This paper presents a technique to teach the continuous risk management (CRM) paradigm using a requirement management tool. The paper explains the CRM process and shows how to implement the risk information using a requirement management tool. Teaching students the continuous risk management implementation is important to show them how the tool will: assist the project management and team members to establish and use consistent documentation; instantiate and store each identified risk; associate for each risk a mitigation or task plan; and visually presents each risk with the capability to be tracked, watched or mitigated throughout the project’s iterative life cycle. The IBM Rational Suite Enterprise RequisitePro tool was used to show the students how to capture and store the organizational and management system risk knowledge into a database. The students gain hands on risk management knowledge that can be used for product, process and project improvement. They learn how to write risk statements, collect risk metrics, and capture the risk lessons learned for future projects.

Introduction

The focus of this paper is to show students how to apply a requirements management tool, IBM Rational Suite Enterprise RequisitePro (Rational 2003), to a Continuous Risk Management (CRM) paradigm (Dorofee 96). The RequisitePro is one of the integrated tools in the IBM Rational Suite Enterprise. IBM Rational Suite Enterprise is designed for software development project managers and software development teams. This tool was employed because it was made available to the University of Houston Clear Lake through the IBM Scholars Program. The RequisitePro tool is designed to support the entire requirements management process throughout a project’s life cycle. The artifacts and requirements information collected for a project are so similar to the risk artifacts that the tool can be applied as a risk management tool. The RequisitePro tool has the ability for a project team to: instantiate template documents; label, classify and group risks taken from an information sheet; track the progress of each risk or group of risks; evaluate risks activities; and maintain schedules and traceability of risks being tracked, watched, or mitigated. The RequisitePro tool also maintains this information in a user-selected database. The database is used to combine and consolidate similar risks to minimize duplication, workload and time needed to resolve risks sets. Assuming the project has already purchased the RequisitePro tool for requirements, the tool will also support the entire continuous risk management activities of identifying analyzing, planning, tracking, controlling, and documenting the artifacts.

Risk Management Paradigm

The CRM paradigm was derived from the Carnegie Mellon Software Engineering Institute (SEI) and is defined in their CRM Guidebook (Dorofee, 96). CRM can be applied to hardware, software, and systems. Some of the tools and techniques for each discipline may change and have different names, but the risk management process is similar. Maintaining a corporate risk database allows reuse of successful risk resolution strategies and a knowledge base of lessons learned. (Hall 97)

Roger L. Van Scoy developed the Carnegie Mellon Software Engineering Institute (SEI) risk management paradigm in 1992 (Van Scoy 92, p.9). The paradigm illustrated in Figure 1 is a set of functions that are identified as continuous activities throughout the life cycle of a project. The paradigm serves as a model indicating how the different elements
of risk management interact and also as a framework for describing how risk management can be implemented. The paradigm has a circular form to highlight its continuous nature. The arrows signify the logical flow of information between the elements of the paradigm. Communication is the center of the paradigm. It is the means by which all information flows.

![Continuous Risk Management Paradigm](image)

**Figure 1. Roger L. Van Scoy’s Continuous Risk Management Paradigm.**

Van Scoy summarized the elements in his paradigm as:

- **Identify** - locate risks before they become problems and adversely affect the program.
- **Analyze** - turn the raw risk data into decision-making information.
- **Plan** - turn the risk information into decisions and actions (both present and future).
- **Track** - monitor the status of risks and actions taken against risks.
- **Control** - correct for deviations from the planned risk actions.
- **Communicate** – provide feedback on the active risk activities, current risks, and emerging risks among the paradigm elements and within the program. The documentation was added to the paradigm by (Dorofee 96).

The Continuous Risk Management paradigm illustrates a set of functions that are identified as continuous and iterative activities throughout the life cycle of a project. The paradigm is a conceptual, or abstract, view of risk management. NASA has adopted this paradigm in guideline NPG 7120.5A (NASA 2003).

**Risk identification** is the first element in the risk management paradigm. The goal of risk identification is to identify the risks to be managed before they can adversely affect a program and to incorporate this information into the project management process (Rosenau 92). The risk team uses techniques to discover risks by exploiting the collective knowledge of the program team. Since each member of the program team has a particular knowledge about the project, anyone involved can be useful in identifying risks.

**Risk analysis** is the second element in the risk management paradigm. The purpose of risk analysis is to convert risk data into useable risk management information for determining priorities and making decisions. Each risk must be understood sufficiently to allow a manager to make decisions. Risk analysis sifts the known risks, and places the information in the hands of the decision maker. Analysis provides the information that allows managers to work on the right risks (Air Force 88).

**Risk planning** is the third element in the risk management paradigm. This element includes developing actions to address individual risks, prioritizing risk actions, and orchestrating a risk action plan for each risk. An individual risk action plan could take many forms, for example:

- Mitigate the impact of the risk by developing a contingency plan (with a triggering event) should the risk occur.
- Avoid a risk by changing the product design.
- Accept the risk and take no further action, thus accepting the consequence if the risk occurs.
- Study the risk further to acquire more information and better determine the uncertainty or loss associated with the risk.
The key to risk planning is to translate risk information into planning decisions and mitigating actions (both present and future), and implementing those actions. “Attempts to plan for the elimination of all risks are almost always futile efforts” (Charette 89).

**Risk tracking** is the forth element in the risk management paradigm. The purpose of risk tracking is to collect accurate, timely, and relevant risk information and to present it in a clear and easily understood manner appropriate to the personnel or group who receives the status report (Dorofee 96). Risk tracking is required to ensure effective action plan implementation. This means that the risk team must devise the risk metrics and triggering events needed to ensure that the planned risk actions are working. Tracking is the watchdog function of the risk action plan. Tracking is done by the person(s) responsible for monitoring “watched” or “mitigated” risks. Project personnel use the status report information, generated during tracking, in the control function of the paradigm to make decisions about managing risks.

**Risk control** is the fifth element in the paradigm. Once the risk metrics and the triggering events have been chosen, there is nothing unique about risk management. Rather, risk management melds into program management and relies on program management processes to control the risk action plans, correct for variations from the plans, respond to triggering events, and improve the risk management process. In fact, if risk management is not integrated with day-to-day program management, it will soon be relegated to an ineffective background activity (Hall 97, Ch. 18).

**Risk communication** is at the center of Van Scoy’s risk management paradigm because, without effective communication, no risk management approach is viable (Van Scoy 92). Communication is critical because it facilitates interaction among the elements of the paradigm. There are higher-level communications to consider as well. Risks must be communicated to the appropriate organizational levels so the risks can be analyzed and managed effectively. This includes levels within the development organization, within the customer organization, and most especially, across that threshold between the developer and the customer. Communication is present in all paradigm functions and is essential for managing risks. Communication of risk information is often difficult because the concept of risk deals with probability and negative consequences.

**RequisitePro**

Rational RequisitePro is a requirements repository tool that organizes requirements and provides traceability and change management throughout the project life cycle. Rational (Rational 2003) defines a requirement as “a condition or capability to which the system must conform.” The risk statement is similar to the requirement statement. The intent of the risk statement is that it be clear, concise, and sufficiently informative that the risk is easily understood. The risk statements in standard format shall contain two parts: the condition and the consequence (Gluch 94a). The condition-consequence format provides a complete picture of the risk, which is critical during mitigation planning. The risk statement is read as follows:

given the <condition>; there is a possibility that <consequence> will occur.

A RequisitePro project is defined as a requirements database and its related documents. A project manager determines the project structure, sets up security permissions for the project’s users, and creates a RequisitePro project. Each RequisitePro project has its own database, where all the requirements for a project are stored. In the project database, requirements can be added, modified, or deleted. When requirements are changed, the changes are updated in the database. These project activities and database can easily be applied to the CRM paradigm.

The currently supported databases are: Microsoft Access, Oracle and Microsoft SQL Server. The back-end database used depends on the size of the project team, location, logged-on users, and cost constraints. For small work groups Microsoft Access is recommended and was used for this research.

RequisitePro has version control to let the project manager trace change by archiving projects. Version control helps the project manager keep a record of changes to project files during the life cycle. Risk attributes must be ranked, tracked, mitigated or deleted.

The Word Workplace is the file within RequisitePro where requirements are created and modified in a document. These can be RequisitePro documents or Word documents. The Views Workplace is a window to the database. Requirements, their attributes, and their relationships to each other are displayed and managed in views. The requirement Workplace thus becomes the risk workplace. RequisitePro includes a Web interface, making
requirements accessible to all team members, especially in remote locations or in a multi-platform environment (Van Epps 2002).

Views present information about the project, a document, or requirements graphically in a matrix or in an outline tree. Views display the attributes assigned to requirements, such as status and priority, or the relationships between different types of requirements (similar to a set of risks). The views can be grouped in packages and traced to one another. RequisitePro has three kinds of views:

The Attribute Matrix View displays all requirements (risk) of a specified type. The requirements are listed in the rows, and their attributes appear in the columns.

The Traceability Matrix View displays the relationships (traceability) between types of requirements (risk).

The Traceability Tree displays the chain of traceability to or from requirements (risk) of a specified type.

A requirements document is a specification that captures requirements, describes the objectives and goals of the project, and communicates development effort. The Risk Management Plan, Risk Implementation Plan, and Detailed Risk List are similar documents that have existing formatted templates in Rational Suite Enterprise. Any Word document can be associated with a project and made available in the document list when a project is opened. This includes the risk mitigation and task plan documents.

Requirement type is a template for inserting the projects requirements. This pull-down window view is employed as a template for inserting the projects risks. Requirement types are used to classify similar requirements so they can be managed, defined in a common set of attributes, display style, tag numbering, and more.

With this overview of the Rational RequisitePro capabilities, it is easy to identify similarities and substitute the risk statements for the requirement statements and input the contents contained in each risk information statement to the RequisitePro Views.

The risk information sheet records the information gathered during each of the paradigms function. Figure 2 is an example format used for a risk information sheet. The contents in the fields of the risk information sheet are the values input into the views and packages managed by RequisitePro. A mitigation or task plan format was also developed for each risk that is mitigated and tracked. The mitigation or task plan is also stored in the database and tracked by the associated risk ID.

<table>
<thead>
<tr>
<th>RISK ID</th>
<th>Risk Information Sheet</th>
<th>Date Identified:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Risk Statement</td>
<td></td>
</tr>
<tr>
<td>Probability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timeframe</td>
<td>Originator</td>
<td>Classification</td>
</tr>
<tr>
<td>Context</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approach: Research / Accept / Watch / Mitigate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contingency Plan and Trigger</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Date_________</td>
<td></td>
</tr>
<tr>
<td>Lessons Learned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approval</td>
<td>Closing Date</td>
<td>Closing Rationale</td>
</tr>
</tbody>
</table>

**Figure 2. Example Risk Information Sheet and Fields.**

During **IDENTIFY** the following fields are completed by the project team members:
- **ID**: unique identifier for the risk, numeric, or alphanumeric; assigned by project or organization or CM office
- **Identified**: date when the risk was identified
- **Statement**: statement of the risk
A hypothetical project manager scenario is depicted to identify the possible risk associated with him and the project. One potential risk is identified as an example. The risk is stated and implemented into a RequisitePro project named CRM_Risk. Six views show a sample of the RequisitePro tool capabilities applied to the risk management project. The six views are:
1. Microsoft Access Risk Management Document Database  
2. Risk Properties Dialogue Box  
3. Risk List Attribute Matrix  
4. Risk Mitigation Attribute Matrix  
5. Mitigation Traced From Risk Matrix  
6. Risk Actions Traced Into Risks  

The following example information concerns the project manager and the software engineers working on the software system the team will build. This is the first system the project manager has managed of this magnitude and complexity. The project manager believes it is going to be a very positive experience for him and the rest of the software engineers on the project. All his other projects were hardware control systems prior to this. The software engineers working for the manager are entry-level people just out of college. The project manager thinks that software can fix just about any problem the hardware group comes up with. The waterfall life cycle model (Royce 1970) is what the project manager will use on this project and he foresees no problems with using this model.

The project is in the requirements stage of the life cycle, the project manager thinks one of the most exciting opportunities of a new project is that this is the first project at the company to use object oriented design and the C++ programming language. Every single one of the software engineers has the chance to learn something new on this
Given this scenario the CRM_Risk Project was opened and new documents were created before the IDENTIFY stage began. Figure 3 is a Microsoft Access window displaying four template documents created for the project.

The documents are:
1. Risk Management Plan
2. Risk Implementation Plan
3. Risk List Document
4. Risk List Glossary

There are numerous risk statements that can be obtained from this scenario. The RISK1 condition-consequence statement is the risk selected and shown in the six RequisitePro views.

This is the first time that the software engineers will use OOD; the engineers may have a lower-than-expected productivity rate and schedules may slip because of the associated learning curve.

The following risk context statement (Higuera 93) taken from the Risk Information Sheet is associated with the RISK1 OOD risk statement.
Object oriented development is a very different approach that requires special training. There will be a learning curve until the entry-level software engineers are up to speed. The time and resources must be built in for this or the schedule and budget will be overrun.

By creating a Package, Risk-Types, each risk is entered using the Requirements pull down menu view. The Dialogue box Figure 4 is used to create or revise a risk gathered from the risk information sheet. The risks are entered under General using Type, Name, and Text, where:

- Type: The new risk inherits the risk type (RISK) displayed in the Attribute Matrix.
- Name: Displays the name associated with the risk.
- Text: Displays the risk condition and consequence statement.

The engineer has five additional fields that are input based on the iterative stage of the CRM process. The five fields are:
1. Revision
2. Attributes
3. Traceability
4. Hierarchy
5. Discussion

For example, the Traceability view is input during the Track stage. The Hierarchy can be used to establish the importance of this risk within the risk collection. The Discussion displays the detailed context statement.

As the risk engineering team progress through the IDENTIFY and ANALYZE stages they develop the Risk List Document using information taken from the fields of the Risk Information Sheet. The Risk List Document contains the detailed information about each risk and serves as a deliverable. As the risks are entered into the Package, Risk-Types, the risk list is displayed in a matrix view. Figure 5 Risk List Attribute Matrix provides the team members, working on the CRM_Risk project, the risk list view. Risks are arranged in rows and listed by a risk number. Attributes are arranged in the following columns:

1. Difficulty of Detection
2. Likelihood
3. Potential Impact
4. Overall Risk
5. Notes
These are the attributes assigned in the PLAN, TRACK, and CONTROL stages of CRM paradigm. A text pane, located at the bottom, displays the description of a risk in its entirety the risk statement is also shown in the lower left view element.

**Figure 5. Risk List Attribute Matrix**

During the PLAN stage the team, using the Package, Risk Mitigation Action, builds a Mitigation Attribute Matrix for each risk decided to mitigate. Two similar Packages (Risk Watch Action, Risk Accept Action) would be instantiated for risks to watch and accept. Figure 6 presents the tree and matrix views. There are five fields to input attributes:

1. Approval
2. Owner
3. Trigger
4. Cost
5. Notes

The MITIGATE pending box is used to input the next mitigation risk (MITIGATE6 Project Manager). In the example, the text field proposes to assign a new project manager. The information for the selected item is displayed in the bottom lower right section of the view.
During the life cycle of the CRM_Risk project, the TRACK and CONTROL stages use views selected from the Package, Risk Traceability Matrix Views. The Traceability Tree, shown in Figure 7, provides a graphical view of relationships to or from risk or sets of risks, including, direct, indirect and suspect traceability relationships. The risk statements are in the matrix rows and mitigation actions in the columns. Another view PREVENT was developed to plan risk prevention actions and trace to the mitigation actions.
In the Package, Risk Traceability Tree View, shown in Figure 8, are risk mitigate and prevent actions traced to the risks. This view shows each risk, the mitigation plan, and the prevention tactics. For each risk a sliding window gives the mitigation prevention number traced to the risk.

Figure 8. Risk Actions Traced Into Risks

Summary

The overall process flow for the continuous risk management paradigm was presented. Each stage of the paradigm was dissected and the activities inserted into RequisitePro. RequisitePro is a powerful requirement tool, which was easily applied to risk management. The tool helps teams manage project risks comprehensively, promotes communication and collaboration among team members, and reduces project uncertainty.

RequisitePro offers the power of documentation and a database linked to all items of a Project. A very important feature of RequisitePro is that, if any item in a View is changed, the associated items are simultaneously changed in all the other Project Package Views.

Its robust architecture maintains live risk documents that are dynamically linked to a database for trace, sort, and query capabilities. This allows system engineers to easily organize and prioritize their risk, to trace, mitigate, and prioritize relationships between risk, and track changes that affect schedules.

The traceability features visually indicate how changes affect the project, giving engineers the ability to perform real-time impact analysis and allowing them to make informed decisions for management or resource allocation. As a result, the project manager is better able to manage risk, and their projects integrity is maintained. RequisitePro captures the change history for each risk, which providing an audit of the projects risk evolution.

References


The Software Engineering Information Repository (SEIR) [https://seir.sei.cmu.edu/seir/](https://seir.sei.cmu.edu/seir/)


