

# CONDUCTING CONTINUOUS RISK MANAGEMENT USING IBM RATIONAL REQUISITEPRO SOFTWARE

---

James C. Helm, University of Houston-Clear Lake; Ipek Bozkurt, University of Houston-Clear Lake

## Abstract

This paper presents a continuous risk management paradigm, as well as implementing the risk information using IBM's Rational® RequisitePro® software requirement management tool. Risk management implementation is important for the following reasons: the tool assists the Project's management and team members with consistent documentation; instantiates and stores each identified risk; associates 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 lifecycle. The RequisitePro tool allows the capture and storage of the organizational and management system risk knowledge into a database. This risk knowledge is used for product, process and project improvement as well as the collection of metrics and lessons learned for future project references.

## Introduction

Current risk management tools essentially capture and track risks early in a project's lifecycle but fall short of supporting the ongoing activities of tracking, mitigating and documenting the artifacts involved with the entire continuous risk management process. The focus of this paper is to apply a software requirements management tool, IBM Rational RequisitePro [1], to a Continuous Risk Management (CRM) paradigm developed by Van Scoy [2] and improved on by Alberts et al. [3], by adding the communicate and document element to the paradigm.

The RequisitePro tool was designed to support the entire requirements management process throughout a project's lifecycle. 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 to: instantiate template documents; track the progress of each individual risk taken from an information sheet; track, watch and mitigate risk as a team evaluates risks activities; and, maintain schedules and traceability of risks 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.

## Background

### Risk Management

Risk management deals with the fundamentals of knowledge engineering. Risk management provides information to the decision makers and team members before a problem occurs so that risk actions can be taken in order to avoid potential loss. Therefore, it is important that Team Risk Management [4] be an integral part of a project's management plan and not a separate activity.

The project team does a risk assessment at the beginning of a project by identifying a few risks and developing a risk management plan. This risk management plan is then placed in a binder, put on a shelf, and possibly not looked at again. However, risk management is not something a project team or manager should do only once during the project lifecycle. There is no risk management season. Also, the risks identified at the beginning of a project are not necessarily the same risks identified in the middle or near the end of the project. The CRM paradigm was derived from the Carnegie Mellon Software Engineering Institute (SEI) [5] and is defined in their CRM Guidebook [3]. Maintaining a corporate risk database allows reuse of successful risk-resolution strategies and a knowledge base of lessons learned [6]. Three additional areas where the rigorous CRM paradigm could be applied are: 1) virtual hardware risks; 2) both product- and project-related risk; and, 3) occupational safety and health. Some of the tools and techniques for each discipline may have different names, but a risk management process was employed.

The CRM paradigm can be applied to Securing Virtualized Datacenters [7] by developing a risk management plan and using a risk information sheet for each of the virtual security threats (risks) identified: Virtual Machine; Hypervisors Threats; Virtual Infrastructure Threats; and, Virtual Network Threats.

### Risk and Requirements Management Tools

The CRM tools and techniques set can aid in the mitigation and control of product- and project-related risk. Product risk is defined as an unacceptable design solution—something that does not meet technical or customer require-

ments. Project risk is defined as the failure to conform to time and budgetary constraints [8]. The CRM tools and documentation set can also be applied to the area of occupational safety and health. One area of interest is Carpal Tunnel Syndrome (CTS), where few studies have focused on using risk assessment similar to the strain index (SI) as a mechanism to establish and monitor the effects of CTS prevention methods over time (lifecycle). One study established a methodology to reduce CTS incidents by using the SI to identify operations that have high CTS risk [9]. This study would have benefitted from the CRM iterative lifecycle process.

A survey conducted by Smith and Savage [9] shows the state of requirements for engineering tools. The survey results (of 37 vendors worldwide) show a total of 157 features of a variety of requirements for engineering tools. IBM Rational, together with their Rational Doors and RequisitePro tool are included in the 21 tools from the U.S. In order to provide model-driven traceability for software product lines, Anquetil et al. [10] identified different dimensions to track requirements, and included trace import and export, modification, query and visualization capabilities in their framework.

Goknil et al. [11] analyzed semantics of trace relations in requirements models for consistency checking and inferencing. They built a tool to support both checking consistencies of relations and inferring new ones. In addition to the two traceability relation types, RequisitePro can provide their tool accounts for more types of requirements and traceability types. In order to produce and manage quality requirements with aeronautical systems, Abo [12] developed a requirements engineering framework and implemented it using IBM Rational DOORS and IBM Rational Change tools.

## Risk Management Paradigm

Van Scoy [2] developed the SEI risk management paradigm in 1992. The paradigm, illustrated in Figure 1, is a set of functions that are identified as continuous activities throughout the lifecycle 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. Communicate & Document is the center of the paradigm. It is the means by which all information flows.

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 and Document:

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 Alberts et al. [3].



Figure 1. Van Scoy's Continuous Risk Management Paradigm

The Continuous Risk Management paradigm illustrates a set of functions that are identified as continuous and iterative activities throughout the lifecycle of a project. The paradigm is a conceptual, or abstract, view of risk management.

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. 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 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.

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.

Risk tracking is the fourth 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 receiving the status report. Risk tracking is required in order 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.

Risk “communicate and document” is at the center of the risk management paradigm because, without effective communication, no risk management approach is viable. 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.

Project Risk Management is defined as the systematic process of identifying, analyzing and responding to risk, according to the Project Management Institute (PMI). Project risk management is intended to support project managers in managing risk and minimizing the impact of risk on the project outcomes and outputs. Kimbrough and Compagnon [13] analyzed the importance of organizational culture in implementing risk management in organizations, and found that organic cultures make greater progress in implementing a risk management program. Yeo and Ren [14] identified a need for progressive risk management capabilities in inherently complex projects (in terms of tasks and human issues). The framework they proposed is a good application of the risk management paradigm discussed above, since it includes a change management framework that deals with risk planning and control processes, as well as organizational and people contexts of the complex project.

In a recent study conducted by Krane et al. [15] on the relationship between the project manager and the project owner, and the impact of this on project risk management, has shown that the main focus is on operational risks (identified as relating to the project’s direct results). In some cases, however, due to cost justifications, a formal risk management process is not always applied [16]. In a further study conducted by Kutsch and Hall [17], the issue of Information Technology project managers deliberately ignoring certain risks and finding them irrelevant has been addressed and, similar to their previous study, concluded that project risk management, when not applied correctly, may be counterproductive in some cases.

## RequisitePro

Rational RequisitePro is a requirements repository tool that organizes requirements and provides traceability and change management throughout the project lifecycle. Requi-

---

sitePro requirement management software was selected to show how it handles risk (requirement) traceability. Two views will be given: 1) Risk Traceability Matrix View to show risk mitigation traced from the risk matrix, and 2) Risk Traceability Tree View to show risk action traced into a risk. The views show the preventative, contingency and second-level actions traced to the risks. If the risk or the action changes, RequisitePro gives a visible indication (suspect links) of potential impacts to the Project Risk Management Plan and other project activities.

Rational [1] 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 such that the risk is easily understood. The risk statements in standard format shall contain two parts: the condition and the consequence. 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 of 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 study. 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 lifecycle. Risk attributes must be ranked, tracked, mitigated or deleted when they are no longer a risk and have become a problem.

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. Re-

quirements, 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 [19] first presented an Automating Risk Management process with RequisitePro.

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:

1. 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.
2. The Traceability Matrix View displays the relationships (traceability) between types of requirements (risk).
3. 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 project’s 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 paradigm’s functions. 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.

<b>ID</b>	<b>Risk Information Sheet</b>		<b>Identified:</b>
<b>Priority</b>	<b>Statement</b>		
<b>Probability</b>			
<b>Impact</b>			
<b>Timeframe</b>	<b>Originator</b>	<b>Class</b>	<b>Assigned To:</b>
<b>Context</b>			
<b>Approach: Research / Accept / Watch / Mitigate</b>			
<b>Contingency Plan and Trigger</b>			
<b>Status</b>			<b>Status Date</b>
<b>Approval</b>	<b>Closing Date</b>	<b>Closing Rationale</b>	

**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
- *Origin*: Organization or person who identified the risk
- *Context*: Associated information that clarifies the risk

During ANALYZE, the following fields would be completed:

- *Priority*: The priority ranking of the risk
- *Probability*: The likelihood of occurrence—exact value depends on the level of analysis

- *Impact*: The degree of impact—exact value depends on the level of analysis
- *Timeframe*: The timeframe in which action is needed
- *Class*: The classification of the risk (could be more than one value) and the class or group the risk belongs to

During PLAN, the following fields would be completed:

- *Assigned to*: Who is responsible for mitigating the risk
- *Mitigation strategy*: The selected mitigation strategy for mitigating the risk
  - This field can also be used to document the other approaches that can be taken and their information (e.g., research approach with its research plan; watch approach with its tracking requirements; accept approach with its acceptance rationale)

During TRACK, the following fields would be completed:

- *Status/status date*: Running status information that provides a history of what is being done for the risk and of any changes in the risk
- *Probability*: Likelihood of occurrence—exact value depends on type of analysis
- *Impact*: Degree of impact—exact value depends on type of analysis
- *Timeframe*: Timeframe in which the risk will occur or action is needed
- *Priority*: Priority ranking of the risk

During CONTROL, the following fields would be completed:

- *Approval*: Approval for mitigation strategies or closure (for transferred risks, this may require the transferor's signature)
- *Closing date*: The date when the risk was closed
- *Closing rationale*: The rationale for closure of the risk (e.g., probability is zero)

During continuous COMMUNICATE and DOCUMENTATION, the following documents would be selected, initialized, maintained and tracked:

- Risk Management Plan
- Risk Implementation Plan
- Risk List Document
- Risk Analysis Reports
- Risk Mitigation Status Reports
- Risk Database
- Risk Tracking Logs
- Test Reports

## CRM Applied to RequisitePro

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 of 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 lifecycle model [19] is what the project manager will use on this project, and he foresees no problems with using this model.

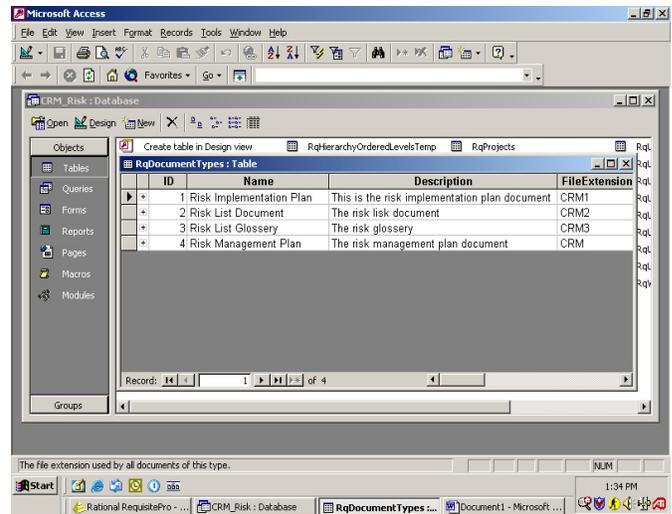
The project is in the requirements stage of the lifecycle; the project manager thinks one of the most exciting opportunities of a new project is that this is the first project at the company to use object-oriented design (OOD) and the C++ programming language. Every one of the software engineers has the chance to learn something new on this project. This will put the software engineers on the forefront of the technology curve and really bring the software team into the future. The manager has also selected one of the newest C++ compilers with all of the latest features to help improve the efficiency of software developments.

Given this scenario, the CRM\_Risk Project was created using a RequisitePro interface to establish a Microsoft Access database in order to store the new document sets that were created before the IDENTIFY stage began. Figure 3 is a Microsoft Access window displaying four template documents instantiated for and used throughout the lifecycle of the project. This figure shows that the RequisitePro tool will support and interface to one of three databases: Microsoft Access, Oracle and Microsoft SQL Server. The back-end

database to be 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 in this study.

The documents are:

1. Risk Management Plan
2. Risk Implementation Plan
3. Risk List Document
4. Risk List Glossary



**Figure 3. Microsoft Access Risk Management Document Database**

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

The following risk context statement 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

By creating a Package, Risk-Types, each risk is entered using the Requirements pull down menu view. The Dialogue box (see 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.

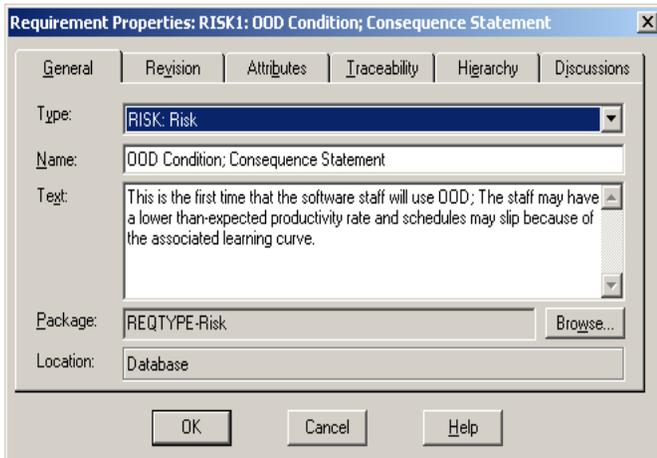


Figure 4. Risk Properties Dialog Box

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 servers 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 the 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.

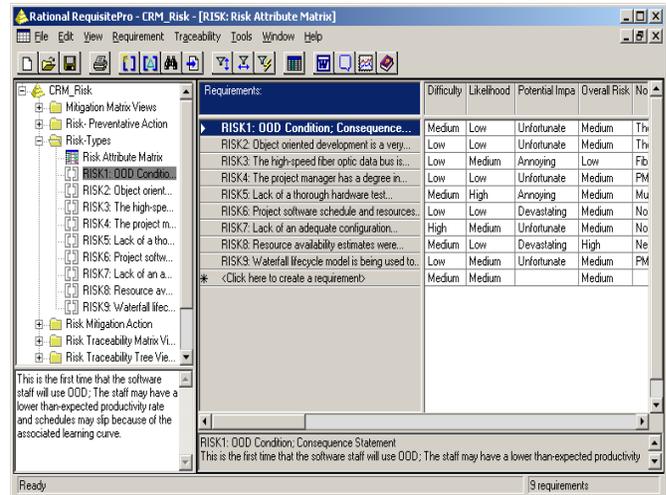


Figure 5. Risk List Attribute Matrix

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 and Risk Accept Action) would be instantiated for risks to watch and accept. Figure 6 shows 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 lifecycle 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 rela-

tionships 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.

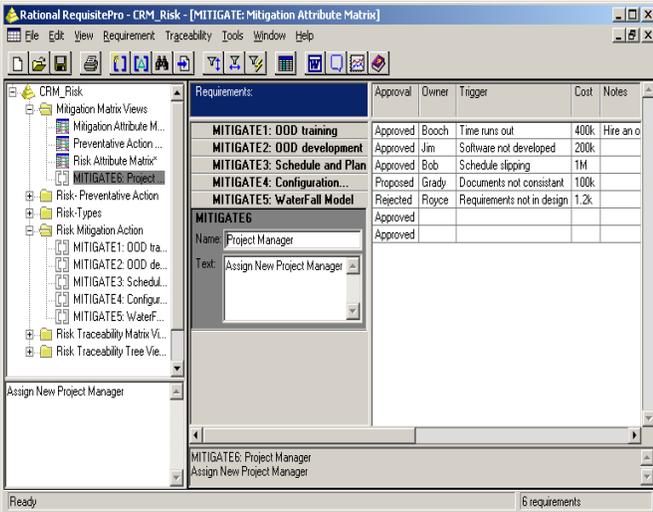


Figure 6. Risk Mitigation Attribute Matrix

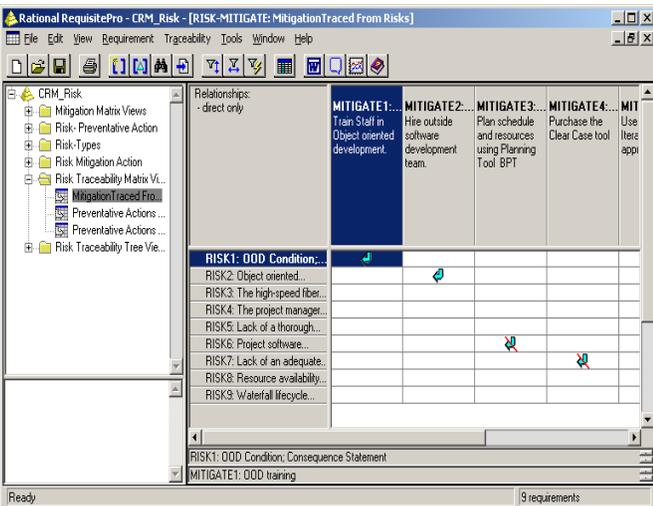


Figure 7. Mitigation Traced from Risk Matrix

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.

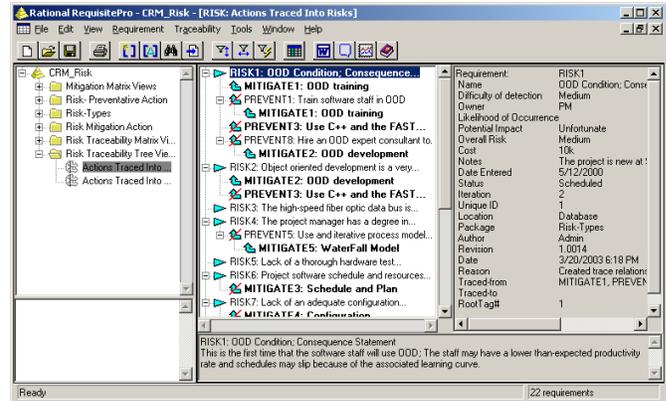


Figure 8. Risk Actions Traced Into Risks

## Conclusion

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 of 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 in order 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 analyses 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 project's integrity is maintained. RequisitePro captures the change history for each risk, which provides an audit of the project's risk evolution.

## Acknowledgements

The authors would like to acknowledge the IBM SEED Program. The SEED has helped our institute to teach advanced Software Engineering practices based on IBM's industrial products, tools and techniques. The Rational products address the entire software-development lifecycle and are used in our engineering programs.

Teaching students continuous risk management implementation is important in order 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 present each risk with the capability to be tracked, watched or mitigated throughout the project's iterative lifecycle. The IBM Rational 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

## References

- [1] IBM Rational® RequisitePro®. Retrieved February 10, 2013, from IBM Software website: <http://www-01.ibm.com/software/awdtools/reqpro/>
- [2] Van Scoy, R., (1992). *Software Development Risk: Opportunity, Not Problem* (CMU/SEI-92-TR-030). Retrieved February 10, 2013, from the Software Engineering Institute, Carnegie Mellon University website: <http://www.sei.cmu.edu/library/abstracts/reports/92tr030.cfm>
- [3] Alberts, C. J., Dorofee, A. J., Higuera, R. P., Murphy, R. L., Walker, J. A., & Williams, R. C. (1996) *Continuous Risk Management Guidebook*. Carnegie Mellon University.
- [4] Higuera, R., Gluch, D., Dorofee, A., Murphy, R., Walker, J., & Williams, R. (1994). *Introduction to Team Risk Management (Version 1.0)*, an (CMU/SEI-94-SR-001). Retrieved February 10, 2013, from the Software Engineering Institute, Carnegie Mellon University website: <http://www.sei.cmu.edu/library/abstracts/reports/94sr001.cfm>
- [5] Higuera, R., Dorofee, A., Walker, J., & Williams, R. (1994). *Team Risk Management: A New Model for Customer-Supplier Relationships* (CMU/SEI-94-SR-005). Retrieved February 10, 2013, from the Software Engineering Institute, Carnegie Mellon University website: <http://www.sei.cmu.edu/library/abstracts/reports/94sr005.cfm>
- [6] Hall, E. M. (1998). *Managing risk: Methods for Software Systems Development*. (1<sup>st</sup> ed.). Addison-Wesley.
- [7] Mirzoev, T., & Yang, B. (Spring 2010). Securing Virtualized Datacenters. *International Journal of Engineering Research and Innovation*, 2(1), 23-29.
- [8] Johnson, M., D., & Parthasarathy, A. (Spring 2010). Evaluating the Accuracy, Time, and Cost Trade-Offs Among Alternative Structural Fitness Assessment Methods. *International Journal of Engineering Research & Innovation*, 2(1), 62-68.
- [9] Smith, G., C., & Savage, M., D. (Spring 2009). Identifying Prevention Methods to Reduce CTS Incidents: Gased on Analysis of Employee Perception and By Utilizing the Strain Index. *International Journal of Engineering Research and Innovation*, 1(1), 17-20.
- [10] Anquetil, N., Kulesza, U., Mitschke, R., Moreira, A., Royer, J., Rummler, A., et al. (2010). A model-driven traceability framework for software product lines. *Software Systems Model*, 9, 427-451.
- [11] Goknil, A., Kurtev, I., Van den Berg, K., & Veldhuis, J. (2011). Semantics of trace relations in requirements models for consistency checking and inferencing. *Software Systems Model*, 10, 31-54.
- [12] Abo, R. (2010). Introduction to a Requirements Engineering Framework for Aeronautics. *Journal of Software Engineering and Applications*, 3, 894-900.
- [13] Kimbrough, R. L., & Componation, P. J. (2009). The Relationship between Organizational Culture and Enterprise Risk Management. *Engineering Management Journal*, 21(2), 18-26.
- [14] Yeo, K. T., & Ren, Y. (2009). Risk Management Capability Maturity Model for Complex Product Systems (CoPS) Projects. *Systems Engineering*, 12(4), 275-294.
- [15] Krane, H. P., Olsson, N. O. E., & Rolstadas, A. (2012). How Project Manager-Project Owner Interaction Can Work Within and Influence Project Risk Management. *Project Management Journal*, 43(2), 54-67.
- [16] Kutsch, E., & Hall, M. (2009) The Rational Choice of Not Applying Project Risk Management in Information Technology Projects. *Project Management Journal*, 40(3), 72-81.
- [17] Kutsch, E., & Hall, M. (2010) Deliberate ignorance in project risk management. *International Journal of Project Management*, 28, 245-255.
- [18] Van Epps, C. (2000). *Automating Risk Management with Rational RequisitePro*. Paper presented at the Rational Users Conference in Session RM01, Orlando, FL.
- [19] Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. *International Conference on Software Engineering - ICS*, (pp. 328-338).

## Biographies

**JAMES C. HELM** received the Bachelor of Science in Mathematics and Physics from Missouri Valley College, the Master of Science in Mathematics from the University of

---

Missouri at Rolla, and the Doctor of Philosophy degree in Industrial Engineering, Operations Research from Texas A&M University. He is an Associate Professor of System and Software Engineering in the School of Science and Computer Engineering at the University of Houston Clear Lake. Prior to this he was the Chair of the Engineering Program for eight year. Dr. Helm was an Adjunct faculty member at the University Of Houston Clear Lake for twenty-three years teaching computer science courses. He had thirty years of industrial experience with NASA contractors in the Houston area and retired from Boeing.

**IPEK BOZKURT** is an Assistant Professor in the Engineering Management and Systems Engineering Programs at University of Houston Clear Lake. She received her Ph.D. and Masters in Engineering Management from Old Dominion University. Her main research areas include personality profiles, agent-based modeling and simulation, complex systems, leadership theories, organizational behavior, online learning and education, and Engineering Management and Systems Engineering.