## Background about Risk

Risk Engineering, which consists of assessing and controlling risks, is a key component to successful project management.

Gambling establishments from the 17th and 18th century were the first known body to investigate risk. Various games were assessed to determine payback and discrete events were studied. Evaluating what playing card would be next or would the next ball be black and white introduced probability theory.

Insurance companies began using probability theory in the 19th century for continuous events, for example, insurance risk exposure was investigated to determine the probability of death at various ages.

From the 1950s through the 1970s decision theory took off, which is the study of identifying values, uncertainties, and related issues to understand a given decision and its justification. In the 1980s software projects began to include risk techniques to reduce errors and produce successful project. In the 1990s, the Software Engineering Institute (SEI) introduced "Team Risk Management" and increased emphasis on risk management. We will talk more about the SEI in Module 11.

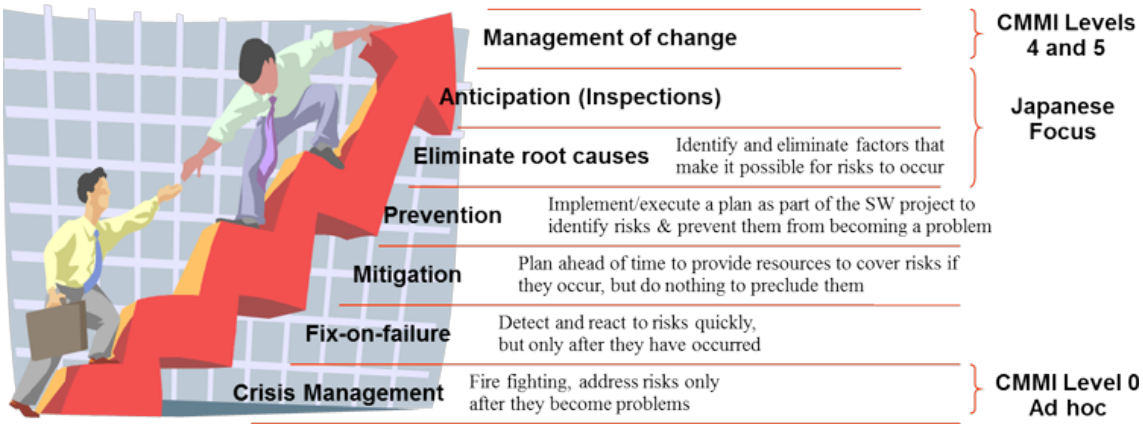
## Definition of Risk

Risk is defined as an event, action, or thing with:

1. A potential **loss** associated with it,
2. An **uncertainty** or chance involved, or
3. Some choice involved. Robert Charette introduced this definition in 1987.

Risk is the lack of resources such as information, control, or time to accomplish a task. Losses may be in the form of reduced quality of the delivered product, increased cost, delayed completion, or failure. The classic Operations Research definition of risk focuses only on the probability of occurrence, not on the consequences or loss. Risks deal with uncertainties and unknowns insofar as they lead to failure or produce adverse outcomes.


Robert N. Charette, Sr. Consultant with Agile Project Management Practice and director of the Risk Management Intelligence Network introduced the Evolutionary Hierarchy of Risk. He proposed that there are levels of risk. The lowest level is **Crisis Management**, where the team is fighting the "project fires." At this level, risks are addressed only after they become problems. On the Capability Maturity Model Integrated® (CMMI®) scale, this would be at level 0. **Fix-on-Failure** is the next level; when a risk is detected, the team reacts quickly. Again this takes place only after the risk has occurred. The third level is **Mitigation** where the team plans ahead of time to provide resources to cover risks as they occur, but the team does not take the next step to prevent them. In **Prevention**, the team implements and executes a plan as part of the software project to identify risks and prevent them from becoming a problem. The next level is **Eliminate Root Causes** where the team identifies and eliminates factors that make it possible for the risks to occur. This and the next step is where the Japanese car industry focused when their cars became popular in the 1990s. **Anticipation** suggests that you expect that you will have risks and you used techniques such as inspections to prevent and eliminate the risks. Lastly **Management of Change**, CMMI® Level 4 and 5 activities, is where the team manages change.



## Risk Engineering and Current Software Programs

Most Requests for Proposal (RFPs) require offerors to identify risks and provide risk mitigation strategies and the Statement of Work (SOW) probably requires risk management. Contract Deliverable Resource Lists (CDRLs) may include a Data Item Description (DID) for a Risk Assess Report. The commercial and Government standards such as ISO 12207 requires a section on risk. The risk management process must be defined for the project. One or more people assigned to the program must have funded responsibility for risk management.

An example from Earth Observing Satellite Data Information System Core System (ECS) SOW has paragraph 3.2.4 which specifies requirements for risk assessment. For example DID 210/SE3 is the Risk Assessment Report. Paraphrased it reads:

Design Analysis, Work Breakdown Structure (WBS) 2.4: The contractor shall conduct performance analyses and risk assessments of the design to evaluate the optimization, correlation, completeness, and risk associated with the design, and to ensure that the implemented system will meet all the requirements of the specification.

SE means the System Engineering Office responsible for production of the DID. SE3 means the document is reviewed and controlled by the contractor, that is, the document must be delivered to the customer for information only yet subject to approval. The document is due one month prior to each Incremental Design Review (IDR) and there will be five IDRs. This report:

- Identifies possible high risk areas in the design, manufacture, integration, and test areas,
- Assesses the risks identified in terms of technical objectives, goals, schedule, and budget; and
- Identifies appropriate plans to mitigate risks and provides costs for alternate or backup approaches.

## Two Phases of Risk: Assessment and Control

- **Assessment Phase**
  - Identification
  - Analysis
  - Prioritization

- **Control Phase**
  - Planning
  - Resolution
  - Monitoring

## Risk Assessment

## Identification

The purpose of Risk Identification is to identify and categorize potential problem areas. Identification is initiated during the proposal and continually updated during the project. Risks can be identified through:

- Visiting the customer operational sites
- Participating in customer working groups
- Examining results from Independent Research and Development (IR&D) programs
- Reviewing historical lessons learned data
- Analyzing assumptions from estimation
- Examining cost drivers from the cost estimation model
- Prototyping
- Benchmarking hardware
- Performing special trade studies or analyses
- Running performance models or simulation
- Analyzing poorly defined items in the specification and plans
- Using checklists of generic software problems

Barry Boehm and Robert Charette both have checklists of common software risk problems and you will notice that they are not identical.

| **Top 10 Checklist - Barry Boehm** | **Common Software Risks - Robert Charette** |
|---|---|
| 1. Personnel shortfalls<br>2. Unrealistic schedules and budgets<br>3. Developing the wrong software functions<br>4. Developing the wrong user interface<br>5. Gold plating (excessive requirements or overdesign)<br>6. Continuous stream of requirements changes (excessive requirements volatility)<br>7. Shortfall in externally furnished components<br>8. Shortfall in externally performed tasks<br>9. Real-time performance shortfalls<br>10. Straining computer science capabilities | 1. Inadequate development model<br>2. Inadequate software development plan<br>3. Unsuitable organizational structure<br>4. Too many new methodologies<br>5. Building a complex project from scratch |

The Software Engineering Institute (SEI) created a taxonomy approach to identify risks based on these assumptions:

- Software development risks are generally known by the development staff but not always communicated.
- A structured and repeatable method of risk identification will encourage consistent risk management.
- Effective risk management must cover all key development and support areas of the project.
- The risk identification process must create and sustain a non-judgmental, objective environment so that tentative or controversial views are heard.
- No overall judgment can be made about the success or failure of a project based solely on the number of risks uncovered.

The software taxonomy is organized into three major classes:

- **Product Engineering** covers the technical aspects of the work to be accomplished.
- **Development Environment** covers the methods, procedures, and tools used to produce the product.

- **Program Constraints** are the contractual, organizational, and operational factors within which the software is developed, but which are generally outside of the direct control of local management.

The classes are organized into 13 elements and 64 attributes.

The next item contains a video that provides a brief overview of software taxonomy.

## Analysis

Risk analysis assigns a quantitative value to identified risks. A quantitative value is preferred over low, medium, and high, although this range is often uses. Analysis is conducted to discover the cause, effect, and magnitude of the perceived risk. It is common to analyze risk in terms of the Risk Factor ($R_f$) which is determined by estimating the Probability of Failure ($P_f$) and the Consequences of Failure ($C_f$).

$P_f$ is determined by considering the maturity (m), complexity (c), and dependency (d) factors in the equation:

$P_f = (P_m + P_c + P_d) / 3$ where:

**Probability of Failure Calculation**

| Magnitude $P_f$ | Maturity Factor $P_m$ | Complexity Factor $P_c$ | Dependency Factor $P_d$ |
|---|---|---|---|
| 0.1 (low) | Existing design | Simple design | Independent of existing system, facility, or associate contractor |
| 0.5 (medium) | Major design change | Moderate increase in complexity | Performance dependent on existing system performance, facility, associate contractor |
| 0.9 (high) | State of art, some research complete | Extremely complex | Performance dependent on new system schedule, facility, or associate contractor |

$C_f$ is determined by considering the technical (t), cost (c), and schedule (s) factors in the equation:

$C_f = (C_t + C_c + C_s) / 3$ where:

**Consequences of Failure Calculation**

| Magnitude $C_f$ | Technical Factor $C_t$ | Cost Factor $C_c$ | Schedule Factor $C_s$ |
|---|---|---|---|
| 0.1 (low) | Minimal or no consequences | Budget estimates not exceeded, some $ transfer | Negligible impact on program, slight schedule change, compensated by available slack |
| 0.5 (medium) | Some reduction in technical | Cost estimates increased by 5% to 20% | Small slip in schedule |

| Magnitude $C_f$ | Technical Factor $C_t$ | Cost Factor $C_c$ | Schedule Factor $C_s$ |
|---|---|---|---|
| | performance | | |
| 0.9 (high) | Technical goals cannot be achieved | Cost estimates increased by > 50% | Large schedule slip, affects segment milestones or possible affect on system milestones |

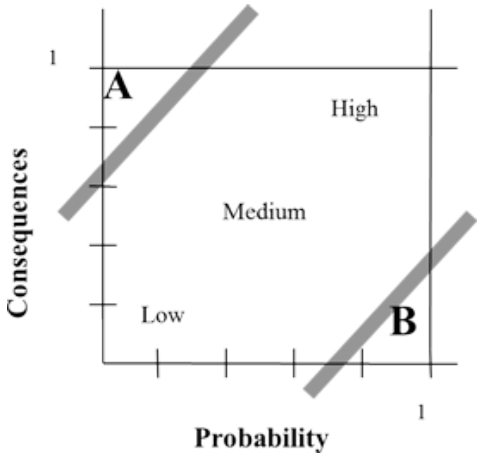
Then the resulting Risk Factor is defined as $R_f = P_f + C_f - (P_f * C_f)$, for example:

| $P_f$ | $C_f$ | $R_f$ |
|---|---|---|
| .7 | .7 | .7 + .7 − (.7*.7) = .90 |
| .5 | .7 | .5 + .7 − (.5*.7) = .85 |
| .5 | .3 | .5 + .3 − (.5*.3) = .65 |
| .4 | .2 | .4 + .2 − (.4*.2) = .52 |

Be aware that the Risk Factor is less meaningful when either probability or consequences of failure approach one or zero. For example: the consequences of failure of the fuel gauge in an airplane are technically unacceptable; hence, consequences could be assessed as one. Even if the probability of failure is zero, that is, the Risk Factor is one:

$$R_f = P_f + C_f - (P_f * C_f) = 0 + 1 - (0 * 1) = 1$$

Other factors should be considered for these situations in regions "A" and "B":




The product of the analysis step includes a list of quantified risk items with attributes. Each risk item must be saved and routinely maintained as in the example.

| Attribute | Description |
|---|---|
| Risk item number | Program unique identifier |
| Risk item name | Name of risk item |

| Attribute | Description |
|---|---|
| Description | Textual description of risk item |
| Potential impact | Description of impact to the program if the risk comes true, in terms of cost, schedule, performance |
| Risk factor | Numeric value based on probability and consequences of risk |
| Current mitigation/contingency plans | Specific activities to mitigate or recover from the risk, including status of plans and responsible organization |
| Monitoring thresholds | Specific values on a scale which defines success or requires additional action |
| Scale | Measurement scale relevant to monitoring thresholds for risk item |

A **monitoring threshold** which results in a reduction of the risk factor to a low level is the **success threshold**; while a monitoring threshold which requires additional risk related activity is an **action threshold**.

## Prioritization

In risk prioritization, the Risk Factor is used to sort the risks where the highest values get highest priority. Each risk factor is evaluated to determine most appropriate risk management plans to prioritize implementation strategies. Consider this sample data:

| Risk Item | $R_f$ | $C_f$ | $P_f$ |
|---|---|---|---|
| 1. Low probability, low consequences | 0.19 | 0.10 | 0.10 |
| 2. Low probability, high consequences | 0.73 | 0.70 | 0.10 |
| 3. High probability, low consequences | 0.73 | 0.10 | 0.70 |
| 4. High probability, high consequences | 0.91 | 0.70 | 0.70 |

The plans for Risk Item 2 should focus on reducing the consequences of the risk; the plans for Risk Item 3 should focus on reducing the probability of the risk, and further evaluation should concentrate on the components of the probability or consequences of failure.