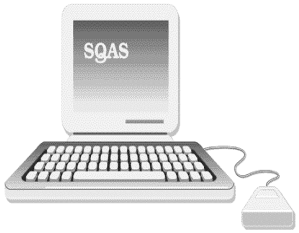


Department of Energy Quality Managers
Software Quality Assurance Subcommittee
Reference Document SQAS21.01.00 - 1999

Software Risk Management
A Practical Guide

February, 2000

Abstract

This document is a practical guide for integrating software risk management into a software project. The purpose of Risk Management is to identify, assess and control project risks. Identified risks are analyzed to determine their potential impact and likelihood of occurrence. Risk Management Plans are developed to document the project's approach to risk management, risks, and decisions made about what should be done with each risk. Risks and risk actions are then tracked to closure.

Acknowledgments

This document was prepared for the Department of Energy (DOE) by a Working Group of the DOE Quality Managers' Software Quality Assurance Subcommittee (SQAS). At the time this document was prepared, the Working Group had the following members:

Y. Faye Brown	OR, Y-12
Kathleen Canal	DOE-HQ
Ray Cullen	SR
Mike Elliott	UK AWE
Jerry Faulkner	KC
Michael Lackner	KC
Carolyn Owens	LLNL
Dave Peercy	SNL/NM
Gerald Reisz	LANL
Patricia Tempel	SNL/NM
Patty Trellue	SNL/NM

Preface

The current version of this document can be found on the Software Quality Assurance Subcommittee website at http://cio.doe.gov/sqas.

Contents

1. Introduction

This document is a practical guide for integrating risk management into a software project. The appendices for this document provide additional information and references that can provide more depth to a software risk management program.

1.1 What is risk?

Risk is the possibility of loss. It is a function of both the probability of an adverse event occurring and its impact; the impact manifests itself in a combination of financial loss, time delay, and loss of performance. A risk is the precursor to a problem; the probability that, at any given point in the software life cycle, the predicted goals cannot be achieved within available resources. Risk can not be eliminated from a software project, but it can be managed. Risk management is critical to the success of any software effort and is a strategic aspect of all software projects.

1.2 What is software risk management?

Software risk management is a software engineering practice with processes, methods, and tools for managing risks in a project. It provides a disciplined environment for proactive decision-making to assess continuously what can go wrong; determine what risks are important to deal with; and implement actions to deal with those risks. [SEIweb] Risk management planning addresses the strategy for risk management, the risk management process, and the techniques, methods, and tools to be used to support the risk management process. A template for software risk management planning is provided in Appendix F.

1.3 What is the role of management?

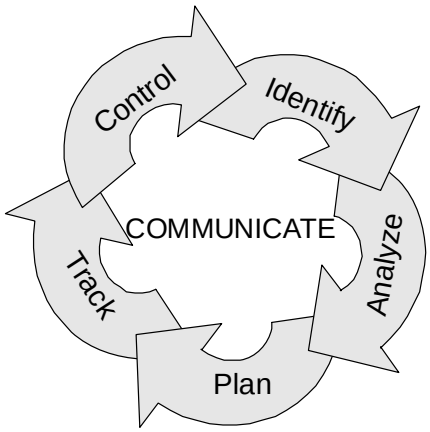
Senior management support and commitment is critical to the success of any risk management initiative. A formal risk management process requires corporate acceptance of risk as a major consideration for software management. Senior management must support project risk management activities by: (1) providing adequate personnel, budget, schedules, and other resources (e.g., tools and equipment); (2) ensuring project management receives the required training in identifying, managing, and communicating software risks; and (3) ensuring project personnel receive the required training in conducting risk management tasks. Senior management reviews risk management activities on a periodic and event-driven basis.

2. Software Risk Management Process

2.1 Process Model

There are several models available for risk management. The model recommended in this guide was developed by the Software Engineering Institute [VanScoy92] [Carr93] and is shown in Figure 2-1 below. This model may be tailored to be consistent with existing site project management processes. In all phases of a project, risks should be assessed continuously and used for decision-making.

This model identifies the fundamental risk management functions that must be taken to effectively manage risk: identify, analyze, plan, track, control, and communicate. Each of these functions is defined in more detail in section 2.2.



A continuous set of activities to identify, confront, and resolve technical risk.

Identify	Search for and locate risks before they become problems adversely affecting the project
Analyze	Process risk data into decision-making information
Plan	Translate risk information into decisions and actions (both present and future) and implement those actions
Track	Monitor the risk indicators and actions taken against risks
Control	Correct for deviations from planned risk actions
Communicate	Provide visibility and feedback data internal and external to your program on current and emerging risk activities

Figure 2-1 SEI Software Risk Management Model

2.2 Process Description

2.2.1 SRM Process: Overview

An overview of the risk management process, along with a mapping to the risk management model in Figure 2-1, is illustrated in Figure 2-2. This process description is based on the software risk management process defined by the Software Engineering Process Office [SEPO97] and the Software Engineering Institute [Carr93].

Although Figure 2-2 indicates the process steps in a sequential manner, typically there are iterations among the steps and, as shown in Figure 2-1, the process is repeated in a continuous manner, as needed.

2.2.2 SRM Process: Special Responsibilities

Personnel from the software engineering staff are selected to participate on a Software Risk Evaluation (SRE) Team. An SRE Team should have from one to five participants. The following criteria should be used in selecting participants:

- knowledge and experience in the technology areas of the effort being assessed,
- risk management experience or will receive risk management training,
- mix of people with various applicable skills (e.g. development, test, quality assurance), and
- representation for any functional areas considered critical to the project.

An individual is selected to serve as the facilitator for the risk management process. The SRE Facilitator should be someone who does not have a vested interest in the results of the process and can effectively move the process to closure. This person should meet the entrance criteria for this process; that is, they should have risk management experience or receive training.

2.2.3 SRM Process: Inputs

Inputs are those items that must be available in order to start the risk management process. Inputs will be used/transformed by the process steps into outputs. Examples of inputs are:

a. Risk Management Forms
b. Risk Management Plan Template
c. Software Project Plan
d. Software artifacts, e.g., system/software requirements
e. Organizational standards, practices, guidelines as tailored to this project

2.2.4 SRM Process: Entrance Criteria

Entrance criteria are those items that must be completed prior to starting the software risk management activities in order to have the best chance of success. Examples of entrance criteria are:

a. Management has provided adequate resources for risk management activities. Resources include funding, staffing, equipment, and tools as required.
b. Personnel have been assigned for all risk management roles and responsibilities, and they have been trained in the risk management approach to be applied to the project.
c. All identified inputs exist and are available to the SRE Team.

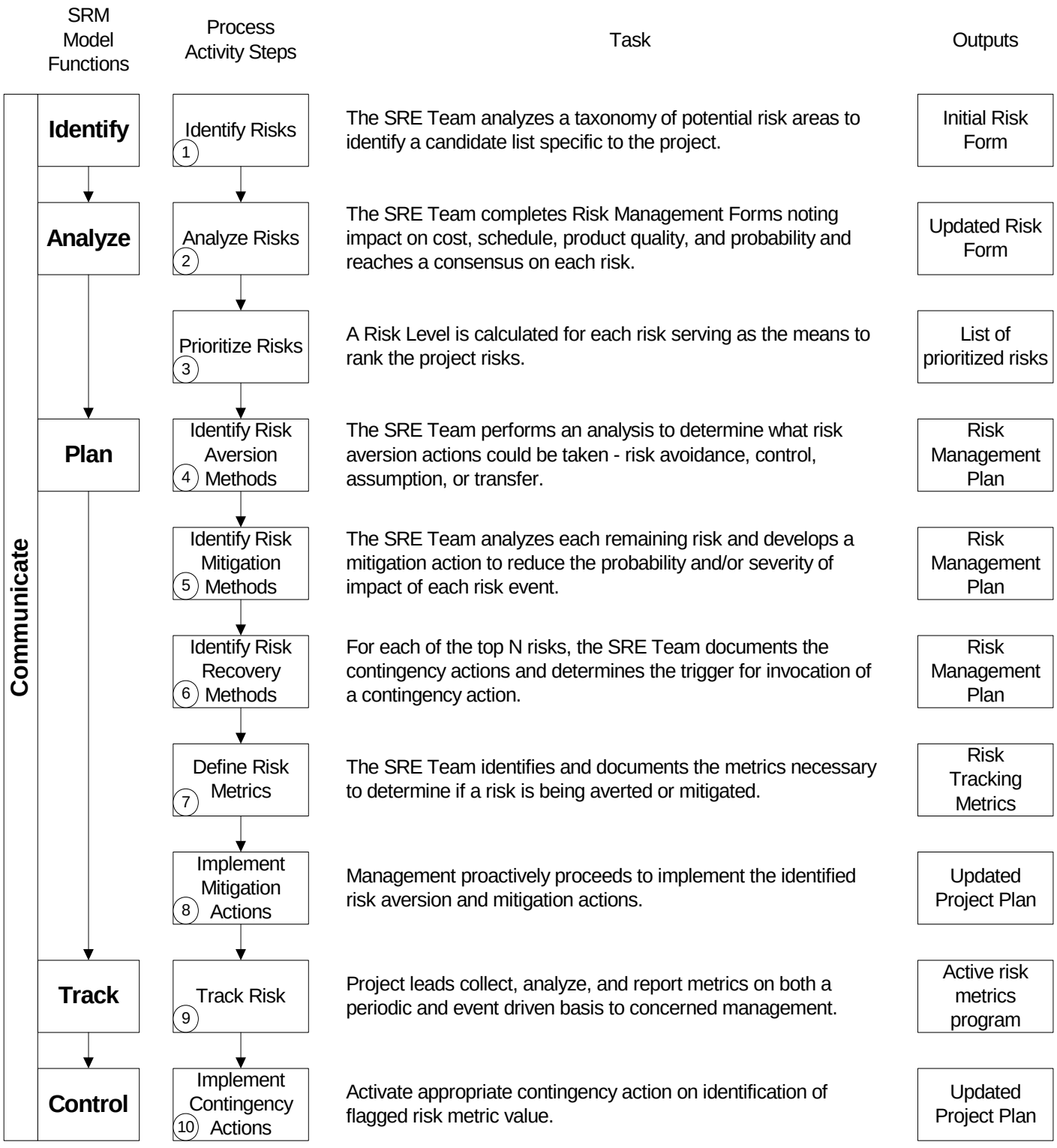


Figure 2-2 Software Risk Management Process Overview

2.2.5 SRM Process: Procedures

The risk management process consists of ten steps as described in the paragraphs that follow. Use of the activities associated with these steps constitutes an acceptable risk management approach and could be incorporated into a Risk Management Plan. The size, visibility, or consequences of the project drives the complexity of the process. The process can be tailored to be consistent with existing site project management processes.

Function 1: Identify

Before risks can be managed, they must be identified, and they must be identified before they become problems adversely affecting the project. Establishing an environment that encourages people to raise concerns and issues and conducting quality reviews throughout all phases of a project are common techniques for identifying risks.

Step 1: Identify risks

The SRE Facilitator conducts a process overview briefing and provides a taxonomy of potential risk areas to the SRE Team. The taxonomy is derived from a list of possible risks such as found in Appendix A of this document and forms the basis for analysis of potential risks to the project at hand.

The process overview briefing is held no less than one week prior to the risk identification session. At this point in the process, the identification of risk is an individual responsibility; it is important for each individual to prepare for the team risk identification session that follows. Each member of the SRE Team, using the taxonomy as a guide, identifies risks associated with the project and documents them using a Risk Management Form (see Appendix E) or another method as appropriate.

Now, the identification of risk shifts from an individual effort to an SRE Team effort. The SRE Facilitator conducts a risk identification session with the SRE Team to identify potential programmatic risks. It can be beneficial to have a person serve as a recorder for this session to ensure all identified risks are captured.

The Risk Management Forms are used to document all of the potential risks. All risks identified at the meeting are recorded, even if they do not seem to fit the taxonomy categories. During this session, only the Statement of Risk is completed on the Risk Management Forms; follow-on steps in the risk management process will fill in other information.

Function 2: Analyze

Analysis is the conversion of risk data into risk decision-making information. It includes reviewing, prioritizing, and selecting the most critical risks to address.

Step 2: Analyze risks

The SRE Team analyzes each identified risk in terms of its consequence on cost, schedule, performance, and product quality. An individual risk may impact more than one of these categories. For example, frequently changing requirements will impact all four.

The SRE Team will mark the Risk Management Form according to their determination. In addition, they determine the severity of impact, rating each risk according to the criticality levels found on the form.

Second, each SRE Team estimates the probability that each risk will occur and its time frame. Probability can be established by a percentage figure or by selection of an abstract term (e.g., High, Medium, Low).

The SRE Facilitator then conducts a series of SRE Team meetings to elaborate on the identified risks. The SRE Facilitator goes through the software development risk taxonomy, presenting the risks identified prior to the meeting and soliciting any other risks from the group. The SRE Facilitator presents possible risks to be combined and the SRE Team reaches consensus on which ones to combine and on the wording of the risks.

After consensus has been reached, the SRE Facilitator leads a discussion to establish consensus on consequence, severity, probability, and time frame for each risk. Again, it can be beneficial to have a person serve as a recorder for this session.

Step 3: Prioritize risks

Using the data from step 2, Analyze Risks, the SRE Team determines a Risk Level for each risk by mapping each risk onto a Risk Matrix, a sample of which is shown in Table 2-1. The project and risk management personnel evaluating the Risk Level for each risk can determine when appropriate mitigation action will be required. This decision making can be facilitated by the use of risk levels agreed to by the SRE Team and project management.

Where the Risk Levels are defined as:

a. **T**olerable Risk is a condition where risk is identified as having little or no effect or consequence on project objectives; the probability of occurrence is low enough to cause little or no concern.

b. **L**ow Risk is a condition where risk is identified as having minor effects on project objectives; the probability of occurrence is sufficiently low to cause only minor concern.

c. **M**edium Risk is a condition where risk is identified as one that could possibly affect project objectives, cost, or schedule. The probability of occurrence is high enough to require close control of all contributing factors.

d. **H**igh Risk is the condition where risk is identified as having a high probability of occurrence and the consequence would affect project objectives, cost, and schedule. The probability of occurrence is high enough to require close control of all contributing factors, the establishment of risk actions, and an acceptable fallback position.

e. **IN**tolerable Risk is the condition where risk is identified as having a high probability of occurrence and the consequence would have significant impact on cost, schedule, and/or performance. These risks would constitute the Top N for the project.

At the conclusion of risk prioritization, a consolidated list of risks is created, and the updated Risk Management Forms are placed under configuration management.

Probability / Severity	Frequent	Probable	Occasional	Remote	Improbable
Catastrophic	IN	IN	IN	H	M
Critical	IN	IN	H	M	L
Serious	H	H	M	L	T
Minor	M	M	L	T	T
Negligible	M	L	T	T	T
LEGEND	T = Tolerable	L = Low	M = Medium	H = High	IN = Intolerable

Probability	Description	Severity	Consequence
Frequent	Not surprised, will occur several times (Frequency per year > 1)	Catastrophic	Greater than 6 month slip in schedule; greater than 10% cost overrun; greater than 10% reduction in product functionality
Probable	Occurs repeatedly/ an event to be expected (Frequency per year 1-$10^{-1)}$	Critical	Less than 6 month slip in schedule; less than 10% cost overrun; less than 10% reduction in product functionality
Occasional	Could occur some time (Frequency per year 10^{-1} - $10^{-2)}$	Serious	Less than 3 month slip in schedule; less than 5% cost overrun; less than 5% reduction in product functionality
Remote	Unlikely though conceivable (Frequence per year 10^{-2} - 10^{-4})	Minor	Less than 1 month slip in schedule; less than 2% cost overrun; less than 2% reduction in product functionality
Improbable	So unlikely that probability is close to zero (Frequency per year 10^{-4} - 10^{-5})	Negligible	Negligible impact on program

Table 2-1 Sample Risk Matrix

Function 3: Plan

Planning turns risk information into decisions and actions for both the present and future. Planning involves developing actions to address individual risks, prioritizing risk actions and creating a Risk Management Plan. The key to risk action planning is to consider the future consequences of a decision made today. The plan for a risk can be to:

a. Avert a risk by changing the design or the process or by taking no further action thus accepting the consequences if the risk occurs
b. Mitigate the impact of the risk by reducing its Risk Level
c. Develop a contingency strategy should the risk occur

Step 4: Identify risk aversion methods

Having generated a ranked list of risks, the SRE Team performs an analysis to determine what risk aversion actions (i.e., risk avoidance, control, assumption, or transfer) could be taken or decisions could be made that would eliminate any of the identified risks. The SRE Team assesses, rates, and decides on the possible consequences of inaction and if the benefits of acting on a risk merits the expense in time and money expended. While a

conscious decision to ignore a high risk may be a creditable option, an unconscious decision to avoid risk is not.

This step could focus on constant process improvement by identifying those organizational (or project) processes that would eliminate, or substantially reduce, a given risk. Employing risk management in concert with metrics and process improvement can be used to measure, track, and improve an organization's development process.

Step 5: Identify risk mitigation methods

Risks that make it to this step are considered contingent upon external events. An SRE Team session is held to determine what actions or decisions can be made that would reduce the probability and/or severity of impact of each risk event. The SRE Team documents and details those that are practical and feasible and incorporates them into the project Risk Management Plan.

For example, if contracting represents a significant part and risk for the project, then defining management practices that include formal methods for monitoring, evaluating, and controlling contractor performance would minimize that risk. To elaborate on risk mitigation specifics for contracting, consider the following actions for the contractor technical interface established by the project manager:

a. Design reviews, development progress, configuration audits, risk analysis, trade-offs, and appropriate program requirement flowdowns should be under the cognizance of the project manager. The technical leads are responsible to the project manager for establishing the technical performance parameters that are tracked and correlated with cost and schedule data.

b. The technical leads furnish technical direction that describes how the contractors will meet their task requirements. The contractors provide plans detailing how they will adhere to contract requirements and how they will establish and report their progress.

c. The contractors are required to supply information to the level of detail necessary to assure the project manager that the cost and schedule reports provide accurate and meaningful data. At monthly contractor reviews, nontechnical and technical risk areas are highlighted.

This type of procedural specification would allow both the technical leads and the contractors to foresee problem areas and take appropriate action to avoid or minimize risk.

Step 6: Identify risk recovery methods

For each of the top N risks, the SRE team conducts a session to validate the nature of the event that would cause the invocation of a contingency action. Contingency actions for risks are document in the project Risk Management Plan along with what measurable or observable circumstances must occur to trigger the implementation of the contingency action.

A simple example could be based on variance tracking. When the actual cost of work performed exceeds one standard deviation from the planned value, a working group is formed to investigate the cause and make recommendations.

Examples of contingency actions for risks impacting product quality would include, but not be limited, to some combination of the following:

- Redesign to correct the deficiency.
- Reallocate requirements to maintain specified overall system performance.
- Define reduced performance thresholds, still exceeding minimum acceptable requirements.

Step 7: Define risk metrics

For each risk, the SRE Team determines and documents what measurable or observable event(s) can be tracked to know whether or not the risk is being averted, prevented, or minimized.

For example, if testing has been identified as a high risk function then tracking test coverage analysis at unit test time and determining error removal rates for design review, unit testing, and integration testing could serve as key metrics. Other possible test metrics could include tracking of the delta between open and closed trouble reports and the tracking of error density by trouble report priority.

In addition, the SRE Team defines and documents the risk management process measurements to be collected and analyzed on the risk process itself. Examples are provided in section 2.2.8.

Step 8: Implement mitigation/reduction actions

For each risk, the SRE Team conducts the activities necessary to implement the mitigation/reduction actions addressed in step 5 above. These activities are documented in the project Risk Management Plan for each risk reduction scenario. Examples of activities that would address the risk levels defined in step 3 are:

a. Tolerable Risk. Good system engineering practices would serve to mitigate any problems of this magnitude.

b. Low Risk. No special program emphasis is required other than normal software engineering group monitoring and control.

c. Medium Risk. This risk level would qualify as an action item at status review meetings.

d. High Risk. This risk level qualifies as an action item at status review meetings.

e. Intolerable Risk. This level requires formal control and monitoring and development of a risk contingency action. Each risk at this level has a definition of the event that would invoke the contingency action. The deviation values are set narrow enough to raise a risk flag in time to allow the project leadership time to respond yet open enough to not create excessive raised flags. The deviation values are documented with the metric description.

Function 4: Track

Tracking consists of monitoring the status of risks and the actions taken against risks to mitigate them. This is done through appropriate risk metrics and serves as the “watchdog” function of risk management.

Step 9: Track risks

Projects implement reporting procedures that raise attention flags whenever a reported metric or parameter is beyond the pre-established monitor threshold or deviation value. The method and time of collecting and reporting each metric are incorporated into the Risk Management Plan.

The RM Manager ensures the reporting procedures of the Risk Management Plan are being followed and derived metrics are computed; receives and analyzes the reports, and takes appropriate corrective actions as required.

Function 5: Control

Risk control corrects deviations from planned risk actions. Risk control is a part of project management and relies on project management processes to control risk action plans, correct for variations from plans, respond to triggering events, and improve risk management processes. Risk control activities are documented in the Risk Management Plan.

Step 10: Implement contingency actions

For each risk, if the data collected shows that the entrance criteria have been met, then:

- the need for implementation of the contingency action should to be raised to the Program Manager, and
- project management needs to provide for the direction necessary to reallocate resources required for the execution of that contingency action.

Function 6: Communicate

Communication lies at the center of the model in order to emphasize its pervasiveness and its criticality. Communication happens throughout all the functions of risk management. Without effective communication, no risk management approach can be viable. It is an integral part of all the other risk management activities. Clearly, personnel associated with a project are the most qualified to identify risk in their work on a daily basis. Project management provides a conducive environment for people to share their concerns regarding potential risks. Effective communication provides both visibility and feedback data, internal and external to your program, on current and emerging risk activities.

2.2.6 SRM Process: Exit Criteria

The results of the risk analysis and the Risk Management Plan are reviewed and the SRE Team reaches consensus on the top N risks. The final activities in a risk analysis event are a presentation of the results and a meeting with the program manager. The presentation is generally conducted as a formal presentation to all program personnel who are involved in the management of the project. Key considerations include having all participants attend the meeting and conducting the presentation such that participants will know what happened to "their" risks. An example outline for the presentation is:

a. Review of the risk assessment processes
b. Review of the complete database of risks with attributes
c. Discussion of the top N risks
d. Identification of the contingency events and a synopsis of each associated plan of action

2.2.7 SRM Process: Outputs

Outputs are those items that remain in the project files after the project is complete. Examples of outputs are:

a. Risk Management Plan

b. Risk Management Forms

2.2.8 SRM Process: Metrics

To provide input for process improvement, metrics should be collected about the software risk management process. SRM process metrics could include, but not be limited to, the following measurement data:

- Effort and funds expended in risk management activities
- Date a risk assessment is performed
- Number and breakdown of risks identified:
  - Number of new risks
  - Number of previously identified risks
- Number and breakdown of previously identified risks that are no longer considered risks:
  - Number of risks that were avoided
  - Number of risks that occurred

3. Roles and Responsibilities

The paragraphs below identify the participating individuals and/or teams of the Software Risk Management Process with their corresponding responsibilities.

3.1 Program Manager

The Program Manager has overall responsibility for several projects and provides the support and funding for risk management activities.

3.2 Project Manager

The Project Manager has overall responsibility for managing the risks associated with the development and maintenance of the system and ensuring that risk management is performed in consonance with the process described herein.

3.3 Risk Management (RM) Manager

The Project Manager may choose to be the Risk Management Manager (RM Manager) depending upon staff size and project complexity. Otherwise, this leadership responsibility is assigned as a collateral duty to one of the technical leads on the project. The RM Manager is responsible for ensuring risk management is preformed as described in the Software Risk Management Plan.

3.4 Software Risk Evaluation Team

Software engineers generally serve as members of Software Risk Evaluation (SRE) Teams. These SRE Teams analyze and document any risks associated with the tasks the project is required to perform.

3.5 SRE Facilitator

The SRE Facilitator is a person who does not have a vested interest in the results of the process and can effectively move the process to closure. This person could be the quality engineer assigned to the project.

3.6 Software Quality Assurance (SQA) Manager

The SQA Manager periodically reviews the risk management activities to ensure that the requirements of the organization’s Software Risk Management Process are being followed.

3.7 Configuration Management (CM) Manager

Depending upon project size, organizational structure, and assigned responsibilities, the CM Manager may play a role in risk tracking (measurement and analysis of risks) and reporting the status of designated risks to the RM Manager. For example, the CM Manager would be responsible for maintaining the database of Risk Management Forms (See Appendix E) and for providing summary information to the RM Manager to be used in developing project risk-related reports.

Appendix A SEI Software Risk Taxonomy

A complete description of the SEI Software Risk Taxonomy is available in [Carr93].

A. Product Engineering

1. **Requirements**
   a. Stability
   b. Completeness
   c. Clarity
   d. Validity
   e. Feasibility
   f. Precedent
   g. Scale
2. **Design**
   a. Functionality
   b. Difficulty
   c. Interfaces
   d. Performance
   e. Testability
   f. Hardware Constraints
   g. Non-Developmental Software
3. **Code and Unit Test**
   a. Feasibility
   b. Testing
   c. Coding/Implementation
4. **Integration and Test**
   a. Environment
   b. Product
   c. System
5. **Engineering Specialties**
   a. Maintainability
   b. Reliability
   c. Safety
   d. Security
   e. Human Factors
   f. Specifications

B. Development Environment

1. **Development Process**
   a. Formality
   b. Suitability
   c. Process Control
   d. Familiarity
   e. Product Control
2. **Development System**
   a. Capacity
   b. Suitability
   c. Usability
   d. Familiarity
   e. Reliability
   f. System Support
   g. Deliverability
3. **Management Process**
   a. Planning
   b. Project Organization
   c. Management Experience
   d. Program Interfaces
4. **Management Methods**
   a. Monitoring
   b. Personnel Management
   c. Quality Assurance
   d. Configuration Management
5. **Work Environment**
   a. Quality Attitude
   b. Cooperation
   c. Communication
   d. Morale

C. Program Constraints

1. **Resources**
   a. Schedule
   b. Staff
   c. Budget
   d. Facilities
2. **Contract**
   a. Type of contract
   b. Restrictions
   c. Dependencies
3. **Program Interfaces**
   a. Customer
   b. Associate Contractors
   c. Subcontractor
   d. Prime Contractor
   e. Corporate Management
   f. Vendors
   g. Politics

Appendix B SEI Risk Taxonomy Questionnaire

A complete version of the SEI Software Risk Taxonomy Questionnaire is available in [Carr93]. This Questionnaire is meant to serve as a guide in the identification of software risks.

A. Product Engineering

Technical aspects of the work to be accomplished.

1. Requirements

a. Stability

Are requirements changing even as the product is being produced?

b. Completeness

Are requirements missing or incompletely specified?

c. Clarity

Are the requirements unclear or in need of interpretation?

d. Validity

Will the requirements lead to the product the customer has in mind?

e. Feasibility

Are there requirements that are technically difficult to implement?

f. Precedent

Do requirements specify something never done before or beyond the experience of program personnel?

g. Scale

Is the system size or complexity a concern?

2. Design

a. Functionality

Are there any potential problems in designing to meet functional requirements?

b. Difficulty

Will the design and/or implementation be difficult to achieve?

c. Interfaces

Are internal interfaces (hardware and software) well defined and controlled?

d. Performances

Are there stringent response time or throughput requirements?

e. Testability

Is the product difficult or impossible to test?

f. Hardware Constraints

Does the hardware limit the ability to meet any requirements?

g. Non-Developmental Software

Are there problems with software used in the program but not developed by the program?

3. Code and Unit Test

a. Feasibility

Is the implementation of the design difficult or impossible?

b. Testing

Is the specified level and time for unit testing adequate?

c. Coding/Implementation

Are the design specifications in sufficient detail to write code: Will the design be changing while coding is being done?

4. Integration and Test

a. Environment

Is the integration and test environment adequate? Are there problems developing realistic scenarios and test data to demonstrate any requirements?

b. Product

Is the interface definition inadequate, facilities inadequate, time insufficient? Are there requirements that will be difficult to test?

c. System

Has adequate time been allocated for system integration and test? Is system integration uncoordinated? Are interface definitions or test facilities inadequate?

5. Engineering Specialties

a. Maintainability

Will the implementation be difficult to understand or maintain?

b. Reliability

Are reliability or availability requirements allocated to the software? Will they be difficult to meet?

c. Safety

Are the safety requirements infeasible and not demonstrable?

d. Security

Are there unprecedented security requirements?

e. Human Factors

Is there any difficulty in meeting the human factor requirements?

f. Specifications

Is the documentation adequate to design, implement, and test the system?

B. Development Environment

Methods, procedures, and tools in the production of the software products.

1. Development Process

a. Formality

Will the implementation be difficult to understand or maintain?

b. Suitability

Is the process suited to the development mode, e.g., spiral, prototyping? Is the development process supported by a compatible set of procedures, methods, and tools?

c. Process Control

Is the software development process enforced, monitored, and controlled using metrics?

d. Familiarity

Are the project members experienced in use of the process? Is the process understood by all project members?

e. Product Control

Are there mechanisms for controlling changes in the product?

2. Development System

a. Capacity
Are there enough workstations and processing capacity for all the staff?

b. Suitability
Does the development system support all phases, activities, functions of the program?

c. Usability
Do project personnel find the development system easy to use?

d. Familiarity
Have project personnel used the development system before?

e. Reliability
Is the system considered reliable?

f. System Support
Is there timely expert or vendor support for the system?

g. Deliverability
Are the definition and acceptance requirements defined for delivering the system to the customer?

3. Management Process

a. Planning
Is the program managed according to a plan?

b. Project Organization
Is the program organized effectively; are the roles and reporting relationships well defined?

c. Management Experience
Are the managers experienced in software development, software management, the application domain, and the development process?

d. Program Interfaces
Is there a good interface with the customer and is the customer involved in decisions regarding functionality and operation?

4. Management Method

a. Monitoring
Are management metrics defined and is development progress tracked?

b. Personnel Management
Are project personnel trained and used appropriately?

c. Quality Assurance
Are there adequate procedures and resources to assure product quality?

d. Configuration Management
Are the change procedures or version control, including installation site(s) adequate?

5. Work Environment

a. Quality Attitude
Does the project lack orientation toward quality work?

b. Cooperation
Does the project lack team spirit; does conflict resolution require management intervention?

c. Communication

Does the project lack awareness of mission or goals, communication of technical information among peers and managers?

d. Morale

Is there a non-productive, non-creative atmosphere? Does the project lack rewards or recognition for superior work?

C. Program Constraints

Methods, procedures, and tools in the production of the software products.

1. Resources

a. Schedule

Is the project schedule inadequate or unstable?

b. Staff

Is the staff inexperienced, lack domain knowledge, lack skills, or is not inadequately sized?

c. Budget

Is the funding insufficient or unstable?

d. Facilities

Are the facilities inadequate for building and delivering the product?

2. Contract

a. Type of contract

Is the contract type a source of risk to the project?

b. Restrictions

Does the contract include any inappropriate restrictions?

c. Dependencies

Does the program have any critical dependencies on outside products or services?

3. Program Interfaces

a. Customer

Are there any customer problems such as a lengthy document-approval cycle, poor communication, or inadequate domain expertise?

b. Associate Contractors

Are there any problems with associate contractors such as inadequately defined or unstable interfaces, poor communication, or lack of cooperation?

c. Subcontractor

Is the program dependent on subcontractors for any critical areas?

d. Prime Contractor

Is the program facing difficulties with its prime contractor?

e. Corporate Management

Is there a lack of support or micro management from upper management?

f. Vendors

Are vendors unresponsive to program needs?

g. Politics

Are politics causing a problem for the program?

Appendix C Software Risk Factors

This following list of Software Risk Factors was taken from Chapter 6 of the Guidelines for Successful Acquisition and Management of Software Intensive Systems [STSC].

A risk is the precursor to a problem. It is the probability that, at any given point in the system life cycle, its predicted goals (either operational or logistical) cannot be achieved with available resources. Trying to totally eliminate risk is a futile endeavor – however, managing risk is something you can and must do.

As a program manager, you risk failure in three ways and combinations thereof:

- The product does not meet performance requirements (operationally or logistically),
- Actual costs are higher than budgeted, and
- Delivery of the product is too late

Software risk factors that impact a product's performance, cost, and schedule can be further segmented into five risk areas. However, any given risk may have an impact in more than one area. The five risk areas are:

- Technical risk (performance related)
- Supportability risk (performance related)
- Programmatic risk (environment related)
- Cost risk
- Schedule risk

Common risk factors. The Software Program Manager's Network [Evans94] identifies common threads among troubled software programs. These conclusions, based on risk assessments performed on 30 software programs since 1988, include:

- Management. Management is inconsistent, inappropriately applied or not applied at all. Management is reactionary.
- Predictable risks ignored. When a problem arises, program personnel identify it but they often ignore it because it is too much trouble to deal with it.
- Disciplines not uniformly applied. Configuration management, product assurance, and technical disciplines are not uniformly applied. Organizations throw away standards as they go through the software development life cycle in order to buy more schedule time or save on cost. This creates a rolling wave of disaster in the long-term.
- Poor training. Managers do not know how to perform a specific task, do not understand how to cost, schedule or track software development; do not know how to do the technical job, or to plan. No training or resources are available to them.
- Fallacy of easy solutions. Software programs often get in trouble when generic solutions are applied to specific problems. One example, programs fail to scale the work to resources.
- Inadequate work plans. Critical constraints and work plans include schedules, budgets, work allocation, and limited, time-sensitive resources. Inadequate schedules, or schedules not enforced, is often the problem.

- Schedule reality. The schedule plan must be realistic. If schedule slips, the impact on delivery must be assessed. Too often short cuts are taken to come up with a success-oriented schedule and avoid announcing an end-date slip.
- Delivery focus. Many programs focus on schedule and progress not on the delivery. Successful programs focus on the incremental completion of an event.
- Constraints. Successful programs use reasonable metrics to status and analyze the program, assess product quality and process effectiveness, and to project the potential for success.
- Customer responsibility. The customer should provide standards and documentation for the software development phase in which they are interested.
- Methods and tool selection. Tools selected are inappropriate for the job. Program staff has too little or no experience with the methods used. Process is not integrated through configuration management – no effective information flow within the program is established.

Appendix D Risk Aversion Techniques

The following list was taken from Chapter 6 of the Guidelines for Successful Acquisition and Management of Software Intensive Systems [STSC].

D.1 Risk Avoidance

Risk can be avoided by choosing an alternative approach with lower risk. This conscious choice avoids the potentially higher risk; however it really results in reduction in risk – not complete risk elimination. While a conscious decision to ignore (or assume) a high risk may be a creditable option, an unconscious decision to avoid risk is not. The possible consequences of inaction must be assessed, rated, and decided on. Whether the benefits of acting on a risk merit the expense in time and money expended is decided. All risk-handling actions are documented with supporting rationale. Risk management is employed in concert with metrics and process improvements to measure, track, and improve the project management process.

D.2 Risk Control

Risks can be controlled by continually monitoring and correcting risky conditions. This involves the use of reviews, inspections, risk reduction milestones, development of contingency, and similar management techniques. Controlling risk involves developing a risk reduction plan, then tracking to that plan.

D.3 Risk Assumption

Risk can be assumed by making a conscious decision to accept the consequences should the event occur. Some amount of risk assumption always occurs in software programs. The SRE team must determine the appropriate level of risk that can be assumed in each situation as it occurs.

D.4 Risk Transference

Risk can be transferred when there is an opportunity to reduce risk by sharing it. Risk can be transferred upward within the organization or to another organization.

Appendix E Risk Management Forms

The Risk Accounting Form and the Risk Information Form are two examples of ways to document risks. A consistent format or content for risk statements helps everyone understand the real risk.

E.1 Example 1: Risk Accounting Form

Risk Accounting Form[1]		
Identified by:		**Date:**
		ID #: *CM Tracking #*
Statement of Risk (with context):		
	Consequence: (**C**ost , **S**chedule, **P**erformance, **Q**uality)	Risk Magnitude Rm
	Severity: (**C**ritical**, S**erious**, M**oderate**, M**inor)	
	Probability of occurrence? (**H**igh, **M**edium , **L**ow, **%**)	
	Timeframe of risk? (**N**ear-term**, F**ar-term)	
Mitigation Strategy: Different strategies to mitigate this risk. When it must be mitigated.		
Contingency Action and Trigger:		
Risk Grouping: *Other risks (by ID) that will impact this risk or are impacted by this risk*		

[1] This form is a modification of the form provided in[Dorofee96].

E.2 Example 2: Risk Information Sheet

<table>
<tr><td colspan="2">ID:</td><td colspan="2">Risk Information Sheet[2]</td><td>Identified:</td></tr>
<tr><td>Priority:</td><td></td><td colspan="3" rowspan="3">Statement of Risk:</td></tr>
<tr><td>Probability:</td><td></td></tr>
<tr><td>Impact:</td><td></td></tr>
<tr><td colspan="2">Timeframe:</td><td>Origin:</td><td>Class:</td><td>Assigned To:</td></tr>
<tr><td colspan="5">Context:</td></tr>
<tr><td colspan="5">Mitigation Strategy:</td></tr>
<tr><td colspan="5">Contingency Action and Trigger:</td></tr>
<tr><td colspan="5">Status: Status Date:</td></tr>
<tr><td colspan="2">Approval:</td><td>Closing Date:</td><td colspan="2">Closing Rationale:</td></tr>
</table>

[2] This form is from [Dorofee96].

Appendix F Risk Management Plan Template

A Risk Management Plan is a controlling document that incorporates the goals, strategy, and methods for performing risk management on a project. The plan describes all aspects of the risk identification, estimation, evaluation, and control processes.

The purpose of developing such a plan is to determine the approach for performing risk management cost-effectively on the project. The plan should be long enough to convey this information to all project participants and be commensurate with the level of consequences for failure of the software product. The plan may be a separate document or part of a larger plan (e.g., project management plan or software development plan). Appendix F contains a Risk Management Plan template that can be used as is or tailored to fit the requirements of a specific project.

The Risk Management Plan should be documented in an easily understood format. It should be approved through the appropriate management levels to promote support and commitment, then be distributed to the project team to provide common goals, strategy, and methods for performing risk management.

Risk Management Plan Template

1. Goals

Explain why the project needs risk management (purpose), what the project expects to gain from the use of risk management (objectives), and how the Risk Management Plan responds to risk management requirements (scope). Goals should provide direction and focus for the project team members.

1.1 Purpose

Describe what the project team hopes to achieve by following the Risk Management Plan. The statement of purpose provides the motivation and expectation for risk management results.

1.2 Objectives

Describe the specific actions that will help to achieve the goals. The objectives may be listed in order of priority and may be written as quantitative targets, such as "100 percent award fee" or "zero defects."

1.3 Scope

Provide an overview of the major sections of the Risk Management Plan. A few sentences for each major section are sufficient to provide a synopsis of the contents of the plan.

2. Strategy

Describe the philosophy and guiding principles of the organization's risk management process, as well as how projects are organized to manage risk. The risk management strategy determines the manner in which the project team will implement the plan.

2.1 Policy

The Risk Management Plan should comply with an organization's risk management policy. The policy can be referenced without duplicating its contents in the Risk Management Plan. If an organizational policy does not exist, an industry standard can be identified as the policy.

2.2 Approach

Define the principles by which the project will manage risks. Projects may share a similar process, but have a diverse approach, which yields different results. A successful risk management approach is proactive, integrated, systematic, and disciplined.

2.3 Project Roles

Describe how responsibility and authority for managing the project risks will be delegated. Identify the Software Risk Evaluation Team and clearly define their roles and responsibilities, how the team fits into the total program, and the person to be assigned the Risk Management Manager. Each role should identify the internal and external interfaces where known risk must be communicated.

3. Process

The risk management process is a systematic and structured way to manage risks that include the activities and mechanisms used to transform project knowledge into decision-making information. This section should contain a description of the risk management process that will be used for the project. A standard risk management process or an existing organizational process can be tailored for each project.

4. Verification

Risk management verification is the method used to ensure that project practices adhere to the documented Risk Management Plan. In this section, describe the verification methods that will be used on the project. Reviews and audits are common methods used for verification.

Review Criteria - The purpose of a review is to understand the activities, agents, and artifacts of the Risk Management Plan to prepare for a compliance audit. Specify the review criteria that set the expectations for compliance.

Audit Procedure - An audit procedure verifies whether planned activities are conducted and participants are trained, and whether there is adherence to the Risk Management Plan.

Audit Report - The audit report is generated to summarize implementation performance and detail any discrepancies against the plan. The report shows if requirements have been achieved and the nature of any nonconformance.

5. Mechanisms

The risk management mechanisms are the techniques and tools used during the risk management process to transform inputs into outputs. Mechanisms such as taxonomys, forms, and databases can be included in the Risk Management Plan to help the project team visualize the organization of risk information. In this section, describe the methods and tools the project team will use to execute the risk management process.

Risk Taxonomy - A risk taxonomy organizes areas of concern into categories to understand the nature of the risk. Risk taxonomys help us to completely identify risks in a given area.[3]

Risk Management Form - A risk management form documents risk information essential for managing risk. The form is used to record risk information systematically and to track it to closure.[4]

Risk Database Structure - The risk database structure shows the organization of identified risks and associated information. It organizes risk information to support queries, status tracking, and report generation.

6. Risk Resolution

Risk resolution is the process that is implemented after the general risk analysis process has been conducted. Risk resolution occurs at four levels of increasing severity as follows:

Aversion/Avoidance Alternatives

Mitigation/Reduction Techniques

Contingency Planning

Crisis Management

[4] See Appendix A of this guide.

[5] See Appendix E of this guide.

6.1 Risk Aversion Alternatives

For each identified risk, risk aversion determines what actions (i.e., risk avoidance, control, assumption, or transfer) will be taken for its mitigation. The risk aversion section of the plan addresses the following information [Charette89][5]

- *A breakdown of all program risk areas and representative risk factors in each area*
- *Identification of priority risk items with a ranking of importance in relation to program objectives*
- *Tracking, decision, feedback points, and monthly reports, including when the priority risk item list and plans are to be updated*
- *Identified risk aversion alternatives and their costs*
- *Integration strategy for individual risk aversion plans (with attention to combining action plans for more than one risk item)*
- *Assignment of resources needed to implement the risk aversion strategy which includes costs, schedule and technical considerations*
- *6.2 Risk Mitigation Techniques*
- *Recommended mitigation strategy for each risk item including action plans for each (risk items requiring no action should also be noted)*
- *Mitigation implementation start date, schedule, and key milestones*
- *Criteria for success (i.e., when will the risk be considered mitigated) and the monitoring approach to be used*

6.3 Contingency Planning

Contingency planning addresses those risks that require monitoring for some future response should the need arise. Develop contingency plans for each prioritized risk that exceeds the predefined threshold. With proper contingency planning, the project team establishes a drop-dead date, and sets aside funds and personnel to implement the plan, while still staying within the preplanned project delivery schedule. [Fairley94]

- *Contingency planning involves the following*
- *Specifying the nature of the potential risk*
- *Considering alternative approaches*
- *Specifying constraints*
- *Analyzing alternatives*
- *Selecting an approach*

6.4 Crisis Management

Crisis management is used if the contingency plan fails to resolve the risk within a specified time. A crisis occurs when your contingency plan fails to resolve an unforeseen event. A crisis is an overall show-stopper! You must act quickly to manage a major unforeseen negative event. All program effort and resources must be focused on resolving the crisis situation. Once in crisis, you must muster your forces, go on the offensive, and attack! If you do not attack this type of risk, it will attack you and win.

[6] See Appendix D of this guide.

Before a crisis materialized, the project team may be able to define some elements of crisis management, such as the responsible parties and drop-dead date, but the team may be hard pressed to plan the exact details until the crisis occurs. [Fairley94] explains what you must do in such a situation:

- *Announce and publicize the problem*
- *Assign responsibilities and authorities*
- *Update status frequently*
- *Relax resource constraints (fly in experts, bring on emergency personnel, provide meals and sleeping facilities to keep people onsite until the crisis is resolved, etc.)*
- *Have program personnel operate in burnout mode*
- *Establish a drop-dead date*
- *Clear out unessential personnel*

A crisis recovery procedure should be invoked when the crisis is over, whether it had a positive or negative outcome. The project team examines what went wrong, evaluates how the budget and schedule have been affected, and rewards key crisis management personnel. During crisis recovery, perform the following:

- *Conduct a crisis postmortem*
- *Fix any systematic problems that caused the crisis*
- *Document lessons learned*
- *Recalculate cost and time to complete the program or project*
- *Rebaseline*
- *Update the project schedule and cost estimates to reflect these new projections.*

Appendix G References and Bibliography

[AWE70197] United Kingdom Atomic Weapons Establishment, Hunting-Brae Ltd., Workplace Risk Assessment, CSP 701, Issue 2, Feb. 1997.

[Boehm91] Boehm, Barry W., "Software Risk Management: Principles and Practices," *IEEE Software*, January 1991.

[Carr93] Carr, Marvin et al, *Taxonomy-Based Risk Identification*, CMU/SEI-93-TR-006. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, June 1993.

[Charette89] Charette, Robert N., *Software Engineering Risk Analysis and Management*. New York: McGraw-Hill, 1989.

[DOECIO] Department of Energy, Chief Information Office Software Management Program website: http://www.cio.doe.gov/smp

[Dorofee96] Dorofee, Audrey et.al., *Continuous Risk Management Guidebook*. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1996.

[Dorofee97] Dorofee, Audrey et.al., *Risk Management in Practice*, Crosstalk, April 1997.

[Evans94] Evans, *Thread of Failure: Project Trends That Impact Success and Productivity*, NewFocus, Number 203, Software Program Managers Network, Naval Information System Management Center, March 1994.

[Fairley94] Fairley, Richard, "Risk Management for Software Projects," *IEEE Software*, May 1994.

[Hall98] Hall, Elaine M., *Managing Risk: Methods for Software Systems Development*, ISBN 0-201-25592-8. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1998.

[Higuera94] Higuera, Ronald et. Al., *Team Risk Management: A New Model for Customer-Supplier Relationships*, CMU/SEI-94-SR-005. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, July 1994.

[ISO1502698] ISO/IEC 15026, *Information technology -- System and software integrity levels*, International Organization for Standardization, 1998.

[Pressman92] Pressman, Roger S., *Software Engineering: A Practitioners Approach*, Third Edition, 1992.

[SEIweb] Software Engineering Institute Web site: http://www.sei.cmu.edu

[SEPO97] Software Engineering Process Office (SEPO), *Risk Management Process V2.0*, Naval Command, Control and Ocean Surveillance Center, Research, Development, Test and Evaluation Division, May 1997.

[Sisti94] Sisti, Francis J. and Sujoe, Joseph, *Software Risk Evaluation Method Version 1.0*, CMU/SEI-94-TR-019. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1994.

[STSC] Software Technology Support Center. *Guidelines for Successful Acquisition and Management of Software Intensive Systems version 2.0* : Chapter 6, Risk Management. Hill AFB, Utah, U.S. Air Force, 1996.

[STSCWeb] Software Technology Support Center website: http://www.stsc.hill.af.mil/stscdocs.html

[USAF88] Air Force Systems Command and Air Force Logistics Command, Acquisition Management Software Risk Abatement, AFSC/AFLC pamphlet 800-45, Sept. 1988.

[VanScoy92] Van Scoy, Roger L. *Software Development Risk: Opportunity, Not Problem*, CMU/SEI-92-TR-030, ADA258743. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1992.

Appendix H Acronyms

CM	Configuration Management
CMM	Capability Maturity Model
RM	Risk Management
SEI	Software Engineering Institute
SQA	Software Quality Assurance
SRE	Software Risk Evaluation
SRM	Software Risk Management