TOP value to alternative ALT_G9. The GoldSim data element was used to calculate the TOP and aggregate values of alternative.

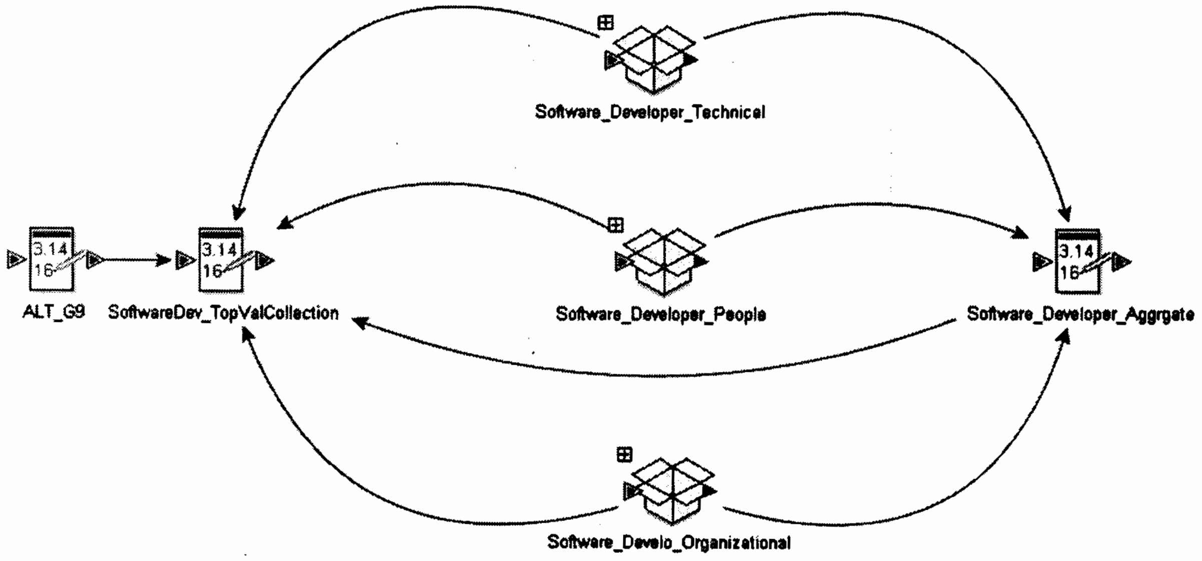


Figure 16: Alternative evaluation diagram

6.4.4 RMonitoring:

Risk monitoring was vital activity in RSM. This activity was performs separately on weekly basis using extension model of RSM.

RSM extension model working was similar as based RSM simulation model except Identification activity. RIdentification subsystem of extension RSM was shown in figure 17.

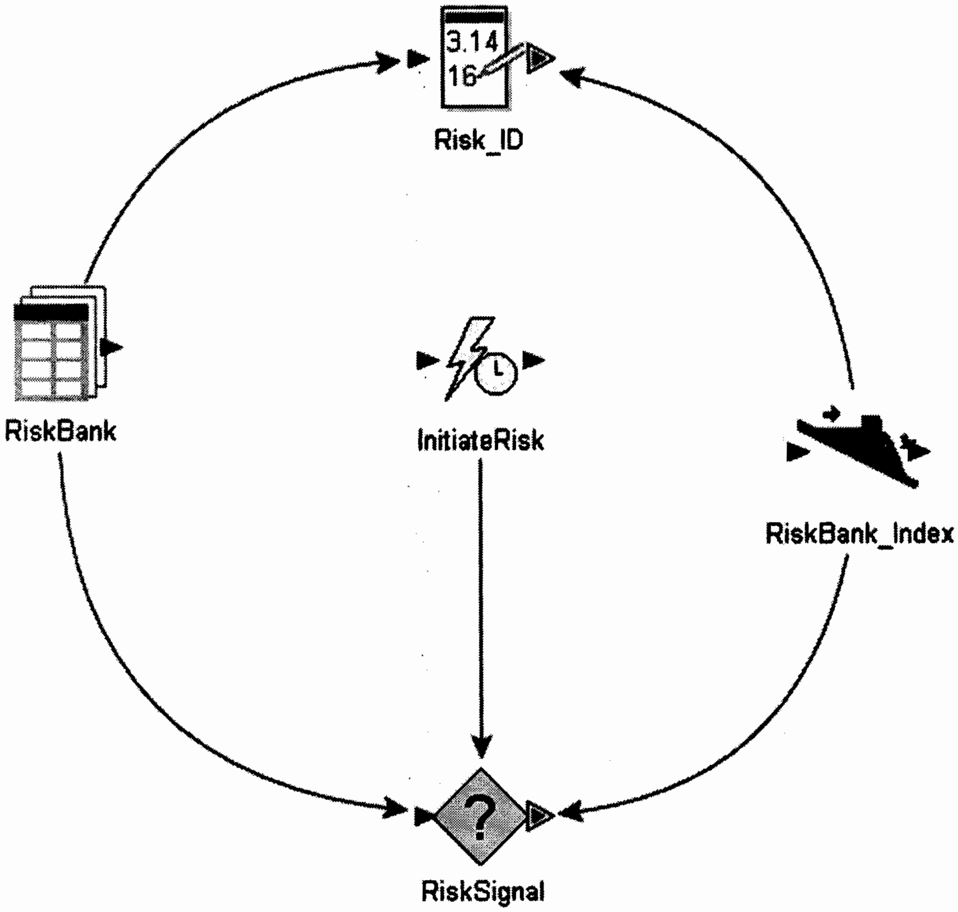


Figure 17 : Identification in monitoring model

In Identification activity all the potential risks were imported in RiskBank on weekly basis and then extension simulation model simulate all the imported risks to find their criticality similarly as in the Base VRRM simulation model.

TOP Value Assignment Mechanism:

In RSM random number was used to assign technical, people and organization value to risk, requirement and treatment alternative excessively. To achieve it, GoldSim provide an extra feature to link any external DLL file to imulation model. An external DLL was linked to with RSM to generate random number See figure 18 for detail.

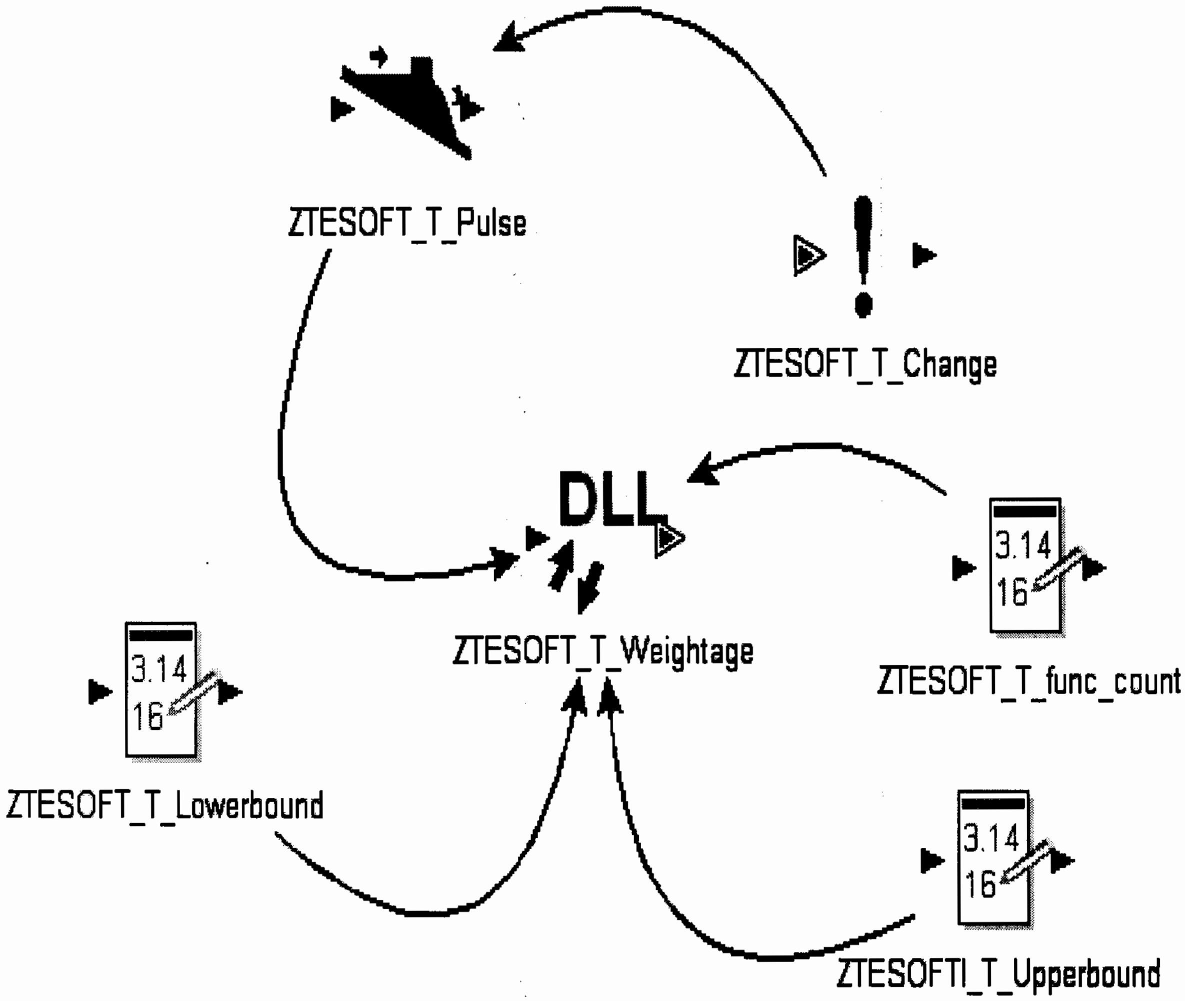
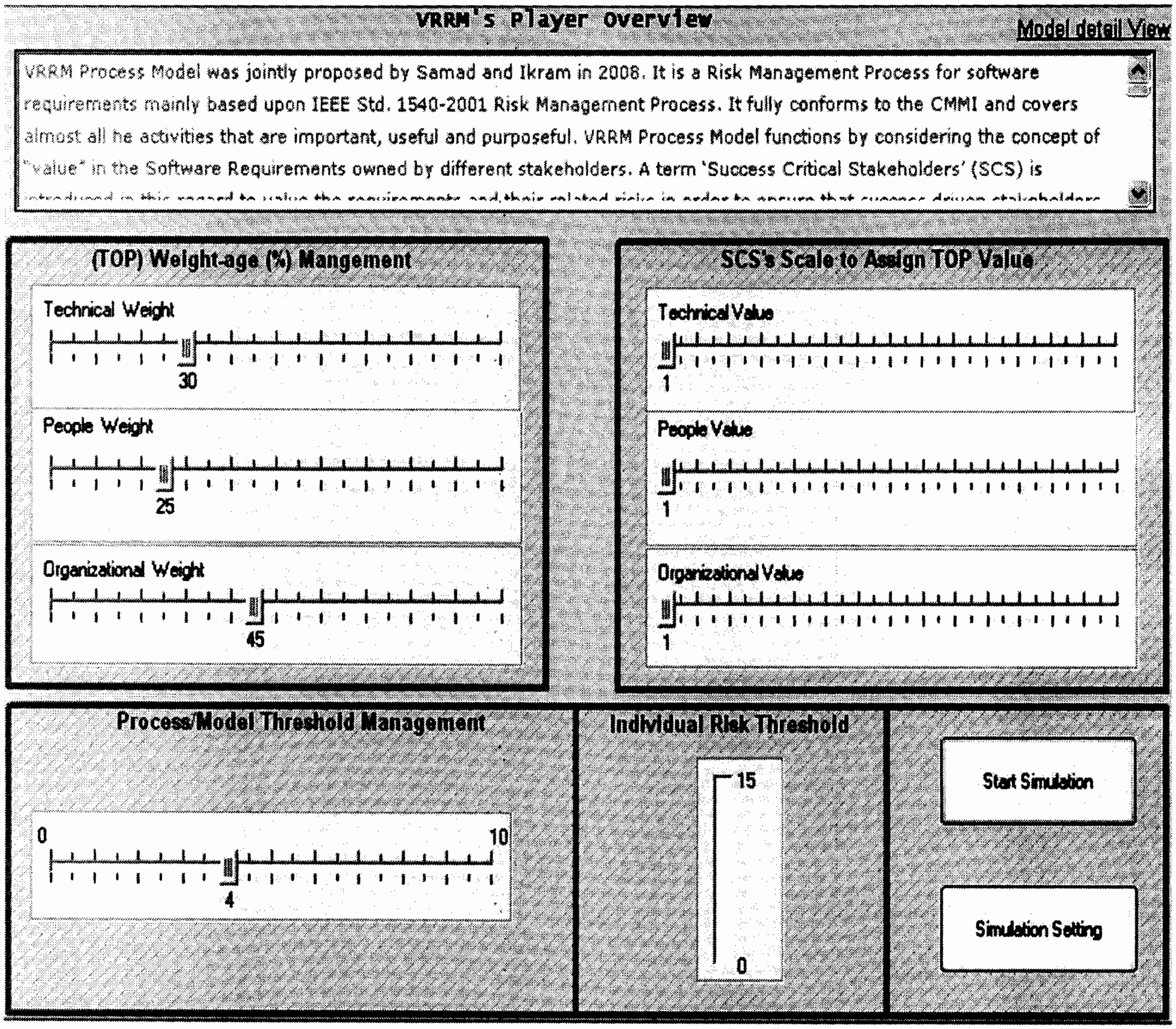


Figure 18; TOP values assignment mechanism

So, external DDL was used to generate random number between the given upper and lower bound. GoldSim called external DDL if and only if the values of its input parameters change. But in RSM, we need that whenever SCS need to assign any of the TOP values. In such cases, GoldSim calls external DDL every time to generate random value. To reproduce this required behavior, Two GoldSim elements were used, discrete change event and reservoir. These two elements change one of the input parameters of DDL whenever the SCS needs to assign the TOP value to risk, requirement or treatment alternative. The code of DLL is given in appendix A

6.4.5 RSM Dashboard

A fully equipped dashboard was developed to run different scenario on RSM. Figure 19 show the graphical representation of dashboard model.



VRRM Process Model was jointly proposed by Samad and Ikram in 2008. It is a Risk Management Process for software requirements mainly based upon IEEE Std. 1540-2001 Risk Management Process. It fully conforms to the CMMI and covers almost all he activities that are important, useful and purposeful. VRRM Process Model functions by considering the concept of "value" in the Software Requirements owned by different stakeholders. A term 'Success Critical Stakeholders' (SCS) is

Technical Weight 30

People Weight 25

Organizational Weight 45

Technical Value 1

People Value 1

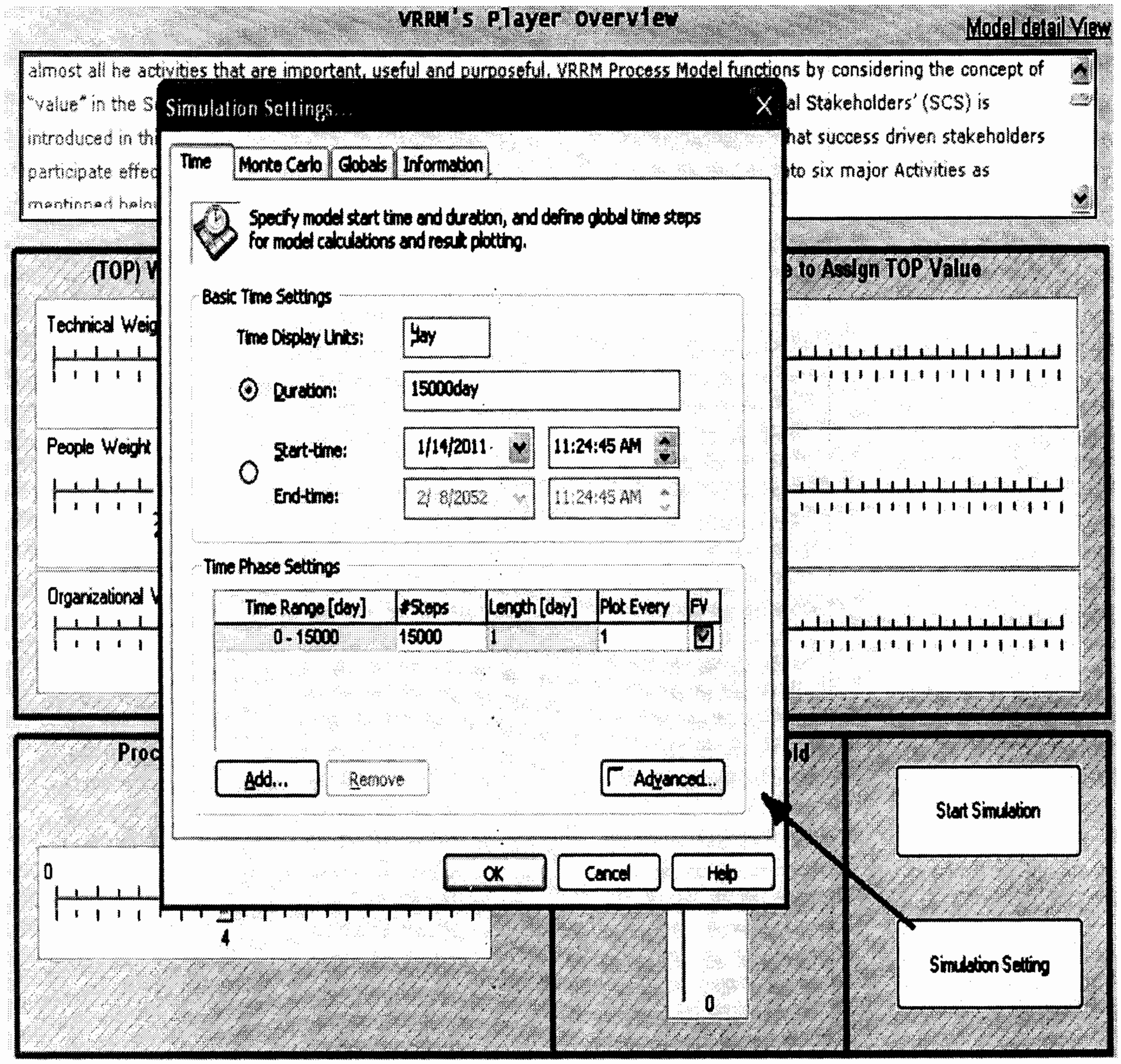
Organizational Value 1

0 10

4

Figure 19: Reference Simulation Model Dashboard

VRRM dashboard is quiet useful for process owner, project manager or any stockholder. Using VRRM dashboard you can change the controls variables (Top values weight, top values scales and process/model threshold) at any time. "Simulation setting" button enable the player to change simulation setting as shown in figure 20.



almost all he activities that are important, useful and purposeful. VRRM Process Model functions by considering the concept of "value" in the S… al Stakeholders' (SCS) is introduced in th… hat success driven stakeholders participate effec… to six major Activities as mentioned belo…

Specify model start time and duration, and define global time steps for model calculations and result plotting.

Basic Time Settings

Time Display Units: day

● Duration: 15000day

○ Start-time: 1/14/2011 11:24:45 AM

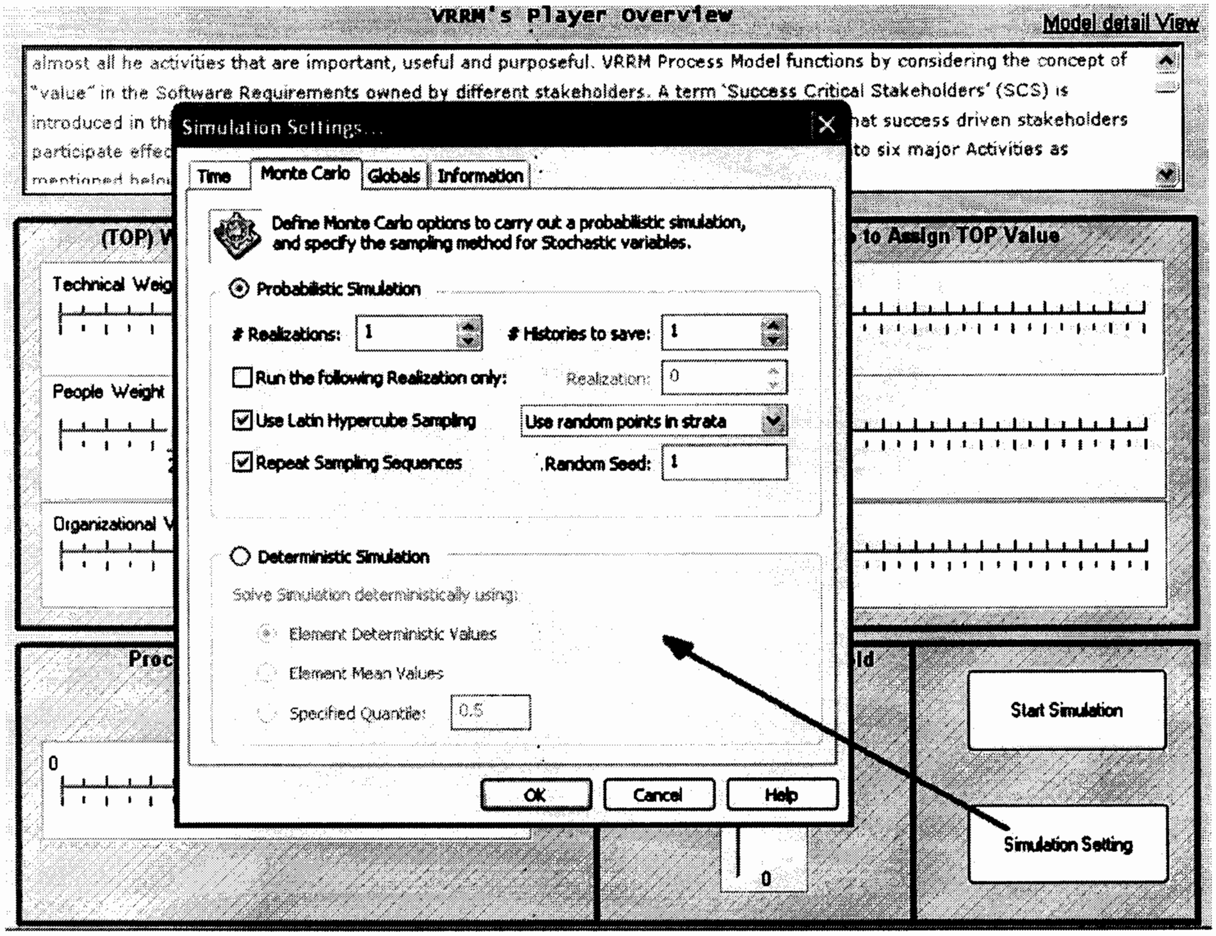
End-time: 2/ 8/2052 11:24:45 AM

Time Phase Settings

Time Range [day]	#Steps	Length [day]	Plot Every	FV
0 - 15000	15000	1	1	☑

Figure 20: Reference Simulation Model Duration Setting

Player can set the project duration in days or starting and ending date of project. It facilitates player to run either monte calro probabilistic or monte calro deterministic simulation. See figure 21 for different option of simulation.



Define Monte Carlo options to carry out a probabilistic simulation, and specify the sampling method for Stochastic variables.

● Probabilistic Simulation

\# Realizations: 1 # Histories to save: 1

☐ Run the following Realization only: Realization: 0

☑ Use Latin Hypercube Sampling Use random points in strata

☑ Repeat Sampling Sequences Random Seed: 1

○ Deterministic Simulation

Solve Simulation deterministically using:

● Element Deterministic Values

○ Element Mean Values

○ Specified Quantile: 0.5

Figure 21: Reference Simulation Model Monte Carlo Setting

6.5 Model Validation

Face validation and parameter validity were user by (David., Marc. 2000, David. 2005, Pfahl., Lebsanft. 2000) to validate simulation model. So we used both techniques to validate VRRM simulation model. Parameter validity technique was already complied with VRRM as VRRM simulation model was run number of times by changing the value of input parameters e.g. threshold value for risks and the internal parameters e.g. TOP values assigned by success critical stakeholders. The behavior of simulation model was studied and results were analyzed for conclusions presented in next section.

Chapter- 7

Scenarios & Model Improvement Areas

7 – Scenarios & Results

In this chapter scenarios was developed and run to collect the data for further analysis, moreover the some improvement areas of VRRM are highlighted.

7.1 Scenarios formulation

In simulation case study two different scenarios was created to know and observe the process behavior. Data was collected from simulation model while running scenarios (one and two), see section 7.1.1 and 7.1.2 for scenario detail. Every scenario had different simulation setting and analyzed for following three cases:

- Risk Analysis
- Risk treatment
- Risk Monitoring

7.1.1 Scenario One

In this scenario, author intension was to observed VRRM RSM model behavior on small scale project from six month to one year project. In this scenario success critical stockholder was able to assigned TOP value on the scale of 1 to 15.

RSM use random value between a given scale to assign top value to risk and requirement. In order to reduce this randomness, weight of 45%, 20% and 35% was attached to Technical, Organizational and People respectively. See the detail of scenario one in table 13.

<table>
<tr><th colspan="6">Case 1st : Risk Analysis</th></tr>
<tr><th colspan="3">TOP Weight-age % Setting</th><th colspan="3">Value Scale Setting for SCS</th></tr>
<tr><th>Technical</th><th>Organizational</th><th>People</th><th>Technical</th><th>Organizational</th><th>People</th></tr>
<tr><td>45%</td><td>20%</td><td>35%</td><td>15</td><td>15</td><td>15</td></tr>
<tr><th colspan="2">Process/Model Threshold</th><th colspan="2">Project Duration</th><th>Simulation Type</th><th>No of Realization</th></tr>
<tr><td colspan="2">4</td><td colspan="2">365 days</td><td>Probabilistic</td><td>1</td></tr>
<tr><th colspan="2">Pre – Condition</th><td colspan="4">- Risk has already identified
- Concerned SCSs are identified</td></tr>
<tr><th colspan="2">Post – Condition</th><td colspan="4">- N number of risks was qualify for treatment
- X – N risk was entered in monitoring phase where X is total risk identified.</td></tr>
<tr><th colspan="6">Case 2nd : Risk Treatment</th></tr>
</table>

TOP Weight-age % Setting			Value Scale Setting for SCS		
Technical	**Organizational**	**People**	**Technical**	**Organizational**	**People**
45%	20%	35%	15	15	15
Process/Model Threshold		**Project Duration**		**Simulation Type**	**No of Realization**
4		- 356 days		- Probabilistic	- 1
Pre – Condition		- Risk identified and has crossed the process threshold - Treatment alternatives have been identified and rated/valued by concerned SCSs			
Post – Condition		- Risk was treated with highest valued treatment alternatives			
Case 3rd: Risk Monitoring					
TOP Weight-age % Setting			**Value Scale Setting for SCS**		
Technical	**Organizational**	**People**	**Technical**	**Organizational**	**People**
45%	20%	35%	15	15	15
Process/Model Threshold		**Project Duration**		**Simulation Type**	**No of Realization**
4		- 365 days		- Probabilistic	- 1
Pre – Condition		- Risk identified and have not cross the threshold			
Post – Condition		- Risk was be monitor regularly on the basis of TOP values assigned by concerned SCSs, until it does not cross the threshold to qualify for treatment			

Table 11: Scenario one

In general a scenario has some pre / post condition, with that in mind some identified pre and post condition of every sub case of a scenario.

7.1.2 Scenario Two

Like "Scenario one" Scenario two was designed to observe VRRM RSM model behavior on large scale project. In this scenario project duration is 2 years and Success Critical stockholders were able to assign TOP value on the scale of 1 to 10. The TOP value weight-age was 65%, 15%, and 25% for Technical, Organizational and People respectively. See the detail of scenario in table14.

Case 1st : Risk Analysis					
TOP Weight-age % Setting			**Value Scale Setting for SCS**		
Technical	**Organizational**	**People**	**Technical**	**Organizational**	**People**
65%	15%	25%	10	10	10
Process/Model Threshold		**Project Duration**		**Simulation Type**	**No of Realization**
5		2 years		Probabilistic	1
Pre – Condition		- Risk has already identified - Concerned SCSs are identified			
Post – Condition		- N number of risks was qualify for treatment			

		- X – N risk was entered in monitoring phase where X is total risk identified.			
Case 2nd : Risk Treatment					
TOP Weight-age % Setting			**Value Scale Setting for SCS**		
Technical	**Organizational**	**People**	**Technical**	**Organizational**	**People**
65%	15%	25%	10	10	10
Process/Model Threshold		**Project Duration**		**Simulation Type**	**No of Realization**
5		2 years		Probabilistic	1
Pre – Condition		- Risk identified and has crossed the process threshold - Treatment alternatives have been identified and rated/valued by concerned SCSs			
Post – Condition		- Risk was treated with highest valued treatment alternatives			
Case 3rd: Risk Monitoring					
TOP Weight-age % Setting			**Value Scale Setting for SCS**		
Technical	**Organizational**	**People**	**Technical**	**Organizational**	**People**
65%	15%	25%	10	10	10
Process/Model Threshold		**Project Duration**		**Simulation Type**	**No of Realization**
5		2 years		Probabilistic	1
Pre – Condition		- Risk identified and have not cross the threshold			
Post – Condition		- Risk was monitored regularly on the basis of TOP values assigned by concerned SCSs, until it does not cross the threshold to qualify for treatment			

Table 12: Scenario two

Although this scenario has different setting and initial values to run than "scenario one" but pre/post condition was remain same.

7.1.3 Results and Observations

During this experimental study model was run for two hypothetic projects with duration one and 2 years. Overall numbers of runs were made against threshold 4 and 5.In each run different TOP values were assigned to risk and requirements. In scenario one and two 8 and 15 risks qualified for treatment when threshold value was 4 and 5 respectively. The table 15 shows the total number of risks occurred and total number of risks treated in every scenario.

	Risk Identified	Risk Treated	Risk Not Treated
Scenario 1	15	8	7
Scenario 2	34	15	19

Table 13: Risks Summary in both scenarios

As mentioned earlier that1st Scenario was design to observe the VRRM RSM behavior on the small scale project, so graph in figure 23 depict the risk identification trends in project duration of one year.

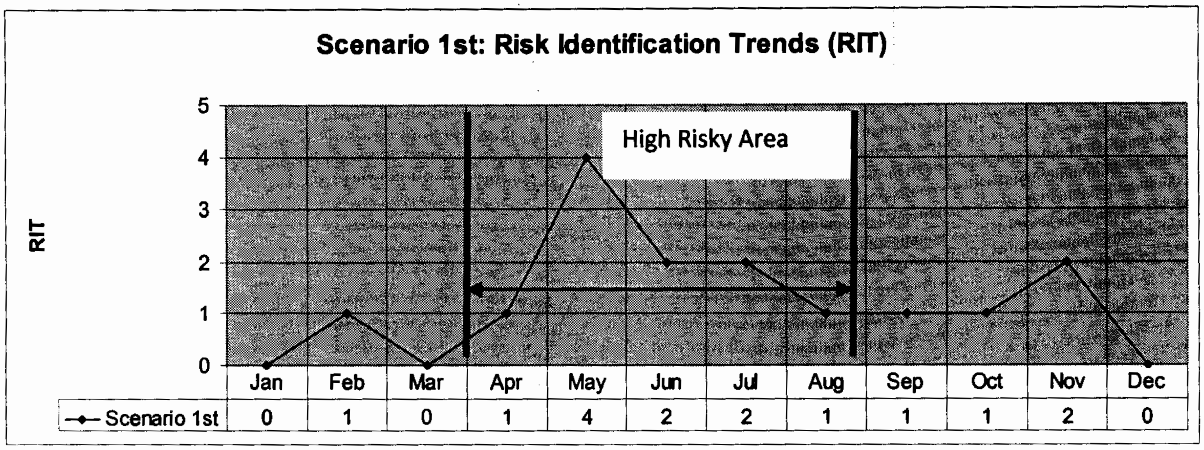


Figure 22: Risk Identification Trends in Scenario one

Risk identification rate increases from mid of March to end of August.

Second scenario was run on project duration of two year. This scenario has different parameter values than scenario one except pre/post condition but the risk identification behavior was same as in scenario one. See graph in figure 24.

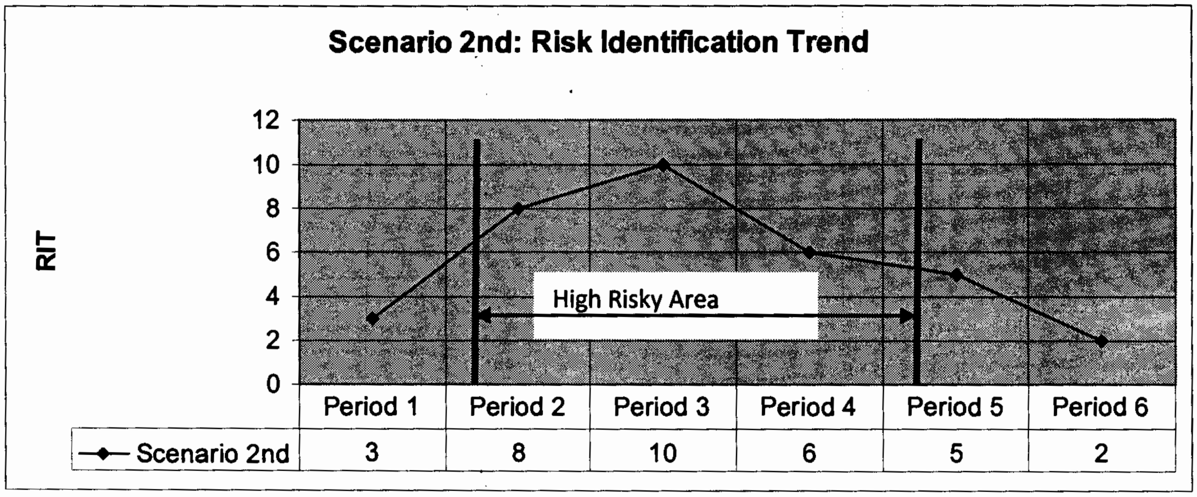


Figure 23: Risk Identification Trends in Scenario Two

In second scenario risk identification remains high from “Period 2” to “Period 4”. So from this behavior, it is clear that requirement related risks can increase in the middle of project if and only if the project duration is one year. The reasons of the this behavior can be due to new or change requirement as in initial stage of project some time client remain unclear about the exact requirement or reluctant to communicate due to some organizational policy.

In the RSM model once risk identified, after analysis it was send for treatment if and only if it crosses the process / model threshold otherwise it was sent to monitoring activity where it was monitor regularly on weekly basis. Monitoring activity run parallel with other activities to keep the track of risk criticality. During monitoring activity if risk threshold crosses the process / model threshold it was consider for treatment. In 1st scenario risk monitoring trends observed as in figure 25.

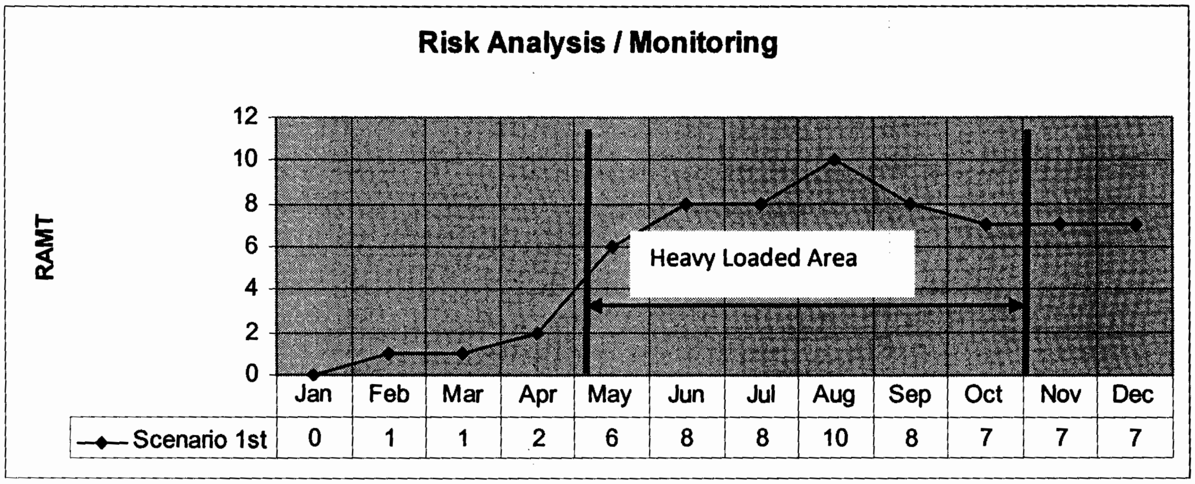


Figure 24: Risk Analysis & Monitoring Trends in scenario one

It was observed that load on monitoring and analysis activity start increasing after 1st quarter, but it remains constant in last quarter of 1st scenario. Similarly in the 2nd scenario same behavior observed in figure 26.

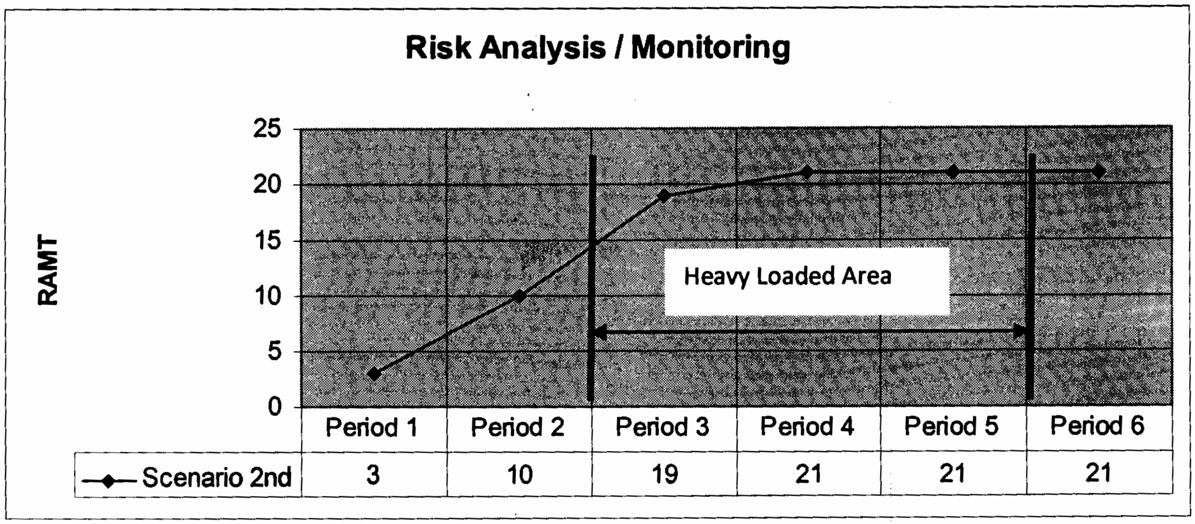


Figure 25: Risk Analysis & Monitoring Trends in scenario two

Treatment activity was a major activity of the RSM. In both scenario different treatment trends were observed. In 1st quarter of first scenario no risk qualify for treatment but second scenario two risk treated in first two quarter. See risk treatment trends in figure 27 of 1st scenario.

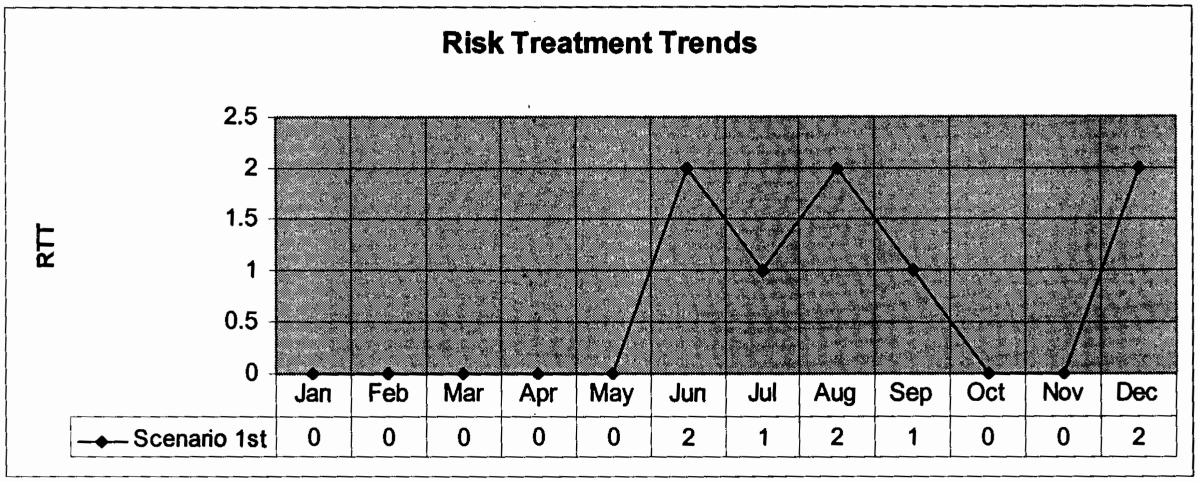


Figure 26: Risk Treatment trends in scenario one

In both scenario treatment activity was heavily load in middle of the project. See figure 28.

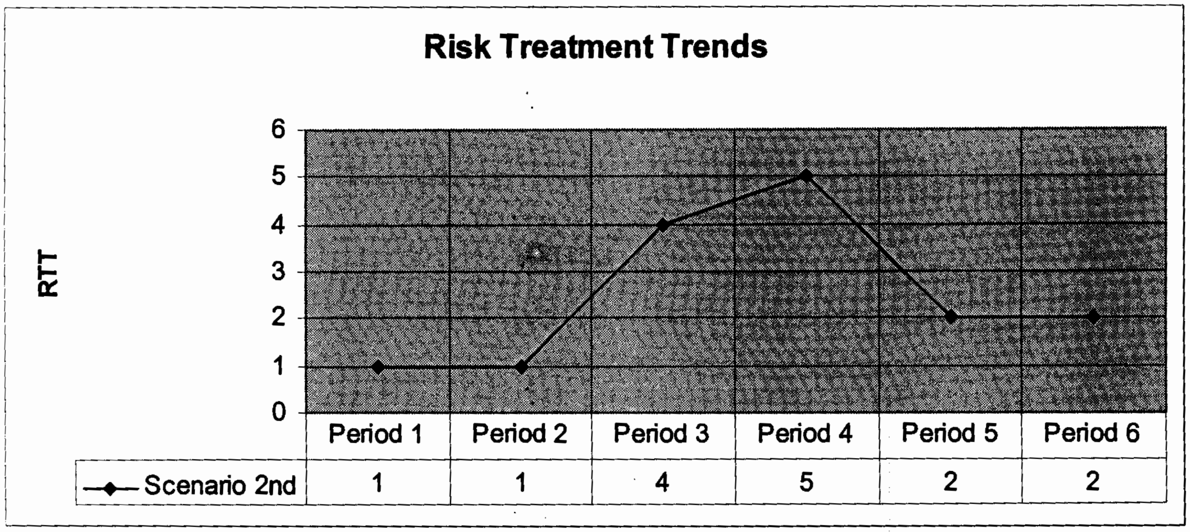


Figure 27: Risk Treatment trends in scenario two

During these simulation runs an inversely proportional relationship observed between threshold and total number of risks treated. That is low values of process / model threshold lead to more no of risks qualification for treatment in contrast with the high values that lead to the less number of risks to be treated, and percentage decreases along with increase in threshold. So following expression shows the exact relationship between threshold and percentage of risks treated.

$$\text{\% of Risks Treated} \propto \frac{1}{\text{Threshold}}$$

Moreover in both scenario positive as well as negative risk maturity trends were observed at given time period T1 and T. For this sake the data (net risk value) of each risk that entered in monitoring phase was collected on weekly basis. The figure 28 shows positive maturity of a risks (R1, R2) at given time.

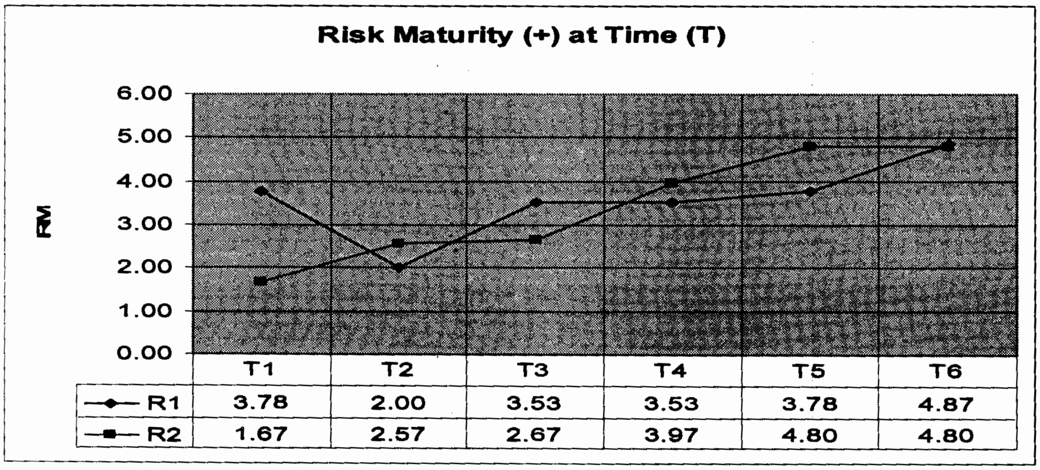


Figure 28: Positive Risk Maturity at time (T)

The figure 29 depicted the negative maturity of risks (R3 and R4) at time T.

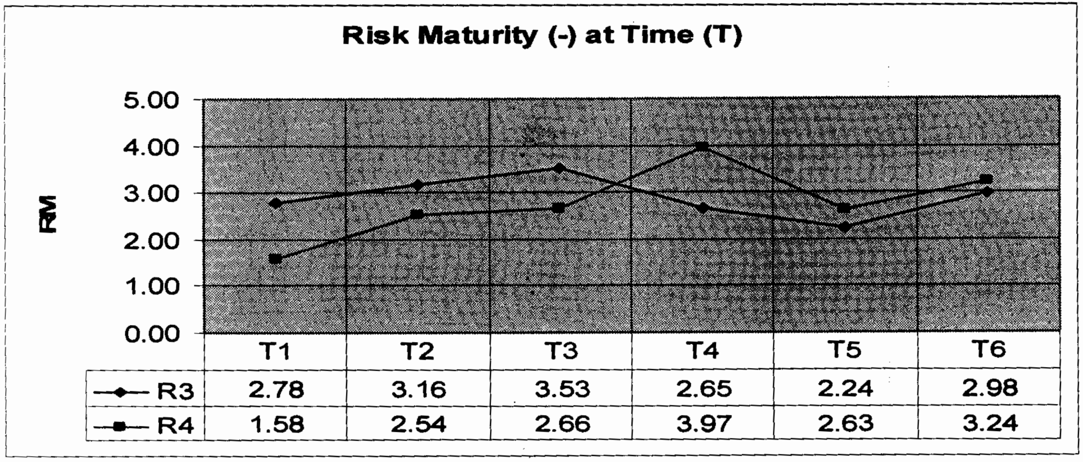


Figure 29: Negative Risk Maturity at time (T)

Intention to represent net risk value with a straight line was to drive a relation for the rate of maturity of risk between any two time periods, on the base of slope of the line representing the net risk value at consecutive day.

The equation above is the verbatim form of equation of slope (rate of change) of line where time period was taken as horizontal axis and net requirement risk vale along y-axis. During these both simulation runs it is also observed that no of requirements linked with risk impact the risk maturity positively and negatively depending on the net value of require requirement.

After that simulation model was run for 100 realizations against each threshold (4 and 5) to predict the probability of risk qualification for treatment. Blew graph plots the probabilities of risk qualify for treatments. In 1st scenario risk treatment probability was 0.01 % as depicted in table blew.

Probability	Value
0.99	false
0.01	true

In 2nd Scenario risk treatment probability was 0.02 % as depicted in table blew.

Probability	Value
0.98	false
0.02	true

7.2 Process Improvement Areas

VRRM introduce new concept to detail with requirement related risk, however few improvement can make it more detail and more effective model:

I. VRRM waits for a risk to cross threshold. Until a risk does not cross threshold, VRRM does not accept it for treatment. However sometimes a risk does not cross threshold but it is believed (based on previous history, organizational context, availability of resources etc.) that this risk was become critical at next stage, so we should handle it immediately instead of waiting it to cross threshold.

II. If some or all risks cross threshold with same net value then VRRM provides no mechanism to prioritize the risks to decide for which risk to handle first. E.g. consider the risks ids 50 and 51 both risks cross threshold with net risk value of 5.33 now there is no mechanism present in VRRM to decide upon which risk was be dealt first. Similarly risk IDs 233 and 254 cross the threshold with net risk value of 5.00 and we cannot prioritize them for treatment.

III. Same is the case with risk treatment alternatives. VRRM lacks the prioritization mechanism to provide best treatment alternatives in case all the proposed alternatives get the same net value. E.g. in case of risk ID 3, the alternative IDs 5.09, 16.70 and 3.10 and 13.60 qualified for being treatment. However note that both IDs 3.10 and 13.60 have the same aggregate value of 8. Now among them which treatment alternative to execute, VRRM does not provide any idea.

IV. Threshold should be fixed during the entire run of VRRM and changing the threshold overloads the model with bulk of risks to become candidate for immediate treatment.

V. VRRM accepts a risk for treatment only in case a net risk value crosses or equals the threshold; however a risk is not treated if the net risk value is less than but close to threshold.

7.3 No mechanism has been provided by VRRM to select the threshold value, i.e. on which factors threshold value must be based, VRRM does not provide any help.

8 – References

Abdel-Hamid. T., & Madnick, S. (1991). Software project dynamics an Integrated approach, Prentice-Hall, Englewood Cliffs, New Jersey

Andersson, C., Karlsson, L., Nedstam, J., Host, M., & Nilsson, B. I. (2002). Understanding software processes through system dynamics simulation: A Case Study. Proceedings of 9th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 41-4.doi: 10.1109/ECBS.2002.999821

Andersson, A., Runeson, P., & Thelin, T (2004).Evaluating the impact of software process simulations - a case study. 5th International Workshop on Software Process Simulation and Modeling (ProSim 2004)" W11L Workshop - 26th International Conference on Software Engineering, 2003(211), 217–221. doi:10.1049/ic:20040461

Averill, M. Law. (2008). How to build valid and credible simulation models. Proceedings of the 40th Conference on Winter Simulation, 39-47. ISBN: 978-1-4244-2708-6

Bandinelli, S., Fuggetta, A., Lavazza, L., Loi, M., & Picco, G.P (1995). Modeling and improving an industrial software process. IEEE Transactions on Software Engineering, 21(5), 440-454. doi:10.1109/32.387473

Bengee, L., & James, M. (2004).Multi-Project management in software engineering using simulation modeling. Software Quality General, 12(1), 59-82. doi:10.1023/B:SQJO.0000013359.71560.47

Basit. A., Murtaza. G., & Ikram, N. (2009).Validation of VRRM Process Model. http://www.wseas.us/e-library/conferences/2010/Cambridge/SEPADS/SEPADS-31.pdf

Christie, A.M., Brown, N.W., & Joe, B.W (1980). Comparison of experimental and analytic simulations of reactor structural response to a simulated core disruptive accident. Nuclear Engineering and Design, 60(2), 257-266. doi: 10.1016/0029-5493(80)90240-X

Chi, Y. Lin., Tarek, Abdel-Hamid., & Joseph, S. Sherif. (1997).Software-engineering process simulation model (SEPS). Journal of Systems and Software, 38(3), 263 – 277. ISBN: 0164-1212

Christie, A.M. (1999). Simulation in Support of CMM-based Process Improvement. Journal of Systems and Software, 46(2), 107-112(6). doi: 0.1016/S0164-1212(99)00004-7

David, M. Raffo. (1999). Getting the Benefits from Software Process Simulation. Conference on Software Engineering and Knowledge Engineering (SEKE'99), Held in Kaiserlautern, Germany

D. Poole., AE. Raftery. (2000). Inference for Deterministic Simulation Models: The Bayesian Melding Approach. Journal of the American Statistical Association, 95(452)

D. Poole., AE. Raftery. (2000). Simulation: An Enabling Technology in Software Engineering. Journal of the American Statistical Association, 95(452)

David, M. R., & Marc, I. K (2000). Empirical analysis in software process simulation modeling. The Journal of Systems and Software, 53 (1), 31–34.

David, M. R. (2005). Software project management using PROMPT: A hybrid metrics, modeling and utility framework. Information and Software Technology, 47(15), 1009-1017

Eickelmann, N.S. (2001). Empirical studies to identify defect prevention opportunities using process simulation technologies. Software Engineering Workshop, 2001. Proceedings. 26th Annual NASA Goddard, 22 – 25. doi: 10.1109/SEW.2001.992651

Eliza, C. (2003). Simulations of very early life cycle quality evaluation, the softgoal Simulation Tool. MS Dissertation, University of British Columbia. http://www.ece.ubc.ca/%7Eelizac/vio /papers/thesis.pdf

George, P. R. (1991). System dynamics: simulation for policy analysis from a feedback perspective, Qualitative Simulation Modeling and Analysis (Ch. 7), Fishwich and Luker, eds., Springer-Verlag, p144 - 169

James, E. T., & Harvey, K. H. (July 1989). Software project management. Curriculum Module, CMU/SEI-89-CM-021

Lin C, Abdel-Hamid T, & Sherif J. (February 1992). Software-engineering process simulation model. TDA Progress Report 42-108, Jet Propulsion Laboratories

Law, A.M., & McComas, M.G. (2001). How to build valid and credible simulation models. Simulation Conference, 2001. Proceedings of the winter, vol.1, 39-47. Doi:10.1109/WSC.2001.977242

Margaret, E. G., (1992). Designing and evaluating games and simulations: A process approach. London: Kogan Page Ltd.

Madachy, R. A. (1994). Software project dynamics model for process cost, schedule and risk assessment, Ph.D. Dissertation, Department of Industrial and Systems Engineering, USC, p-127

Metzger, P.W., & Boddie, J. (1996). Managing a Programming Project (3rd Ed.). Prentice Hall, Englewood Cliffs, New Jersey.

Madachy, R.J. (1996).System dynamics modeling of an inspection-based process. Proceedings of the 18th international conference on Software engineering, p376-386. doi:10.1109/ICSE.1996.493432

Marc, I. K., Raymond, J. M., & David, M. R. (1999). Software process simulation modeling: Why? What? How? Journal of Systems and Software, 46(2-3), 91-105: doi: 10.1016/S0164-1212(99)00003-5

M. Müller., Dietmar. Pfahl. (2008). Simulation Methods, Guide to Advance Empirical Software Engineering. Springer, 117-152 : DOI: 10.1007/978-1-84800-044-5_5

N.T. Koussoulas., S. Daskalaki. (1993). Knowledge based simulation in systems analysis and design. International Conference Proceedings Systems Engineering in the Service of Humans, vol.4, 409-412. 10.1109/ICSMC.1993.390747

Paulk, M.C., Weber, C.V., Garcia, S.M., Chrissis, M.B., & Bush, M. (1993). Key practices of the capability maturity model for software, Version 1.1. Technical Report SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA

Pagendarm, H. G., & Posts, E. H. (1994). Comparative Visualization - Approaches and Examples, Fifth Eurographics Workshop on Visualization in Scientific Computing, Rostock, Germany, May 30 - June 1, 1994

Pfahl, D., & Lebsanft, K. (2000). Using simulation to analyse the impact of software requirement volatility on project performance. Information and Software Technology, 42(14), 1001-1008. doi: 10.1016/S0950-5849(00)00152-X

Raffo, D.(1996). Modeling Software Processes Quantitatively and Assessing the Impact of Potential Process Changes on Process Performance, Ph.D. Dissertation, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, Pennsylvania, p-341

Raffo, D.M., Vandeville, J.V., & Martin, R.H. (1999). Software process simulation to achieve higher CMM levels. Journal of Systems and Software, 46(2), 163-172(10). doi: 10.1016/S0164-1212(99)00009-6

Raymond, M. (2006). Simulation for business value and software process/product tradeoff decisions. Proceedings of the 2006 international workshop on Economics driven software engineering research Shanghai, China, 25 - 30. ISBN: 1-59593-396-4

Sargent, R. G. (2005). Verifying and Validating Simulation Models. Proceedings of the 37th conference on winter simulation, 130-143. ISBN: 0-7803-9519-0

Samad, J., Ikram, N., & Muhammad, U. (2008). VRRM: A Value-based Requirements' Risk Management Process. Proceedings of the IASTED International Conference on Software Engineering, 184-191. ISBN:978-0-88986-716-1

Saurabh. K., Krishnan. B. (2009). Software Engineering: A System Dynamics Simulated Pedagogical Practice. IEEE conference on Software Engineering Education and Training, 280 - 283. 10.1109/CSEET.2009.59

Tezduyar,T., Aliabadi,S., Behr, M., Johnson, A., Kalro, V., & Litke, M. (1996). Flow simulation and high performance computing. Computational mechanics 16(6), 397-412. DOI: 10.1007/BF00350249

Weber, J.D., Overbye, T.J., Sauer, P.W., & DeMarco, C.L (1998). A simulation based approach to pricing reactive power. Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, 3, 96 – 103. doi:10.1109/HICSS.1998.656072

Zhang, H., Kitchenham, B., & Jeffery, R. (2007). A framework for adopting software process simulation in CMMI organizations. Proceedings of the 2007 international conference on Software process, 4470/2007, 320-331: doi: 10.1007/978-3-540-72426-1_27

9 – Glossary

VRRM: Value base requirement risk management model

RIdentification: Risk Identification

RAnalysis: Risk Analysis

RTreatment: Risk Treatment

RSM: Reference Simulation Model

SCS: Success Critical Stockholders

Appendix A

```
class ManageRiskValue : public SExtFcnBase
{
public:
        static long cnt;
        ManageRiskValue() { }
        // Override of SExtFcnBase virtual method to return the version
number.
        virtual double GetVersion() const { return 1.02; }
        // Override of SExtFcnBase virtual method to return the number of
input arguments.
        virtual int GetNumInputs() const { return 4; }
        // Override of SExtFcnBase virtual method to return the number of
output arguments.
        virtual int GetNumOutputs() const { return 2; }
        // Override of SExtFcnBase virtual method to perform the calculation.
        virtual StatusID Calculate1(const double* inargs, double* outargs)
        {
                cnt+= rand()%10;
                outargs[0] = rand() % (int)(inargs[1] - inargs[0] + 1) +
inargs[0];
                //outargs[0] = (rand() % 100) + 90 ;
                outargs[1]=cnt;
                return SUCCESS;
        }
virtual StatusID Calculate2(const double* inargs, double* outargs)
        {
                cnt+= rand()%50;
                outargs[0] = rand() % (int)(inargs[1] - inargs[0] + 1) +
inargs[0];
                //outargs[0] = (rand() % 100) + 90 ;
                outargs[1]=cnt;

                return SUCCESS;
        }
virtual StatusID Calculate3(const double* inargs, double* outargs)
        {
                cnt+= rand()%100;
                //outargs[0] = (int)(::GetTickCount()* inargs[2]+cnt) %
((int)inargs[1]-(int)inargs[0]) + (int)inargs[0] ;
                outargs[0] = rand() % (int)(inargs[1] - inargs[0] + 1) +
inargs[0];
                //outargs[0] = (rand() % 100) + 90 ;
                outargs[1]=cnt;
                return SUCCESS;
        }

};

long ManageRiskValue::cnt=0;
//-------------------------------------------------------------------------
---
// Entry point for add_mult_scalars call from a GoldSim External element.
//
extern "C" void __declspec(dllexport) get_values(const int      methodID,

int*           status,
```

```
const double*  inargs,

         double*        outargs)
{
      // Create an instance of ManageRiskValue to perform the
initialization
      // and calculation.
      ManageRiskValue sExternal;
      *status = sExternal.ProcessRequest(methodID, inargs, outargs);
                                          }
```