# Risk Analysis & Management

Module 5

# Risk Analysis and Management:

- Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty.
- 1. Risk identification
- 2. Risk Projection
- 3. Risk assessment
- 4. Risk management

# Risk Management Process:

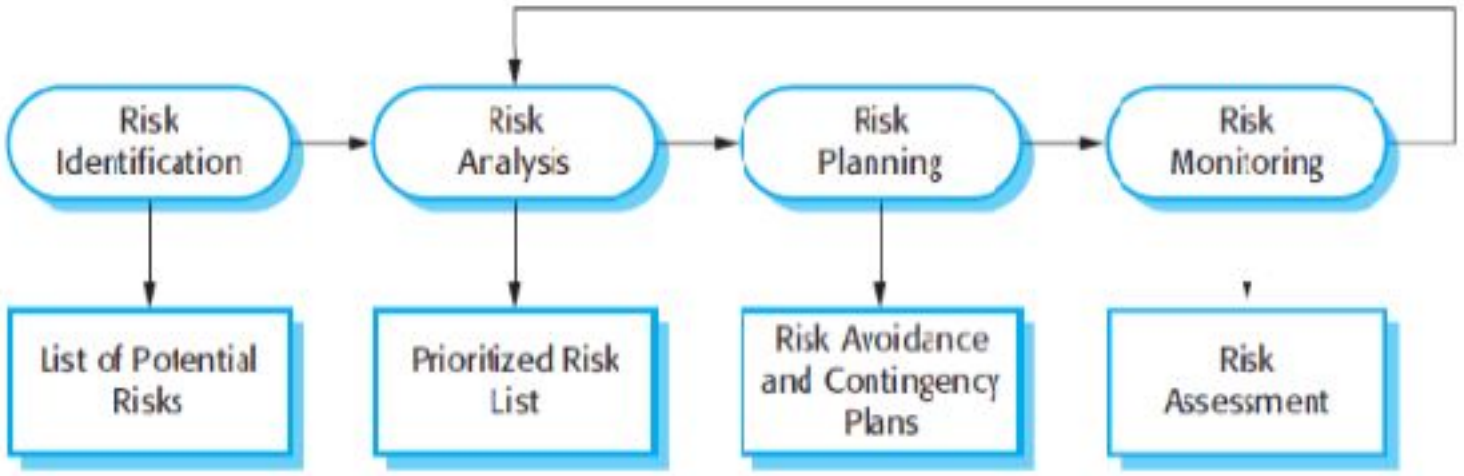



**Fig 1. Risk management process**

# Risk Characteristics:

Risk always involves two characteristics:

- ***Uncertainty***—the risk may or may not happen; that is, there are no 100% probable risks.
- ***Loss***—if the risk becomes a reality, unwanted consequences or losses will occur.

# Types of Risks:

- ***Project risks*** threaten the project plan. That is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and
- requirements problems and their impact on a software project.
- ***Technical risks*** threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible.
- ***Business risks*** threaten the viability of the software to be built. Business risks often jeopardize the project or the product. Candidates for the top five business risks are
- Market risk,
- Strategic risk,
- Management risk, and
- Budget risks.

- ***Known risks*** are those that can be uncovered after careful evaluation of the project plan.
- ***Predictable risks*** are extrapolated from past project experience (e.g., staff turnover, poor
- communication with the customer, dilution of staff effort as ongoing maintenance
- requests are serviced).
- ***Unpredictable risks*** are the joker in the deck. They can and do occur, but they are
- extremely difficult to identify in advance.

- **Reactive risk strategies:-**
- 1. Reactive risk strategies follows that the risks have to be tackled at the time of their
- occurrence.
- 2. No precautions are to be taken as per this strategy.
- 3. They are meant for risks with relatively smaller impact.
- **Proactive risk strategies:-**
- 1. Proactive risk strategies follows that the risks have to be identified before start of
- the project.
- 2. They have to be analyzed by assessing their probability of occurrence, their
- impact after occurrence, and steps to be followed for its precaution.
- 3. They are meant for risks with relatively higher impact.

# Risk identification

- ***Risk identification*** is a systematic attempt to specify threats to the project plan
- One method for identifying risks is to create a **risk item checklist.**


Risk events and their relationships are defined
Probabilities and consequences of risk events are assessed
Assess Probaility & Consequence
Identify Risks
1. Risk Identification
2. Risk Impact Assessment
Consequences may include cost, schedule, technical performance impacts, as well as capability or functionality impacts
Reassess existing risk events and identify new risk events
Risk Tracking
Watch-listed Risks
Assess Risk Criticality
4. Risk Mitigation Planning, Implementation, and Progress Monitoring
Risk Mitigation
3. Risk Prioritization Analysis
Decision-analytic rules applied to rank-order identified risk events from "most to least" critical
Risk events assessed as medium or high criticality might go into risk mitigation planning and implementation; low critical risks might be tracked/monitored on a watch list.

- ***Product size***—risks associated with the overall size of the software to be built or modified.
- ***Business impact***—risks associated with constraints imposed by management or the marketplace.
- ***Customer characteristics***—risks associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
- ***Process definition—***risks associated with the degree to which the software process has been defined and is followed by the development organization.
- ***Development environment***—risks associated with the availability and quality of the tools to be used to build the product.
- ***Technology to be built***—risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
- ***Staff size and experience***—risks associated with the overall technical and project experience of the software engineers who will do the work.

| Risk ID | Risk Description | Probability | Influence | Effect on Cost/ Schedule/Quality |
|---|---|---|---|---|
| RI.1 | Late submission of information, delays in document approval by the Customer | Medium | High | Schedule |
| RI.2 | Incorrect or incomplete stated requirements | High | High | Cost, Schedule |
| RI.3 | Changes in the requirements during development | High | High | Cost, Schedule |
| RI.4 | Problems with the delivery of the product into production because of the unavailability of servers. | High | Medium | Schedule |
| RI.5 | Problems of integration with internal systems of the Customer | Medium | Medium | Cost, Schedule, Quality |
| RI.6 | Tight time limits that influence the testing flow | Medium | High | Cost, Schedule, Quality |

# I. Assessing Overall Project Risk

- Is the software project we're working on at serious risk?
- The questions are ordered by their relative importance to the success of a project.
- 1. Have top software and customer managers formally committed to support the project?
- 2. Are end-users enthusiastically committed to the project and the system/product to be built?
- 3. Are requirements fully understood by the software engineering team and their customers?
- 4. Have customers been involved fully in the definition of requirements?

- 5. Do end-users have realistic expectations?
- 6. Is project scope stable?
- 7. Does the software engineering team have the right mix of skills?
- 8. Are project requirements stable?
- 9. Does the project team have experience with the technology to be implemented?
- 10. Is the number of people on the project team adequate to do the job?
- 11. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
- If any one of these questions is answered negatively, mitigation, monitoring, and management steps should be instituted without fail.

# II. Risk Components and Drivers

- The risk components are defined in the following manner:
- **Performance** risk—the degree of uncertainty that the product will meet its requirements and be fit for its intended use.
- **Cost** risk—the degree of uncertainty that the project budget will be maintained.
- **Support** risk—the degree of uncertainty that the resultant software will be easy to correct, adapt, and enhance.
- **Schedule** risk—the degree of uncertainty that the project schedule will be maintained and that the product will be delivered on time.
- The impact of each risk driver on the risk component is divided into one of four impact categories—**negligible, marginal, critical, or catastrophic.**

# Impact assessment

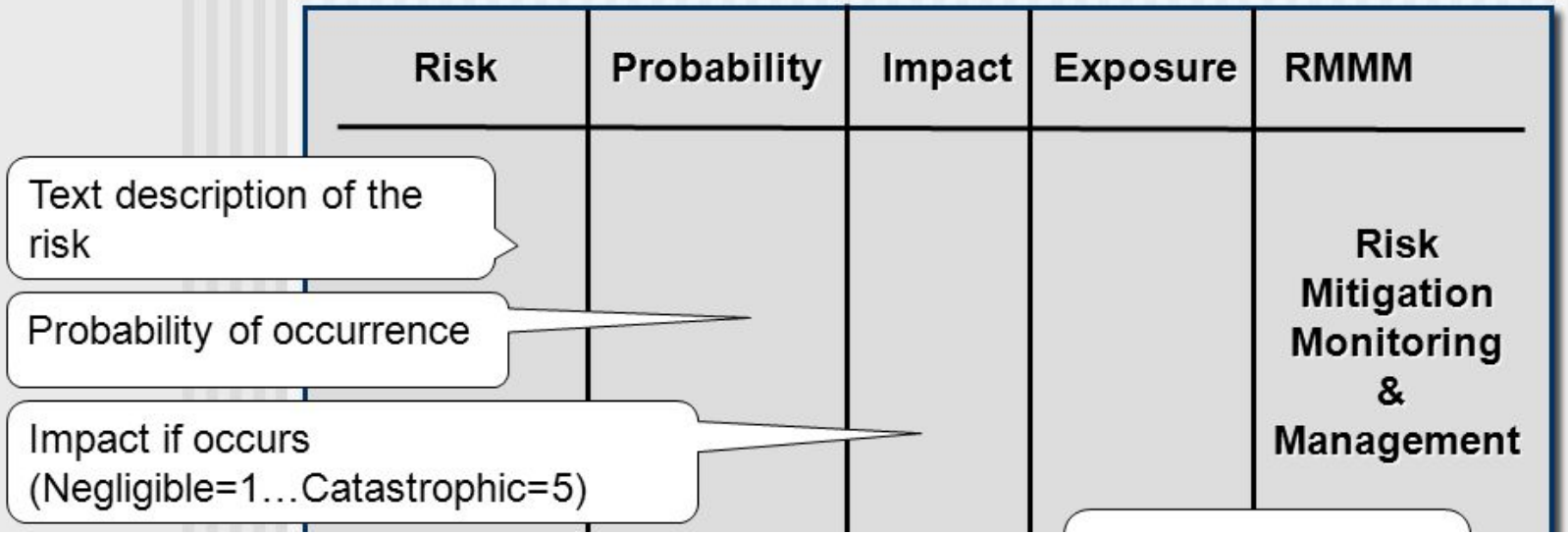
| Category \ Components | | Performance | Support | Cost | Schedule |
|---|---|---|---|---|---|
| Catastrophic | 1 | Failure to meet the requirement would result in mission failure | | Failure results in increased costs and schedule delays with expected values in excess of $500K | |
| | 2 | Significant degradation to nonachievement of technical performance | Nonresponsive or unsupportable software | Significant financial shortages, budget overrun likely | Unachievable IOC |
| Critical | 1 | Failure to meet the requirement would degrade system performance to a point where mission success is questionable | | Failure results in operational delays and/or increased costs with expected value of $100K to $500K | |
| | 2 | Some reduction in technical performance | Minor delays in software modifications | Some shortage of financial resources, possible overruns | Possible slippage in IOC |
| Marginal | 1 | Failure to meet the requirement would result in degradation of secondary mission | | Costs, impacts, and/or recoverable schedule slips with expected value of $1K to $100K | |
| | 2 | Minimal to small reduction in technical performance | Responsive software support | Sufficient financial resources | Realistic, achievable schedule |
| Negligible | 1 | Failure to meet the requirement would create inconvenience or nonoperational impact | | Error results in minor cost and/or schedule impact with expected value of less than $1K | |
| | 2 | No reduction in technical performance | Easily supportable software | Possible budget underrun | Early achievable IOC |

Note: (1) The potential consequence of undetected software errors or faults.
(2) The potential consequence if the desired outcome is not achieved.

# RISK PROJECTION

- *Risk projection,* also called *risk estimation,* attempts to rate each risk in two ways—the likelihood or probability that the risk is real and the consequences of the problems associated with the risk, should it occur.
- The project planner, along with other managers and technical staff, performs four risk projection activities.
- 1. Establishing a scale that reflects the perceived likelihood of a risk.
- 2. Delineating the consequences of the risk.
- 3. Estimating the impact of the risk of the project and the product.
- 4. Noting the overall accuracy of the risk projection so that there will be no misunderstandings.

# Building a Risk Table



# RISK MATRIX

| RISK RATING KEY | LOW | MEDIUM | HIGH | EXTREME |
|---|---|---|---|---|
| | 0 – ACCEPTABLE<br>OK TO PROCEED | 1 – ALARP<br>as low as reasonably practicable<br>TAKE MITIGATION EFFORTS | 2 – GENERALLY UNACCEPTABLE<br>SEEK SUPPORT | 3 – INTOLERABLE<br>PLACE EVENT ON HOLD |

SEVERITY →

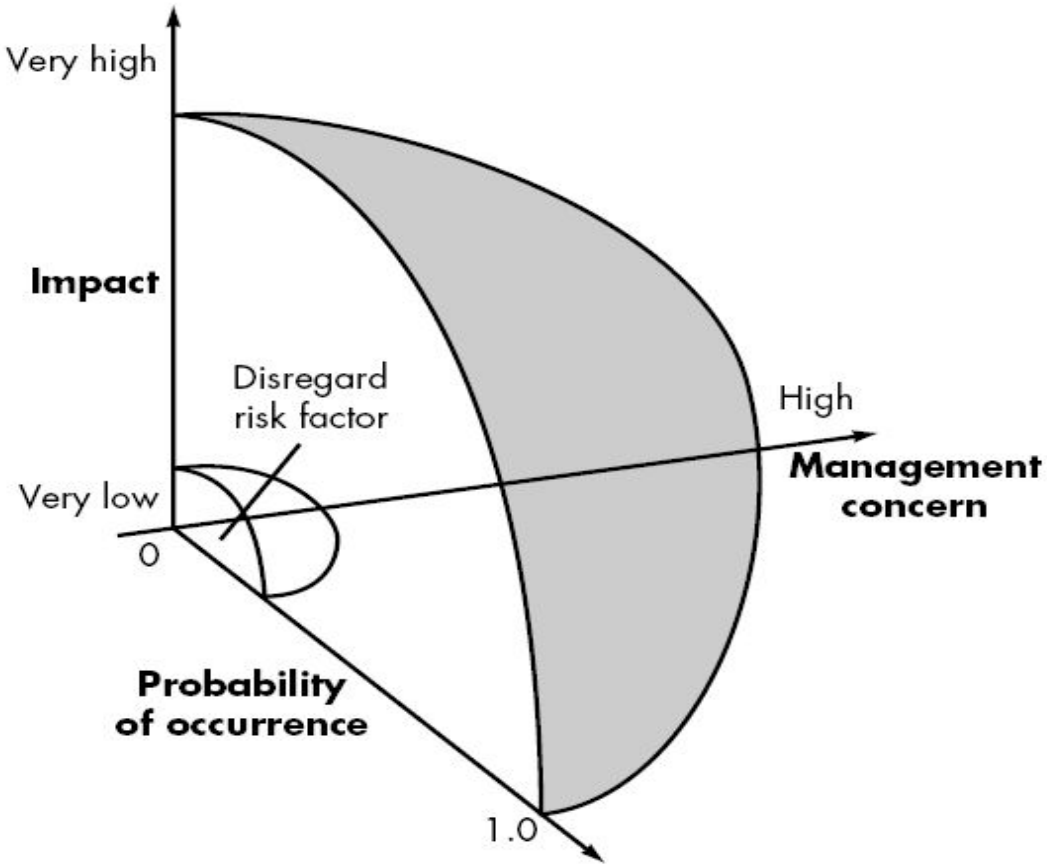
| LIKELIHOOD ↓ | ACCEPTABLE<br>LITTLE TO NO EFFECT ON EVENT | TOLERABLE<br>EFFECTS ARE FELT, BUT NOT CRITICAL TO OUTCOME | UNDESIRABLE<br>SERIOUS IMPACT TO THE COURSE OF ACTION AND OUTCOME | INTOLERABLE<br>COULD RESULT IN DISASTER |
|---|---|---|---|---|
| IMPROBABLE<br>RISK IS UNLIKELY TO OCCUR | LOW<br>– 1 – | MEDIUM<br>– 4 – | MEDIUM<br>– 6 – | HIGH<br>– 10 – |
| POSSIBLE<br>RISK WILL LIKELY OCCUR | LOW<br>– 2 – | MEDIUM<br>– 5 – | HIGH<br>– 8 – | EXTREME<br>– 11 – |
| PROBABLE<br>RISK WILL OCCUR | MEDIUM<br>– 3 – | HIGH<br>– 7 – | HIGH<br>– 9 – | EXTREME<br>– 12 – |

# I. Developing a Risk Table

| Risks | Category | Probability | Impact | RMMM |
|---|---|---|---|---|
| Size estimate may be significantly low | PS | 60% | 2 | |
| Larger number of users than planned | PS | 30% | 3 | |
| Less reuse than planned | PS | 70% | 2 | |
| End-users resist system | BU | 40% | 3 | |
| Delivery deadline will be tightened | BU | 50% | 2 | |
| Funding will be lost | CU | 40% | 1 | |
| Customer will change requirements | PS | 80% | 2 | |
| Technology will not meet expectations | TE | 30% | 1 | |
| Lack of training on tools | DE | 80% | 3 | |
| Staff inexperienced | ST | 30% | 2 | |
| Staff turnover will be high | ST | 60% | 2 | |
| • | | | | |
| • | | | | |
| • | | | | |

Impact values

**FIGURE 6.3**
Risk and management concern



# For example, assume that the software team defines a project risk in the following manner:

- **Risk identification. Only 70 percent of the software components scheduled for reuse** will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.
- **Risk probability. 80% (likely).**
- **Risk impact. 60 reusable software components were planned. If only 70 percent can be** used, 18 components would have to be developed from scratch (in addition to other custom software that has been scheduled for development). Since the average component is 100 LOC and local data indicate that the software engineering cost for each LOC is $14.00,
- the overall cost (impact) to develop the components would be 18 x 100 x 14 = $25,200.
- **Risk exposure.** ***RE = 0.80 x 25,200 ~ $20,200.***

# RISK MITIGATION, MONITORING, AND MANAGEMENT

- risk avoidance
- risk monitoring
- risk management and contingency planning

- **risk mitigation** is a problem avoidance activity.
- **Risk monitoring** is a project tracking activity with three primary objectives:
- (1) to assess whether predicted risks do, in fact, occur;
- (2) to ensure that risk aversion steps defined for the risk are being properly applied; and
- (3) to collect information that can be used for future risk analysis.

- **Risk Mitigation :**
  It is an activity used to avoid problems (Risk Avoidance).
  Steps for mitigating the risks as follows.

1. Finding out the risk.
2. Removing causes that are the reason for risk creation.
3. Controlling the corresponding documents from time to time.
4. Conducting timely reviews to speed up the work.

- **Risk Monitoring :**
  It is an activity used for project tracking.

1. To check if predicted risks occur or not.
2. To ensure proper application of risk aversion steps defined for risk.
3. To collect data for future risk analysis.
4. To allocate what problems are caused by which risks throughout the project.

- **Risk Management and planning :**
  It assumes that the mitigation activity failed and the risk is a reality. This task is done by Project manager when risk becomes reality and causes severe problems. If the project manager effectively uses project mitigation to remove risks successfully then it is easier to manage the risks. This shows that the response that will be taken for each risk by a manager. The main objective of the risk management plan is the risk register. This risk register describes and focuses on the predicted threats to a software project.

- https://www.geeksforgeeks.org/risk-mitigation-monitoring-and-management-rmmm-plan/

## Risk information sheet

| Risk ID: P02-4-32 | Date: 5/9/02 | Prob: 80% | Impact: high |
|---|---|---|---|

**Description:**
Only 70 percent of the software components scheduled for reuse will, in fact, be integrated into the application. The remaining functionality will have to be custom developed.

**Refinement/context:**
Subcondition 1: Certain reusable components were developed by a third party with no knowledge of internal design standards.
Subcondition 2: The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components.
Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.

**Mitigation/monitoring:**
1. Contact third party to determine conformance with design standards.
2. Press for interface standards completion; consider component structure when deciding on interface protocol.
3. Check to determine number of components in subcondition 3 category; check to determine if language support can be acquired.

**Management/contingency plan/trigger:**
*RE* computed to be $20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly.
Trigger: Mitigation steps unproductive as of 7/1/02

**Current status:**
5/12/02: Mitigation steps initiated.

| Originator: D. Gagne | Assigned: B. Laster |
|---|---|

## Risk Summary

View All

| Impact | Probability: Rare | Unlikely | Possible | Likely | Certain |
|---|---|---|---|---|---|
| Catastrophic | | 1 | | | |
| Critical | 1 | 1 | | | 1 |
| Marginal | | 1 | 1 | | |
| Negligable | | | | | |

## Top Open Risks

View All

| Description | Exposure | Type | Owned By | Review Date |
|---|---|---|---|---|
| The v1.1 release may not be ready in time | 15 | Schedule | Fred Bloggs | |
| We may not get enough authors to sign up | 8 | Business | Fred Bloggs | |
| The database may not support the volume | 6 | Technical | Joe P Smith | |
| The book pages may not be easy enough to use | 6 | Business | Joe P Smith | |
| The software licenses may be too expensive | 4 | Financial | Fred Bloggs | |

| No. | Risk Item | Risk Management Techniques |
|---|---|---|
| 1 | Personnel shortfalls | Staffing with top talent, job matching; teambuilding; morale building; cross-training; pre-scheduling key people |
| 2 | Unrealistic schedules and budgets | Detailed, multisource cost and schedule estimation; Design to cost; incremental development; software reuse; requirements scrubbing |
| 3 | Developing the wrong software functions | Organization analysis; mission analysis; ops-concept formulation; user surveys; prototyping; early users' manuals |
| 4 | Developing the wrong user interface | Task analysis; prototyping; scenarios; user characterization (functionality, style, workload) |
| 5 | Gold plating | Requirements scrubbing prototyping; cost-benefit analysis; design to cost |
| 6 | Continuing stream of requirement changes externally furnished components | High change threshold; information hiding; incremental development (defer changes to later increments) |
| 7 | Shortfalls in computer-science capabilities | Benchmarking; inspections; reference checking; compatibility analysis |
| 8 | Shortfalls in externally performed tasks | Reference checking; pre-award audits; award-fee contracts; competitive design or prototyping; teambuilding |
| 9 | Real-time performance shortfalls | Simulation; benchmarking; modeling; prototyping; instrumentation; tuning |
| 10 | Straining Computer-science capabilities | Technical analysis; cost-benefit analysis; prototyping; reference checking |

- https://www.guru99.com/risk-based-testing.html