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.

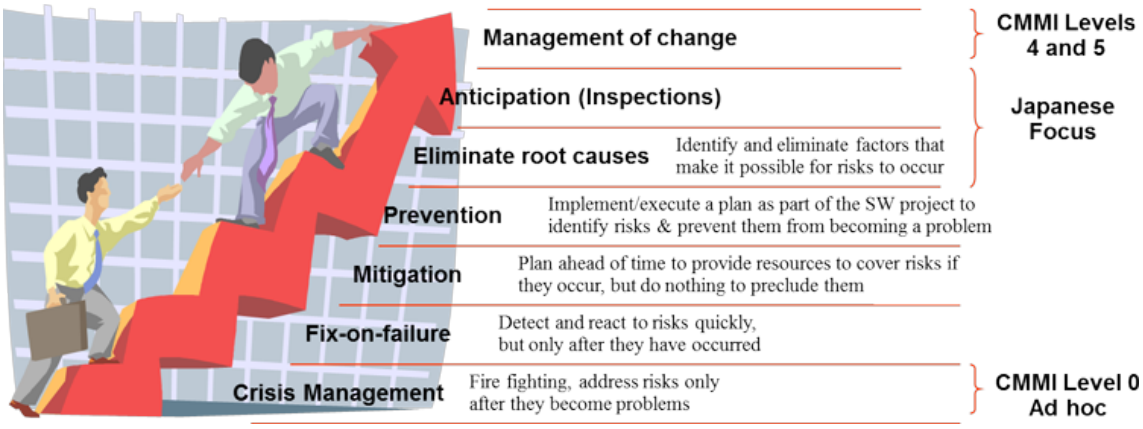
Definition of Risk

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 2. Unrealistic schedules and budgets 3. Developing the wrong software functions 4. Developing the wrong user interface 5. Gold plating (excessive requirements or overdesign) 6. Continuous stream of requirements changes (excessive requirements volatility) 7. Shortfall in externally furnished components 8. Shortfall in externally performed tasks 9. Real-time performance shortfalls 10. Straining computer science capabilities	1. Inadequate development model 2. Inadequate software development plan 3. Unsuitable organizational structure 4. Too many new methodologies 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

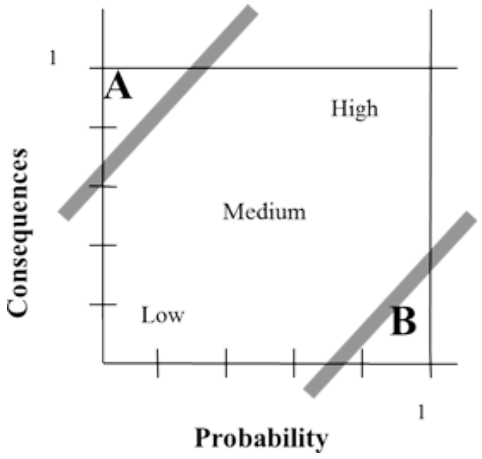
Then the resulting Risk Factor is defined as $R_f = P_f + C_f - (P_f * C_f)$, for example:

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:

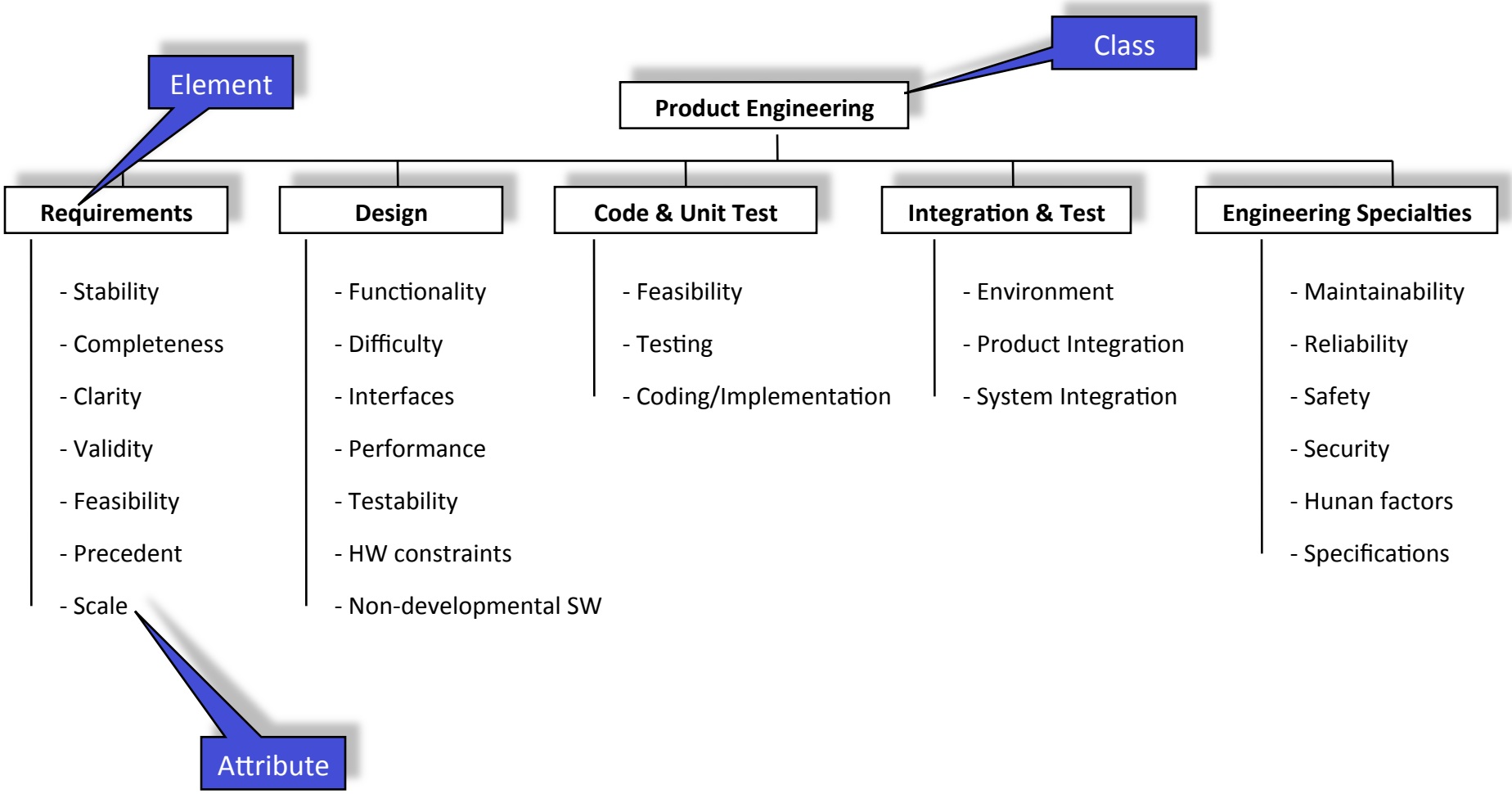
Risk Item	R_f	C_f	P_f
1. Low probability, low consequences	0.19	0.10	0.10
2. Low probability, high consequences	0.73	0.70	0.10
3. High probability, low consequences	0.73	0.10	0.70
4. High probability, high consequences	0.91	0.70	0.70

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.

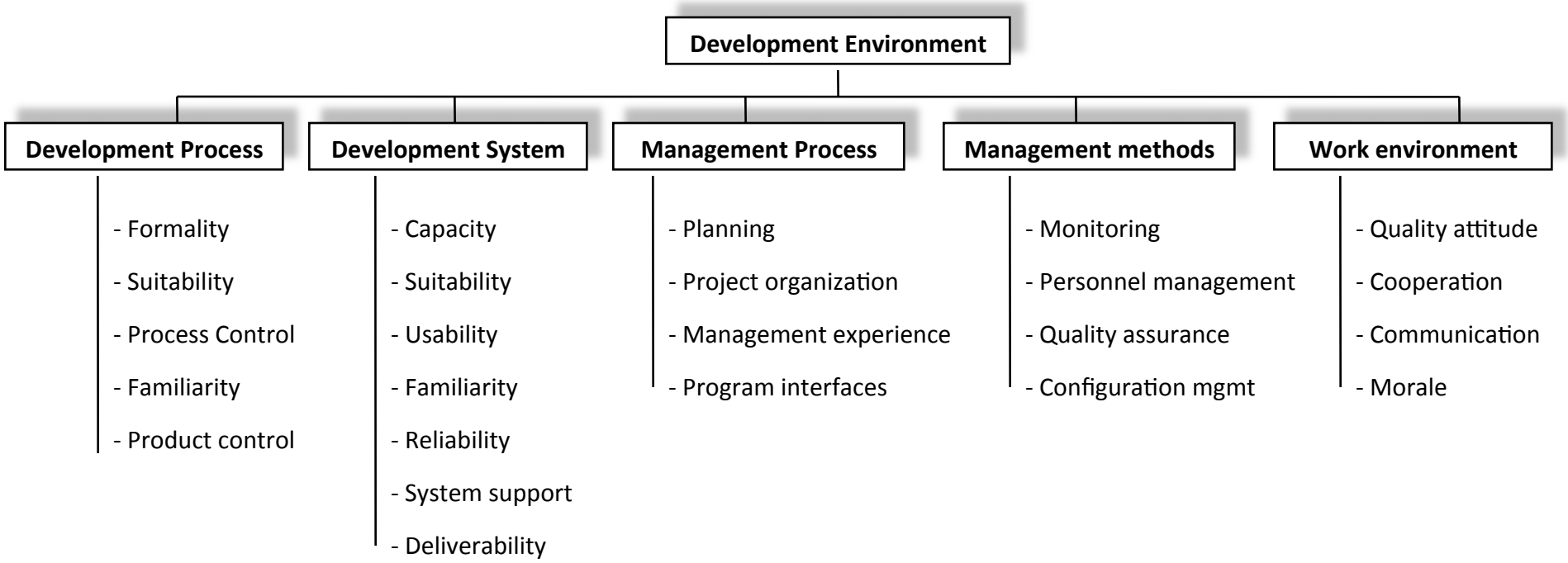
Software Engineering Institute Risk Taxonomy

Risk Identification

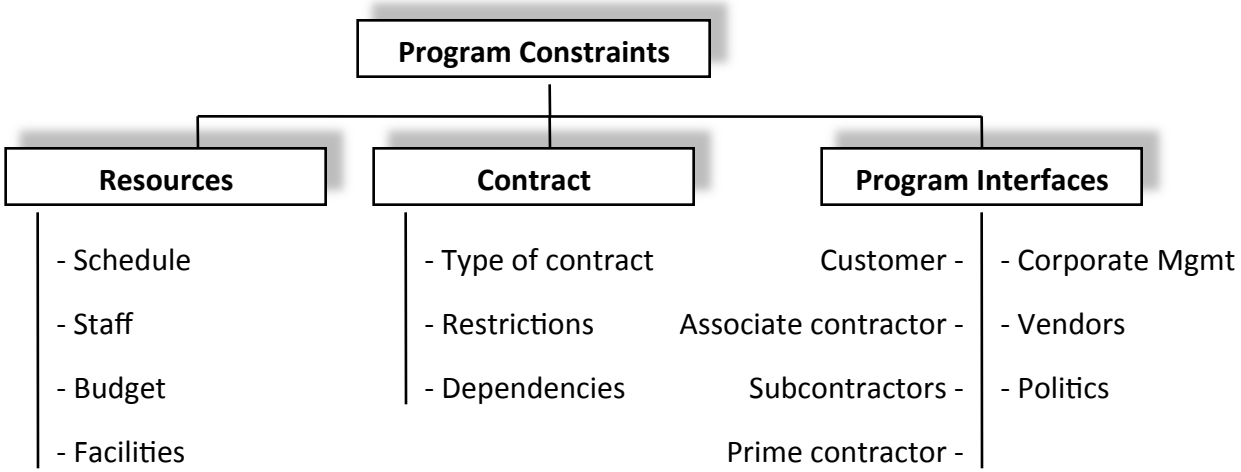
Product Engineering



Development Environment



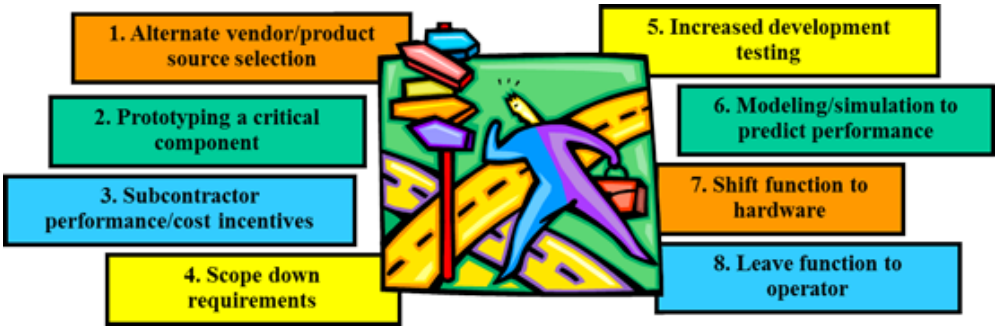
Program Constraints



Risk Control

Planning

The first step in Risk Control is Risk Planning. Feasible options are identified for managing the risk; examples of risk management options include:



Note that some strategies are aimed at avoiding risk including (2) prototyping, (4) scope down requirements, and (6) modelling and simulation, while others are aimed at transferring risk including (3) subcontractor performance and cost incentives, (7) shift function to hardware, and (8) leave function to operator. You need to compare effectiveness and the cost of various options. One method is to apply the Risk Reduction Leverage (RRL), a quality for price ratio, which is defined as:

RRL = (Risk Exposure Before − Risk Exposure After) / Risk Reduction Cost

For example, assume the loss due to a software interface error is estimated at $100,000 and the probability of error being introduced is estimated at 0.3 or 30%. Then the Risk Exposure Before = $100,000 x (.3) = $30,000. Your research shows that there are three approaches for eliminating the error:

- An interface (I/F) checking tool that costs $5,000 and reduces the probability to 0.2 or 20%,
- An I/F testing approach that costs $15,000 and reduces the probability to 0.1 or 10%, or
- Reusing tried and proven interface software that costs $3,000 and reduces the probability to 0.1 or 10%

Using this data, we can solve the equations for the RRL as:

RRL = (Risk Exposure Before − Risk Exposure After) / Risk Reduction Cost

Option 1: RRL (checker) = [$30K − ($100K x (.2))] / $5K = 2

Option 2: RRL (testing) = [$30K − ($100K x (.1))] / $15K = 1.33

Option 3: RRL (reuse) = [$30K − ($100K x (.1))] / $3K = 6.66 (**Best Solution**)

Now with this quality for price data, specific action plans including the implementation criteria for success and/or action thresholds can be developed for the options. The risk management plans may be **contingency plans**, in case the failure occurs, or **mitigation plans**, to proactively reduce the likelihood that the risk will occur, or both.

Project management then reviews the risk management plans, refines them, and the appropriate approach is selected. The plans must answer the following questions for each risk item:

- Why? Answering risk item importance and relation to project objectives
- What? When? Answering risk resolution deliverables, milestones, and activities
- Who? Where? Answering responsibilities and organization
- How? Answering the approach
- How much? Answering budget, schedule, and key personnel

Resolution

Effective risk resolution often hinges on finding a creative solution to the specific risk. Thus the solution depends on the risk. There are eight (8) generic methods to resolve risks.

Avoid the risk	Do not do the risky activity: negotiate elimination of the requirement or hire a consultant (specialist) to do the risky work
Transfer the risk	Buy special purpose hardware rather than develop software or find a COTS solution
Buy information about the risk	Spend money to investigate the risk: prototype, bring in a consultant to evaluate your design, or establish an IR&D project well ahead of time
Eliminate the root cause of the risk	Analyze the risk to look for underlying causes using techniques such as Kaizen, Lean, or Six Sigma; and minimize or eliminate root causes
Assume the risk	If the consequences are small and the cost to avoid the risk is large, simply accept the consequences if the risk occurs
Publicize the risk	Notify upper management, marketing, and the customers about the risk and consequences:

	minimize the surprise if it occurs and get buy in for resolutions
Control the risk	Accept that the risk might occur and develop plans to handle it if it does
Remember the risk	Build a collection of risk management plans and resolutions for use on future projects

Monitoring

In Risk Monitoring, the software manager tracks the status of open and potential risk areas. Technical, cost, and schedule performance of each implemented risk plan is qualitatively assessed at internal periodic project management reviews to:

- Verify outcomes were as envisioned,
- Identify opportunities for refinement of risk plan, and
- Provide feedback to those making programmatic decisions

Customer personnel are apprised of each risk area at the monthly progress reviews and major technical reviews. Risk monitoring techniques include collecting metrics, reviewing risks, and tracking the top ten risks as in the chart.

Top 10 Risk Tracking: Third Week

This week	Last week	Weeks on list	Risk Item	Risk resolution progress
1	1	3	Scoping product to fit required schedule	Working session with customer
2	4	3	Resolving subcontractor OS I/F uncertainties	New changes proposed by subcontractor
3	2	3	Algorithm boundary checking	Prototype on schedule
4	6	2	Workstation availability	Delay in delivery reported
5	5	3	User I/F definition	User study complete, awaiting prototype
6	3	3	Staffing lead developer for control software	Candidate identified, offer extended
7	-	1	Reusable text-formatting software uncertainties	Evaluation underway
8	7	3	Application performance risk	Benchmark to complete this week

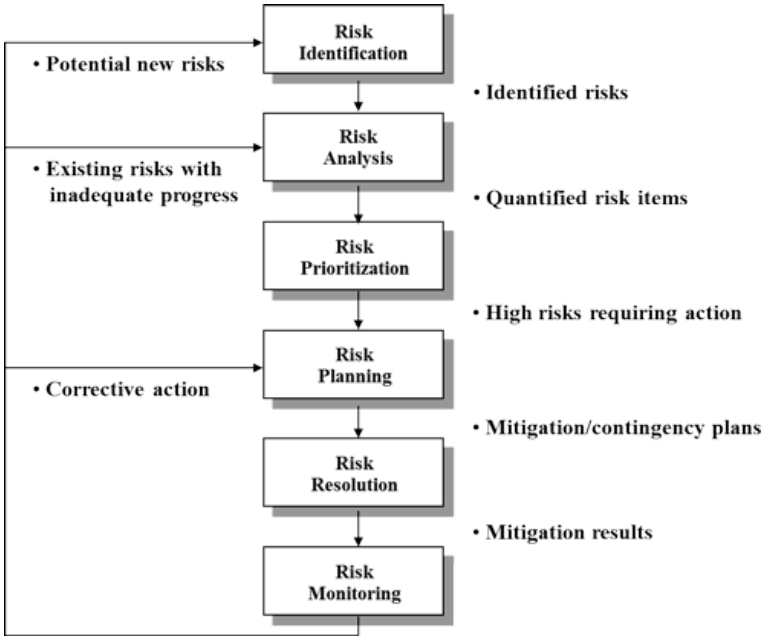
This week	Last week	Weeks on list	Risk Item	Risk resolution progress
9	9	2	CM procedures for incremental development	CM procedures identified, SEPG tailoring
10	8	2	Staffing programmer for control software	Programmer accepted offer; reports on 08/01

Risk Management Panel

Large programs/projects often create a Risk Management Panel (RMP) which should be documented in the appropriate Project Instruction. The purpose of the RMP is to provide accurate current data and multi-disciplinary inputs to Program/Project Management to make informed decisions about managing the risks. The panel typically includes the Deputy Program/Project Manager who acts as the RMP chair, the Chief System Engineer, the Quality Assurance Office Manager, the Software Development Manager, the Operations and Maintenance Manager, the Independent Test Office Manager, the Commercial off the Shelf (COTS) Procurement Manager and a Customer Representative. The RMP acts as a reviewing body at each checkpoint of the risk management process: identification, analysis, prioritization, planning, resolution, and monitoring. The outputs of the RMP are approvals and redirections for each checkpoint in the risk management process. Decisions of the RMP are reported to other relevant management forums such as the Project Review, Configuration Control Boards (CCBs), and monthly status meetings. Issues or feedback from these forums may require the RMP to deliberate further.

Risk Management Summary

In summary, the risk engineering process is defined as:



Organizations have a choice. They can take a risk based management approach or they can choose not to invest in reducing risk. There are positives and negatives to both approaches, but the software industry's current stance is the risk based management approach is better and cheaper in the long run.

	Risk-based Management	No Risk Investment
Orientation	"Realistically oriented", risks set up boundary conditions, foresee and contain possible failure, proactive not reactive	"Success oriented", "can do" attitude, minimize thought of possible failure, leave little extra to handle contingencies
Problems	• Risk analysis may be inaccurate, underestimation can cause major problems, overestimation can lead to excessive mitigation effort • Takes resources away from other tasks • Effects of risk engineering hard to measure: total avoidance removes proof that cost was justified	• Reactive • No planning • No options for management consideration
Benefits	• Better perception of risks & options for management: reduction in project exposure to risk • Improved credibility of plans • More proactive	• Funds are not diverted to risk mitigation activities • "Can do" attitude
Example: Think of the car industry in the 1990s	U.S. software development • Change to meet user-requested requirements, independent of component availability • More functional, higher cost, less reliable • Greater need for risk-based management approach	Japanese software development • Change functionality in response to availability of components • Less functional yet more reliable • Less need for risk management