



# Risk Management Guidelines

---

**Version No:** 1.0

**Date:** May 03, 2017

## Revision History

Version	Date	Prepared by / Modified by	Change Summary	Approved By	Approved On
1.0	May 03, 2017	Sridhar	First Draft Made	Anitha T G	May 18, 2017

## Table of Contents

<b>1. Risk .....</b>	<b>4</b>
<b>2. Risk Identification Sources .....</b>	<b>4</b>
<b>3. Risk Categories .....</b>	<b>4</b>
<b>4. Risk Identification Questions .....</b>	<b>5</b>
<b>5. Risk Impact Evaluation .....</b>	<b>12</b>
5.1. Risk Documentation Guidelines .....	13
5.2. Risk Response Strategy .....	13

## 1. Risk

**Risk** is defined as the effect of uncertainty on objectives. Risk can be characterized as a negative uncertainty, commonly referred to as a threat, as well as a positive uncertainty, commonly referred to as an opportunity.

## 2. Risk Identification Sources

Risk sources are the drivers that cause risks in a project or organization. There are many sources of risks, both internal and external to a project. Risk sources will identify where risks can originate from.

Below are the Risk Sources:

- New/Rare Technologies
- Market Demand
- Unavailability of skills
- Environmental Changes
- Newly Created Processes
- Observations from Audit
- Uncertain requirements
- Difficult to Estimated
- Complex Requirements
- Unrealistic Expectations
- Inadequate subcontractor capability
- Communication Issues
- Regulatory Constraints
- Inadequate tools
- Unavailability of Resources

## 3. Risk Categories

Categories are the bins used for collecting and organizing the risks. Identifying these categories will help in future consolidation and analysis of risks and come out with the actions. A risk category is the characterization of risk source. Below are the risk categories

- Life cycle phases (Requirements, Design, Coding, Continuous integration, testing etc.,)
- Processes / procedures
- Project Management
- Engineering / Technical (Safety, Performance, Functionality etc.,)
- Business

- Environmental

## 4. Risk Identification Questions

Below are the list of questions that the project manager can use while identifying the project risks and this will help him in coming out with all the potential risks for the project considering all the potential sources and categories as identified above.

### Life Cycle Phases:

- **Requirements**
  - Stability
    - Are the requirements stable? If not, what is the impact on the system?
    - Are the external interfaces changing?
  - Completeness
    - Are there any TBD's in the user stories?
    - Does the customer have any unwritten expectations?
  - Clarity
    - Are the requirements clear and understood?
    - Are there any ambiguities in the user stories?
  - Validity
    - Do the customer and the project has the same understanding of the requirements?
    - How the requirements are being validated? (prototype, signoff etc.,)
  - Feasibility
    - Are the requirements feasible?
    - Do you have the tools and technologies to implement the requirements?
  - Precedent
    - Do requirements specify something never done before?
  - Scale
    - Do requirements specify a product larger, more complex which was not done earlier?
- **Design**
  - Interfaces
    - Are the internal interfaces hardware and software well defined and controlled?
  - Performance
    - Are there stringent response time or throughput requirements?
  - Testability
    - Is the product difficult or impossible to test?
  - Hardware Constraints
    - Are there tight constraints on the target hardware?

- Does the hardware limit your ability to meet any requirements?
- Non-Developmental Software
  - Are there tight constraints on the target hardware?
  - Are you reusing or re-engineering software not developed on the program?
  - Are there any problems with using COTS commercial off-the-shelf software?
  - Requires a large share of memory or database storage
  - Difficult to interface with application software
  - Do you foresee any problem with integrating COTS software updates or revisions?
- **Code and Unit Test**
  - Feasibility
    - Is the implementation of the design difficult or impossible?
  - Coding/Implementation
    - Are there any problems with coding and implementation?
  - External storage
    - Is the language suitable for producing the software on this program?
    - If multiple languages used, there interface compatibility between the code produced by the different compilers?
    - Is the development computer the same as the target computer?
  - Developmental hardware
    - Are the hardware specifications adequate to code the software?
    - Are the hardware specifications changing while the code is being written?
  - Unit Testing
    - Are the specified level and time for unit testing adequate?
- **Integration and Test**
  - Environment
    - Is the integration and test environment adequate?
    - Will there be sufficient hardware to do adequate integration and testing?
    - Is there any problem with developing realistic scenarios and test data to demonstrate any requirements?
    - Are you able to verify performance in your facility?
    - Does hardware and software instrumentation facilitate testing?
  - Product
    - Is the interface definition adequate, facilities inadequate, time insufficient?
    - Will the target hardware be available when needed?
    - Have acceptance criteria been agreed to for all requirements?
    - Are the external interfaces defined, documented, and base lined?
    - Are there any requirements that will be difficult to test?

- Has sufficient product integration been specified?
- System
  - System integration uncoordinated, poor interface definition, or inadequate facilities?
  - Has sufficient system integration been specified?
  - Has adequate time been allocated for system integration and test?
  - Is all contractors' part of the integration team?
  - Will the products be integrated into an existing system?
  - Will system integration occur on customer site?

## Engineering

- **Maintainability**
  - Will the implementation be difficult to understand or maintain?
  - Does the architecture, design, or code create any maintenance difficulties?
  - Are the maintenance people involved early in the design?
  - Is the product documentation adequate for maintenance by an outside organization?
- **Reliability**
  - Are the reliability or availability requirements difficult to meet?
- **Safety**
  - Are the safety requirements infeasible and not demonstrable?
  - Do you see any difficulty in meeting the safety requirements?
  - Will it be difficult to verify satisfaction of safety requirements?
- **Security**
  - Are the safety requirements more stringent than the current state of the practice or program experience?
  - Are there unprecedented or state-of-the-art security requirements?
- **Human Factors**
  - Will the system will be difficult to use because of poor human interface definition?
- **prototyping**
  - Are you experienced in this type of development?
- **Specifications**
  - Is the documentation adequate to design, implement, and test the system?
  - Are the hardware specifications adequate to design and implement the software?
  - Are the external interface requirements well specified?
  - Are the test specifications adequate to fully test the system?

## Development Process

- **Formality**
  - Will the implementation be difficult to understand or maintain?
  - Are there formal, controlled plans for all development activities?

- **Suitability**
  - Is the process suited to the development model, e.g., v-process, maintenance, conversion?
  - Is the development process adequate for this product?
  - Is the development process supported by a compatible set of procedures, methods, and tools?
- **Process Control**
  - Is the software development process enforced, monitored, and controlled using metrics? Are distributed development sites coordinated?
  - Can you measure whether the development process is meeting your productivity and quality goals?
  - Is there adequate coordination among distributed development sites?
- **Familiarity**
  - Are the project members experienced in use of the process? Is the process understood by all staff members?
- **Product Control**
  - Are there mechanisms for controlling changes in the product?
  - Is the traceability mechanism used in evaluating requirement change impact analyses?
  - Is there a formal change control process?
  - Are changes at any level mapped up to the system level and down through the test level?
  - Is there adequate analysis when new requirements are added to the system?
  - Do you have a way to track interfaces?
  - Are the test plans and procedures updated as part of the change process?

## Development System

- **Capacity**
  - Is there sufficient work station processing power, memory, or storage capacity?
  - Are there enough workstations and processing capacity for all staff?
  - Is there sufficient capacity for overlapping phases, such as coding, integration and test?
- **Suitability**
  - Does the development system support all phases, activities, and functions?
- **Usability**
  - How easy is the development system to use?
- **Familiarity**
  - Is there little prior company or project member experience with the development system?
  - Is team got familiarity with the tools being used in the project development?
- **Reliability**
  - Does the system suffer from software bugs, down-time, insufficient built-in back-up?
- **System Support**



- Is there timely expert or vendor support for the system?
- Are the people trained in use of the development tools?
- Do you have access to experts in use of the system?
- Do the vendors respond to problems rapidly?
- **Deliverability**
  - Are the definition and acceptance requirements defined for delivering the development system to the customer not budgeted?
  - Have adequate budget, schedule, and resources been allocated for this deliverable?

### Management Process

- **Planning**
  - Is the planning timely, technical leads included, contingency planning done?
  - Do people routinely get pulled away to fight fires?
  - Is re-planning done when disruptions occur?
  - Are there contingency plans for known risks?
  - Are long-term issues being adequately addressed?
- **Project Organization**
  - Are the roles and reporting relationships clear?
- **Management Experience**
  - Are the managers experienced in software development, software management, the application domain, the development process, or on large programs?
- **Program Interfaces**
  - Is there poor interface with customer, other contractors, senior and/or peer managers?
  - Are conflicts with the customer documented and resolved in a timely manner?
  - Does management work to ensure that all customer factions are represented in decisions regarding functionality and operation?
  - Is it good politics to present an optimistic picture to the customer or senior management?

### Management Methods

- **Monitoring**
  - Are management metrics defined and development progress tracked?
  - Do people get a response to their status reports?
  - Do you track progress versus plan?
- **Personnel Management**
  - Are project personnel trained and used appropriately?
  - Do people get assigned to the program who does not match the experience profile for your work area?
  - Do people feel it's important to keep to the plan?
  - Does management consult with people before decisions that affect their work?

- **Quality Assurance**
  - Are there adequate procedures and resources to assure product quality?
  - Is the software quality assurance function adequately staffed on this program?
  - Do you have defined mechanisms for assuring quality?
- **Configuration Management**
  - Are the change procedures or version control, including installation sites adequate?
  - Is the configuration management function adequately staffed?
  - Does the configuration management system synchronize your work with site changes?
  - Does the configuration management system provide for multiple sites?

### Work Environment

- **Quality Attitude**
  - Is there a lack of orientation toward quality work?
- **Cooperation**
  - Is there a lack of team spirit? Does conflict resolution require management intervention?
  - Do people work effectively toward common goals?
  - Is management intervention sometimes required to get people working together?
- **Communication**
  - Is there poor awareness of mission or goals, poor communication of technical information among peers and managers?
  - Are the managers receptive to communication from program staff?
  - Do the program members get timely notification of events that may affect their work?
- **Morale**
  - Is there a non-productive, non-creative atmosphere? Do people feel that there is no recognition or reward for superior work?
  - How is morale on the program?

### Resources

- **Schedule**
  - Is the schedule inadequate or unstable?
  - Is the estimation method based on historical data?
  - Is there anything for which adequate schedule was not planned?
  - Are there external dependencies which are likely to impact the schedule?
- **Staff**
  - Is the staff inexperienced, lacking domain knowledge, lacking skills, or understaffed?
  - Are there any areas in which the required technical skills are lacking?
  - Do you have adequate personnel to staff the program?
  - Is the staffing stable?

- Do you have access to the right people when you need them?
- **Budget**
  - Is the funding insufficient or unstable?
  - Do budget changes accompany requirement changes?
  - Is this a standard part of the change control process?
- **Facilities**
  - Are the facilities adequate for building and delivering the product?
  - Is the integration environment adequate?

**Contract**

- **Type of Contract**
  - Is the contract type a source of risk to the program?
  - Is the required documentation burdensome?
- **Restrictions**
  - Does the contract cause any restrictions?
  - Are there problems with data rights?
- **Dependencies**
  - Does the program have any dependencies on outside products or services?

**Customer**

- Are there any customer problems such as: lengthy document-approval cycle, poor communication, and inadequate domain expertise?
- Is the customer approval cycle timely?
- Do you ever proceed before receiving customer approval?
- Does the customer understand the technical aspects of the system?
- Does the customer understand software?
- Does the customer interfere with process or people?
- Does management work with the customer to reach mutually agreeable decisions in a timely manner?
- Are all customer factions involved in reaching agreements?
- Does management present a realistic or optimistic picture to the customer?

**Associate Contractors**

- Are there any problems with associate contractors such as inadequately defined or unstable interfaces, poor communication, or lack of cooperation?
- Is there any problem with getting schedules or interface data from associate contractors?

**Subcontractors**

- Is the program dependent on subcontractors for any critical areas?
- Are there any ambiguities in subcontractor task definitions?

- Is the subcontractor reporting and monitoring procedure different from the program's reporting requirements?
- Is subcontractor administration and technical management done by a separate organization?
- Are you highly dependent on subcontractor expertise in any areas?
- Is subcontractor knowledge being transferred to the company?
- Is there any problem with getting schedules or interface data from subcontractors?

**Prime contractor**

- Is the program facing difficulties with its Prime contractor?
- Are your task definitions from the Prime ambiguous?
- Do you interface with two separate prime organizations for administration and technical management?
- Are you highly dependent on the Prime for expertise in any areas?
- Is there any problem with getting schedules or interface data from the Prime?

**Corporate Management**

- Is there a lack of support or micro management from upper management?
- Does corporate management give you timely support in solving your problems?

**Vendors**

- Are vendors affected groups responsive to program needs?

**Politics**

- Are politics causing a problem for the program?
- Are politics affecting the program?
- Are politics affecting technical decisions?

## 5. Risk Impact Evaluation

Impact is the potential effect that the risk can have on the project objectives like:

- Cost
- Delivery Time
- Quality
- Functionality
- Customer Satisfaction

For each risk identified, the impact and the probability of occurrence are identified. The impact can be classified as low, medium or high based on the following table.

Impact	Schedule	Cost	Quality	Functionality	Impact Weight
Low	<5%	<5%	<2% increase in defect	<2% of the	2

Impact	Schedule	Cost	Quality	Functionality	Impact Weight
			rate	functionality cannot be delivered	
Medium	5% to 10%	5% to 10%	2% to 5% increase in defect rate	2% to 5% of the functionality cannot be delivered	5
High	>10%	>10%	>5% increase in defect rate	>5% of the functionality cannot be delivered	10

The probability of impact can be classified as improbable, likely or very likely based on the following table or other schemas.

Occurrence	Probability	Weight
Improbable	<5% chance of occurrence	2
Likely	5% to 50% chance of occurrence	5
Very likely	>50% chance of occurrence	10

The priority of the risk is decided based on the RPN (Risk Priority Number) which is calculated as below

$RPN = (\text{Impact on schedule} + \text{Impact on Cost} + \text{Impact on Quality} + \text{Impact on functionality}) * \text{Probability of Occurrence}$

## 5.1. Risk Documentation Guidelines

Use below template to document the identified risks to ensure that it is very clear and concise.

[Event that has an effect on objectives] **caused by** [cause/s] **resulting in** [consequence/s].

OR

[Event that has an effect on objectives] **caused by** [cause/s]. This may **result in** [consequence/s].

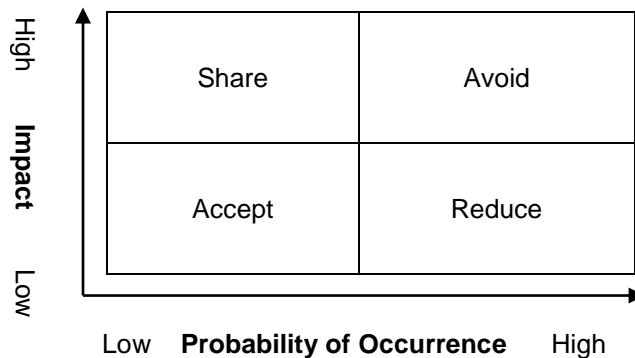
- An effect is a deviation from the expected
- Uncertainty is the state, even partial, of deficiency of information related to understanding or knowledge of an event, its consequence, or likelihood.
- An event is an occurrence or change of a particular set of circumstances and can have several causes
- A cause is that which gives rise to any action, phenomenon or condition
- A consequence is the outcome of an event affecting objectives
- Likelihood is the chance of something happening; risk is a combination of potential events and consequences along with the associated likelihood of occurrence

## 5.2. Risk Response Strategy

Identified risks needs to be addressed with appropriate risk response strategy. The risk response strategy can be of two types

- Reactive strategies
  - Very common, also known as fire fighting
  - Project team sets resources aside to deal with problems
  - Team does nothing until a risk becomes a problem
- Proactive strategies
  - Risk management begins long before technical work starts, risks are identified and prioritized by importance
  - Team builds a plan to avoid risks if they can or to minimize risks if they turn into problems

Based on the following matrix the project manager can identify the risk response strategy for the risks.



- Mitigating risk (Reduce)
  - Actions are taken during the project to either A) reduce the likelihood of a risk, or B) reduce the impact of risk
  - For example, testing electrical components after receipt would reduce the likelihood that “bad” parts would be used in a circuit
- Retaining risk (Accept)
  - Usually for events with low probability but high impact when no alternate strategy is feasible
  - Have a contingency plan ready in case event occurs
- Sharing risk or Transferring the risks (Share)
  - Multiple units associated with the project assume some portion of the risk
  - Risk is assumed and managed by a unit outside the immediate project
  - Transfer the risk to third-party / vendors
- Risk avoidance (Avoid)
  - Action is taken to halt the activities giving rise to risk