Lecture Slides for
Managing and Leading Software Projects

Chapter 9: Risk Management

developed by

Richard E. (Dick) Fairley, Ph.D.

to accompany the text

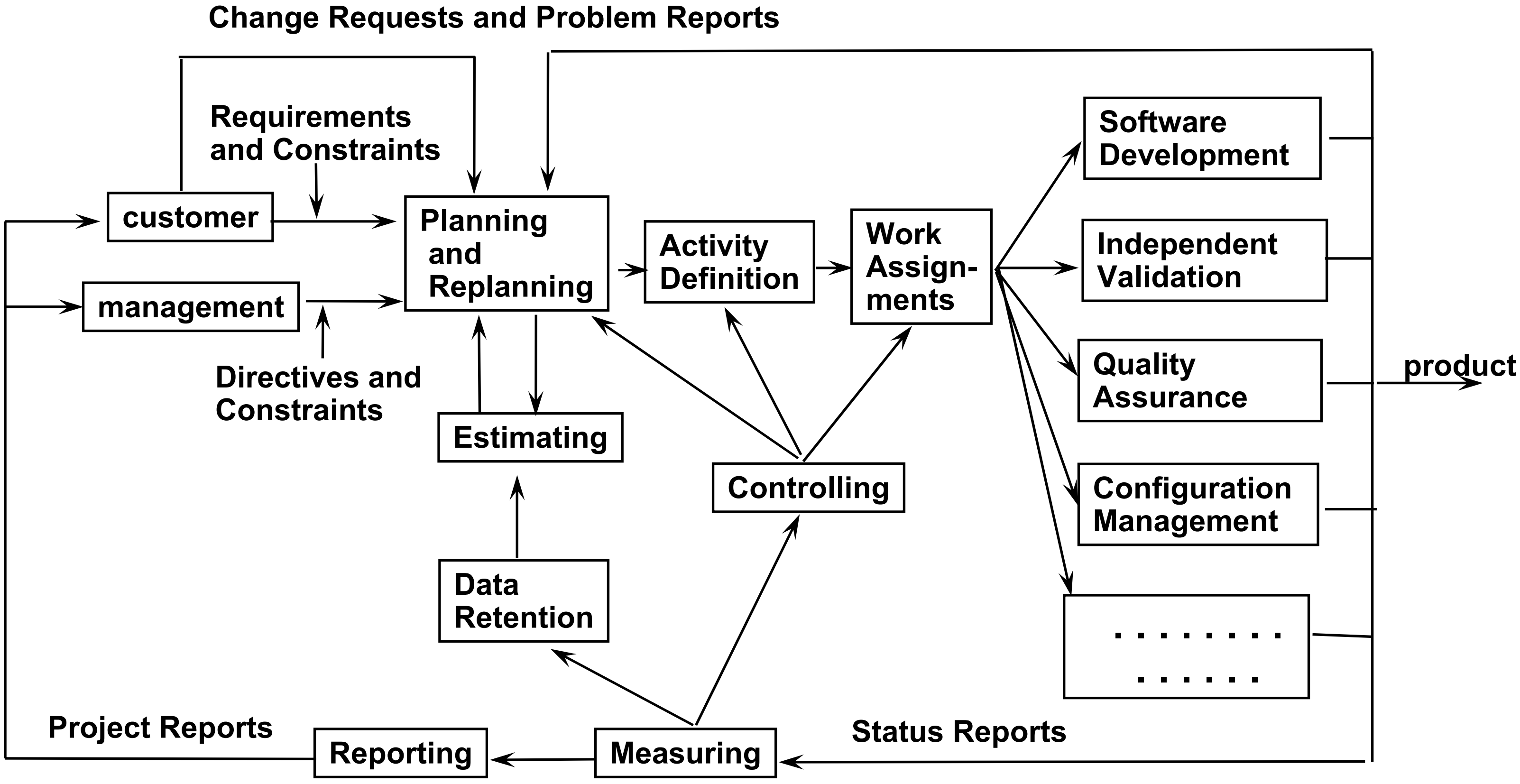
Managing and Leading Software Projects

published by Wiley, 2009

FOUR TYPES OF PROJECT MANAGEMENT ACTIVITIES

1. Plan and Estimate
   - tasks, schedule, budget, resources
2. Measure and Control
   - schedule, budget, resources, work products (quantity & quality), risk factors
3. Communicate and Coordinate
   - help people do their work activities
   - represent the project to others
4. Manage Risk
   - Identify and confront risk factors

A Workflow Model for Software Projects



Risk management is ubiquitous throughout the workflow model

Chapter 9 Topics

- Introduction to risk management
- A risk management paradigm
- Risk areas for software projects
- The risk management process
- Crisis management

Additional Sources of Information (1)

- Because software projects are inherently high risk endeavors, each of the frameworks, standards, and guidelines presented in this text (CMMI-DEV-v1.2, ISO/IEEE Standard 12207, IEEE Standard 1058, and PMBOK) include procedures and recommended practices for risk management.
- The relevant aspects of these standards and guidelines are contained in Appendix 9A of Chapter 9.

Additional Sources of Information (2)

- In addition, an overview IEEE/EIA Standard 1540-2001 for Life Cycle Processes—Risk Management is presented in appendix 9A.
- Additional terms used in this chapter and throughout this text are defined in Appendix A to the text. Terms specific to risk management are defined in Appendix 9B to this chapter.
- These presentation slides and other supporting material are available at the URL listed in the Preface.

Objectives for Chapter 9 (1)

- After reading this chapter and completing the exercises you should understand:
  - the terminology of risk management for software projects
  - the role of conventional project management techniques in managing generic risk factors for software projects
  - methods and techniques used to identify, analyze, prioritize, and mitigate project-specific risk factors
  - risk mitigation strategies of avoidance, transfer, acceptance, immediate action, and contingency plans and actions

Objectives for Chapter 9 (2)

- After reading this chapter and completing the exercises you should understand:
  - contents of risk management plans
  - top-N risk tracking and reporting
  - format, content, and use of risk registers
  - crisis management for software projects
  - risk management at the organizational level
  - joint risk management with customers and subcontractors

Annotated IEEE 1058, Section 5.4: Risk Management Plan (1)

5.4 Risk Management Plan

What mechanisms will be used to identify, analyze, and prioritize project risk factors?

How will contingency plans be developed?

What methods will be used to track the identified risk factors, evaluate changes in the levels of risk factors, and respond to those changes?

How will risk factors be identified, assessed, and mitigated on an on-going basis during the project?

the annotated version of IEEE Standard 1958 is available at the URL listed in the Preface to the text

Annotated IEEE 1058, Section 5.4: Risk Management Plan (2)

Risk factors that should be considered include:

- risks in the acquirer-supplier relationship,
- contractual risks,
- technological risks,
- risks caused by the size and complexity of the product,
- risks in the development and target environments,
- risks in personnel acquisition, skill levels, and retention,
- risks to schedule and budget, and
- risks in achieving user, customer, and acquirer acceptance of the product.

Risk Factors*

- A risk factor is a potential problem
- Risk factors are characterized by
  1. probability: 0 < p < 1

     or Low, Medium, High
  2. impact: negative consequences

     loss of life, money, resources, property

     often quantified in dollars or utility

     or Low, Medium, High
  3. urgency

     the timeframe within which the potential problem may become a real problem

* "Software Risk Management" by R. Fairley
IEEE Software, May/June, 2005

Some Commonly Occurring Risk Factors

Risk factors	**Examples**
Schedule	Inadequate calendar time
Budget	Insufficient funds when needed
Requirements	Infeasible, unstable, incorrect, incomplete, inconsistent
Personnel	Recruitment, ability, retention
Process	Inefficient and /or ineffective procedures
Resources	Host & target machines; supporting organizations
Technology	Platform and domain
Geography	Multiple development sites
External factors	Vendors and subcontractors
Operational risks	Missing features, inadequate performance
Quality	User and customer dissatisfaction
Maintenance	Corrections, missing features

Risk Factors

- In general, risk factors in the following areas and the tradeoffs among them must be identified and confronted:
  - Cost
  - Schedule
  - Resources
  - Product Objectives
    - Product Features
    - Quality Attributes
  - Assumptions
  - Constraints

Aggregated Risk

- Aggregated risk is the collection of risk factors that have the potential to negatively impact a desired outcome
- Risk factors within an aggregate are typically analyzed by considering:
  - o risk exposure of each risk factor: RE = p * I
  - o urgencies of the risk factors
  - o social and political contexts

aggregated risk may include compound and conditional risk factors

- compound: one event may trigger several potential problems
- conditional: a subsequent risk is based on the condition of an "upstream" risk factor becoming a problem

What is Opportunity?

- The *chance* that something *desirable* will happen: e.g., first to market, reduced cost, exceed customer expectations
- A situation is considered an opportunity if:

  1. uncertainty is involved: $(0 < p < 1)$

  2. a gain is associated with it: life, property, money, reputation

- One person's risk is another's opportunity

The Goal of Risk Management

The primary goal of risk management is to identify and respond to potential problems with sufficient lead time to prevent crisis situations

- a risk is a potential problem
- a problem is a risk that has materialized
- a crisis is a "show-stopper"

How Does Risk Management Differ from Traditional Project Management?

- The conventional techniques of project management presented in this text can be thought of as *institutionalized* risk management.
- Over time, it has become apparent that applying conventional techniques of project management improve the chance of success by reducing risk exposure
  - o it is better to do conventional project management than to not do it.

Conventional Project Management

- Conventional techniques of project management include:
  - o planning and estimating
  - o managing requirements,
  - o preparing work breakdown structures,
  - o establishing schedule networks, and
  - o measuring progress using techniques such as:
    - iterative development,
    - binary tracking, and
    - earned value reporting

How Does Risk Management Augment Traditional Project Management? (1)

- First, you can actively manage assumptions and constraints by:
  - explicitly itemizing them,
  - identifying the associated risk factors,
  - prioritizing the risk factors,
  - tracking risk indicator metrics,
  - periodically reviewing the risk factors, and
  - pursuing mitigating actions as necessary.

How Does Risk Management Augment Traditional Project Management? (2)

- Second, you can:
  - set threshold values for the risk indicator metrics and other project parameters
    - e.g., schedule performance index, cost performance index, requirements volatility and
  - prepare responses (i.e., develop mitigation plans) to be initiated if those thresholds are exceeded.

Ways to Adjust for Problems Encountered

- Acceptable methods include:
  - increasing the schedule;
  - increasing the budget;
  - applying more resources;
  - apply better resources;
  - reducing the requirements; and
  - improve the work processes
- Unacceptable methods include:
  - requiring excessive overtime;
  - reducing verification and validation activities; and
  - reducing user, customer, support, and maintenance aids and documentation.

How Does Risk Management Augment Traditional Project Management? (3)

- The third way in which risk management augments conventional project management is by using a systematic approach to:
  - o identify,
  - o analyze,
  - o prioritize, and
  - o mitigate specific risk factors

  for your project, during initial planning

  and on an on-going basis.

Risk Management Techniques (1)

- Table 9.3 in Chapter 9 of the textbook lists some typical risk factors and risk management techniques for dealing with them.

Risk Management Techniques (2)

Risk Management involves:

- risk identification
  - section 9.5 of the text describes several risk identification techniques
- analysis and prioritization
  - section 9.6 describes techniques for analysis and prioritization risk factors
- development and implementation of mitigation strategies
  - section 9.7 presents mitigation strategies

Risk Identification

Risk factors can be identified by using:

- checklists
- brainstorming
- SWOT analysis
- expert judgment
- assumptions analysis
- constraints analysis
- lessons-learned files
- cost modeling
- schedule analysis
- requirements triage
- assets inventory
- tradeoff analysis

SWOT: Strengths, Weaknesses, Opportunities, Threats

A Checklist: The SEI Risk Taxonomy

- SEI has developed a three-level taxonomy of risk factors for software projects
- The top level of the taxonomy has three major categories:
  - Product Engineering
    - the technical aspects of the work to be done
  - Development Environment
    - the methods, procedures, and tools to be used
  - Program Constraints
    - contractual, organizational, and operational factors that are outside the control of local management

http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf

The SEI Risk Taxonomy, Second Level

A. Product Engineering Risk
   1. Requirements
   2. Design
   3. Code and Unit Test
   4. Integration and Test
   5. Engineering Specialties

B. Development Environment
   1. Development Process
   2. Development System
   3. Management Process
   4. Management Methods
   5. Work Environment

C. Program Constraints
   1. Resources
   2. Contract
   3. Program Interfaces

The SEI Risk Taxonomy for Product Engineering (3rd Level)

A.1. Requirements
- a. Stability
- b. Completeness
- c. Clarity
- d. Validity
- e. Feasibility
- f. Precedent
- g. Scale

A.2 Design
- a. Functionality
- b. Difficulty
- c. Interfaces
- d. Performance
- e. Testability
- f. Hardware Constraints
- g. Non-Developmental Software

A.3 Code and Unit Test
- a. Feasibility
- b. Testing
- c. Coding/Implementation

A.4 Integration and Test
- a. Environment
- b. Product
- c. System

A.5 Engineering Specialties
- a. Maintainability
- b. Reliability
- c. Safety
- d. Security
- e. Human Factors
- f. Specifications

The SEI Risk Taxonomy for Development Environment (3^{rd} level)

B.1 Development Process

- a. Formality
- b. Suitability
- c. Process Control
- d. Familiarity
- e. Product Control

B.2 Development System

- a. Capacity
- b. Suitability
- c. Usability
- d. Familiarity
- e. Reliability
- f. System Support
- g. Deliverability

B.3 Management Process

- a. Planning
- b. Project Organization
- c. Management Experience
- d. Program Interfaces

B.4 Management Methods

- a. Monitoring
- b. Personnel Management
- c. Quality Assurance
- d. Configuration Management

B.5 Work Environment

- a. Quality Attitude
- b. Cooperation
- c. Communication
- d. Morale

The SEI Risk Taxonomy for Program Constraints (3rd Level)

C.1 Resources
- a. Schedule
- b. Staff
- c. Budget
- d. Facilities

C.2 Contract
- a. Type of Contract
- b. Restrictions
- c. Dependencies

C.3 Program Interfaces
- a. Customer
- b. Associate Contractors
- c. Subcontractors
- d. Prime Contractor
- e. Corporate Management
- f. Vendors
- g. Politics

Accessing SEI Risk Management Information

- The SEI Report “Taxonomy-Based Risk Identification” contains the taxonomy and lots of other information
- It can be found at:

  http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf

- Several related reports are on the SEI Web site

Using Work Packages to Identify Risk Factors

- Each work package includes an entry for risk factors:

  Activity : 3.2.2.1 DESIGN_PROCESSOR_COMPONENT
  Activity description: Specify internal architecture of the Trans. Processor
  Estimated duration: 3 weeks
  Resources needed:
  Personnel: 2 senior telecom designers
  Skills: Designers must know the X25 protocol
  Tools: One Sun workstation running Statemate
  Travel: 3 day Design Review in San Diego for 2 people
  Predecessor tasks: 3.2.1 - Develop ATM system architecture
  Successor tasks: 3.3.2.2 - Implement Processor
  Work Products: Architectural specification for Processor
  Test plan for Processor
  Baselines: Architectural specification for Processor
  Test plan for Processor
  Completion criteria: design inspection by peers and
  approval of Processor design by the Chief Architect
  Risks: senior designers not identified

Quantifying and Prioritizing Risk Factors

- Risk factors can be quantified and prioritized using
  - Risk exposure
  - Risk leverage factors

Risk Exposure (1)

- Risk exposure is the product of

  PROBABILITY x POTENTIAL LOSS

- A project with 30% probability of late delivery and a penalty of $100,000 for late delivery has a risk exposure of:

  0.3 x 100,000 = $30,000

- QUESTION: what risk to quality is created by delivering the product on time but with poor reliability?
- OBSERVATION: reducing one risk factor may cause an increase in another (perhaps worse) one

Risk Exposure (2)

- What to do when we cannot quantify our risk factors? (use Ordinal scales: LOW, MEDIUM, HIGH)

IMPACT: PROB:	LOW	MED	HIGH
LOW	LOW	MED	HIGH
MED	MED	MED	HIGH
HIGH	HIGH	HIGH	VERY HIGH

Using Ordinal Scales

- How do you determine “Low,” “Medium,” and “High” ?
  - o Expert judgment
  - o Group consensus
  - o Historical data

Risk Exposure (4)

- How do you balance a LOW probability, HIGH impact risk factor against a HIGH probability, LOW impact risk factor?
- Is a (0.25 x 75) risk the same as a (0.75 x 25) risk?
  - both have the same risk exposure

How Much Should We Spend to Mitigate a Risk Factor?

- Clearly, we should not spend more than the cost of the potential problem
  - don’t spend $200,000 to avoid a $100,000 problem that might not happen
- We can calculate the Risk Leverage Factor (RLF) to provide some guidance:

$$RLF = (REb - REa) / RMc$$

- where

REb is the risk exposure before risk mitigation,

REa is the risk exposure after risk mitigation and

RMc is the cost of the risk mitigating actions

A Risk Leverage Calculation

- Suppose we are considering spending $25,000 to reduce the probability of a risk factor with potential impact of $500,000 from 0.4 to 0.1
- then the RLF is:

  (200,000 - 50,000) / 25,000 = 6.0
- Larger RLFs indicate better investment strategies
  - RLFs can be used to prioritize risk
  - and determine mitigation strategies
- Spending another $100,000 to further reduce the probability might not be a good investment

The “Cost” of Risk Mitigation

- “Clearly, we should not spend more than the cost of the potential problem”
  - however, the “cost” may include psychological and subjective factors
- The U.S. has spent much more to reduce the risk of terrorist attacks and space shuttle accidents than the $$ cost of a few attacks and accidents

Risk Evaluation

- Evaluating and prioritizing risk factors ultimately requires subjective judgment
- Numerical calculations are useful as guidelines, but they are only guidelines
- Humans must make risk management decisions

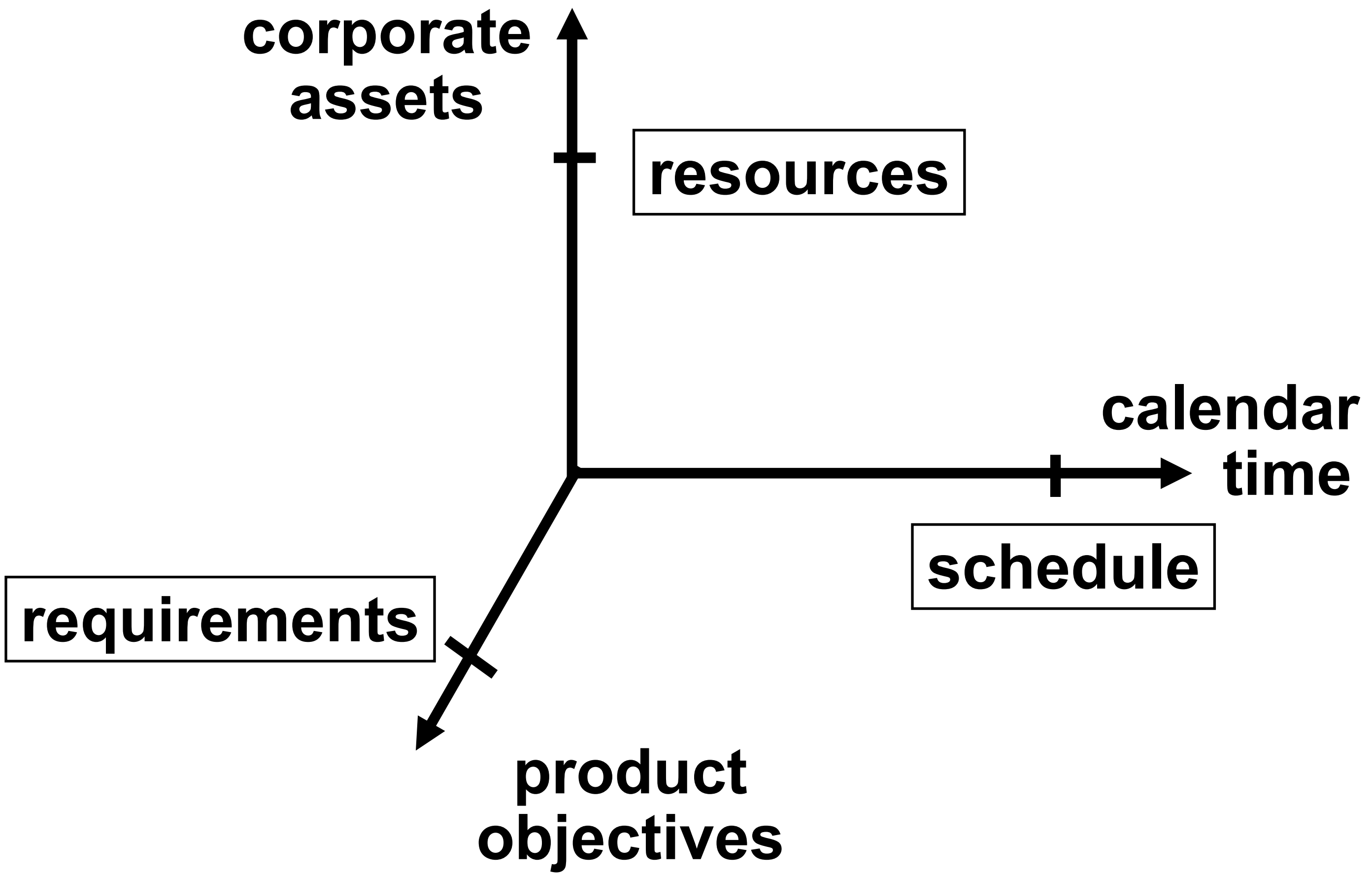
A Risk Management Paradigm (1)

An effective model of project risk must account for the following factors:

- Cost
- Schedule
- Project Objectives
  - Product Features
  - Quality Requirements
- Resources
- Constraints
- Tradeoffs

and the non-linear relationships among these factors

A Risk Management Paradigm (2)



corporate assets

resources

calendar time

schedule

requirements

product objectives

Assets

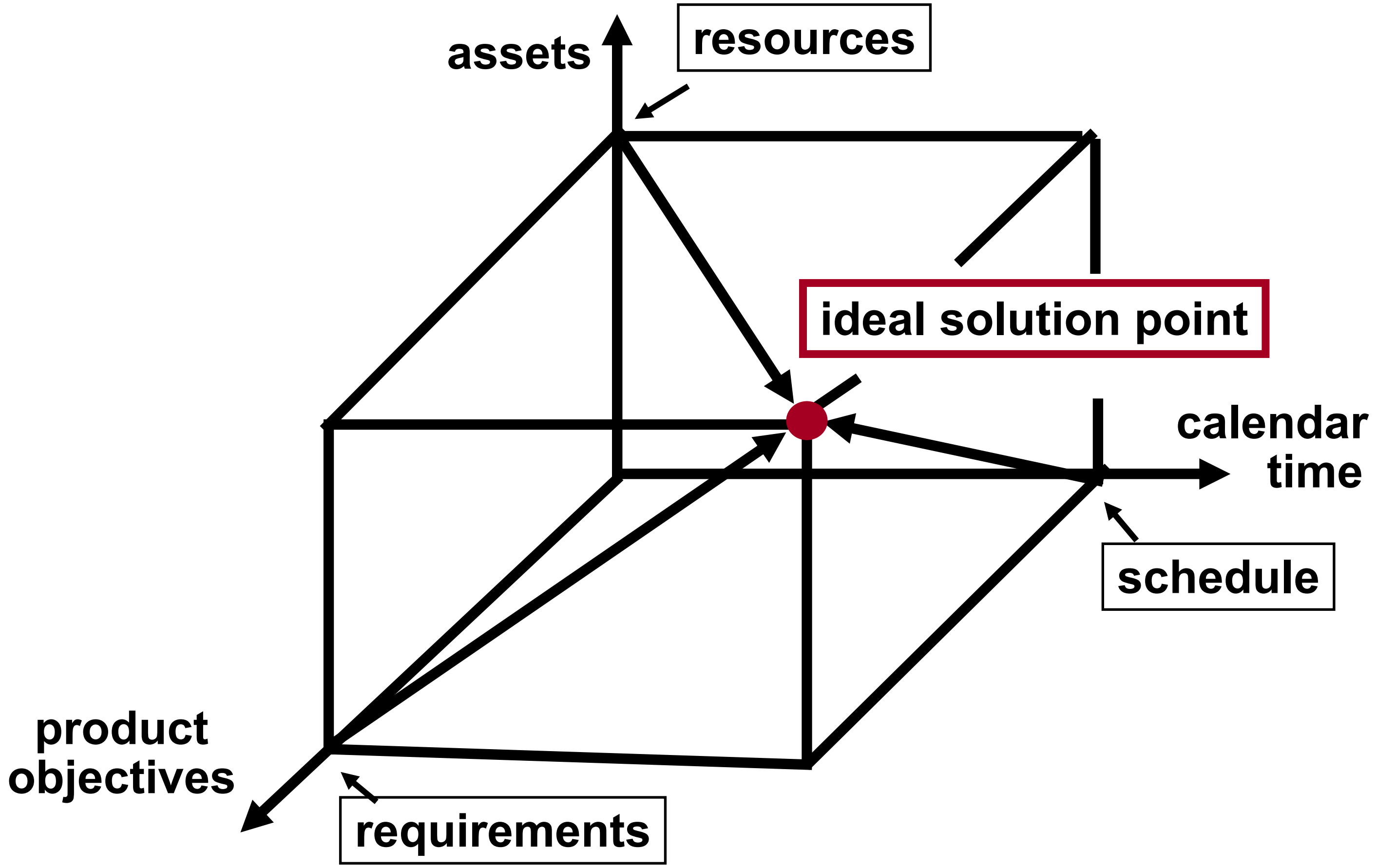
Assets include:

- Personnel:
    - numbers, skills, availability
- Development Environment:
    - methods and tools
- Target Environment:
    - speed and size
- Supporting Functions:
    - CM, QA, publications
- Process Model:
    - waterfall, incremental, evolutionary

Kinds of Project Objectives

- Functional Requirements
  - features to be provided by the software
- Performance Requirements
  - sample rates, throughput, response
- Capacities
  - memory, bandwidth, processor speed
- Design Constraints
  - standards, programming language, resource limits, host and target machines
- External Interfaces
  - interactions with people, other software, and hardware
- Quality Attributes
  - usability, portability, maintainability, safety, security, reliability

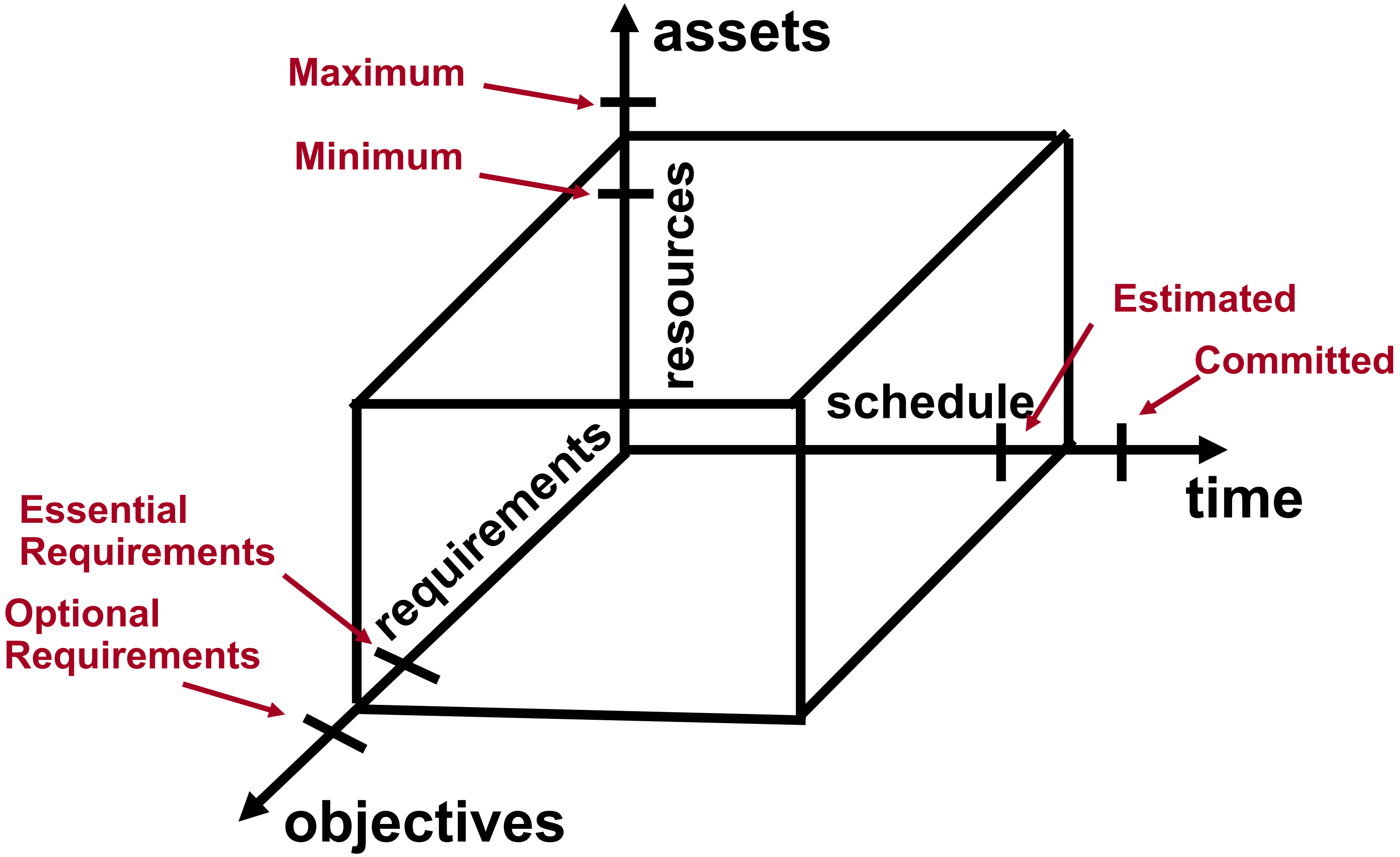
The Solution Space



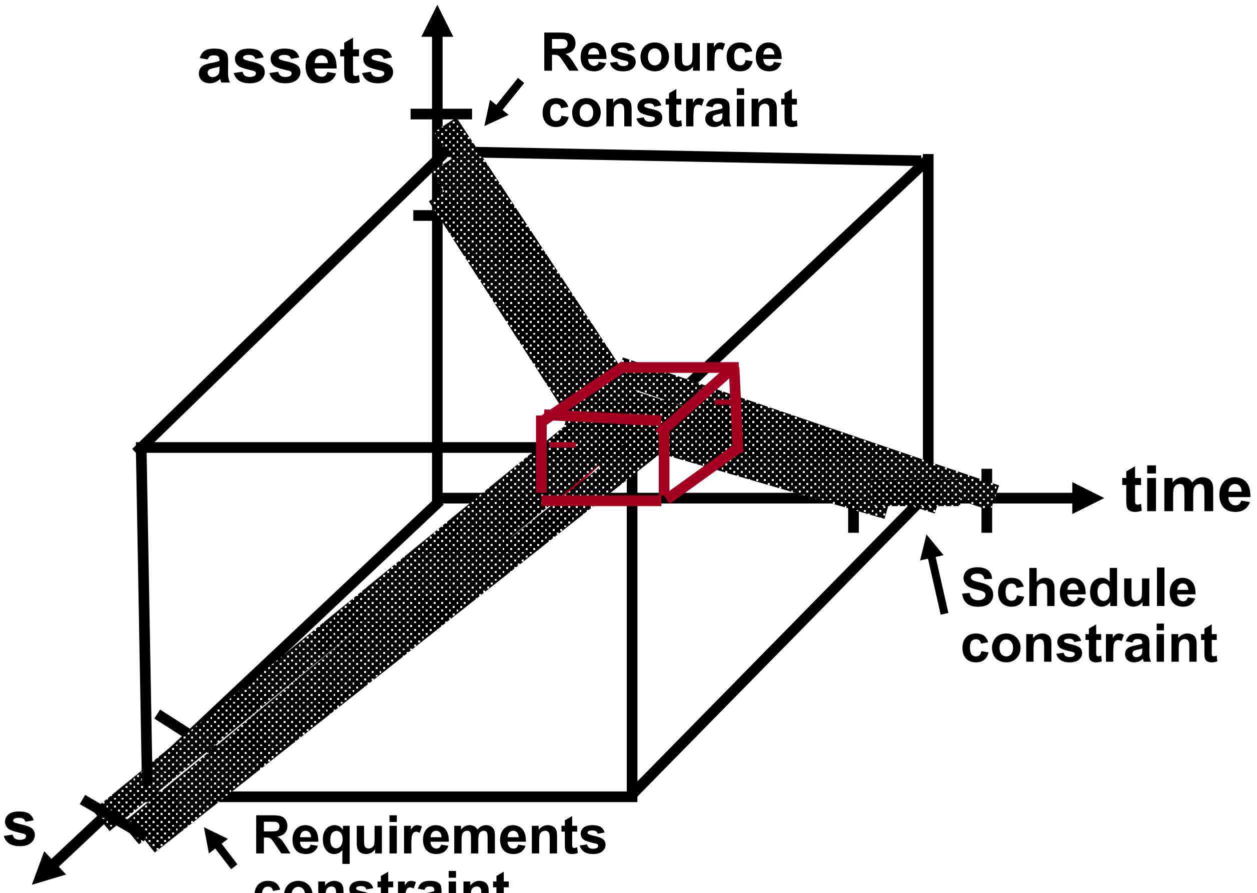
Constraints

- Constraints are externally imposed conditions on Schedule, Budget, Assets, and Objectives:
  - Schedule: the product must be ready for the trade show in 9 months
  - Budget: the department budget is over-committed; we must do the project for $50K or less
  - Assets: the 5 people available to work this project are not outstanding; we have 3 workstations and no dedicated office space is available
  - Objectives: marketing has pre-announced more features than engineering can build in the time available

A Constrained Paradigm



THE RISK MANAGEMENT PARADIGM



Resource constraint

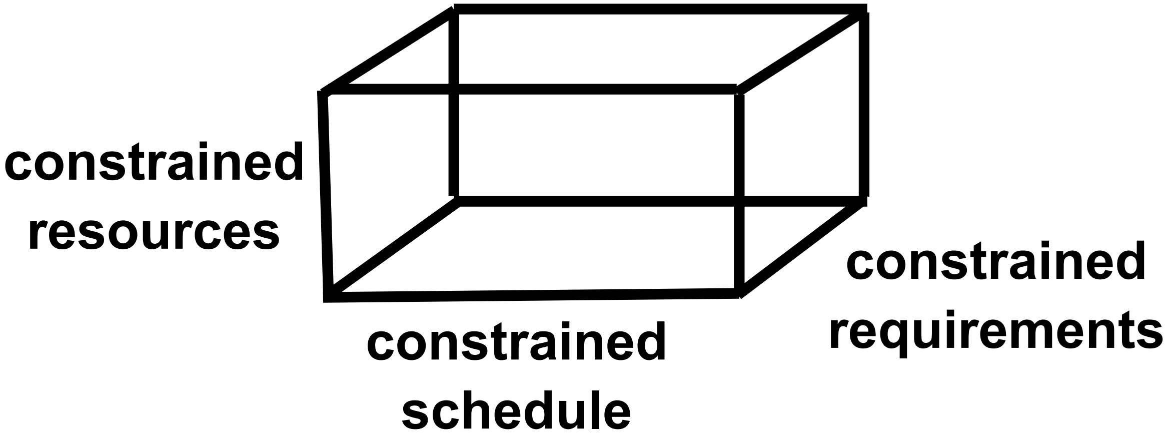
Schedule constraint

Requirements constraint

The goal of risk management is to keep a project in the constraints box

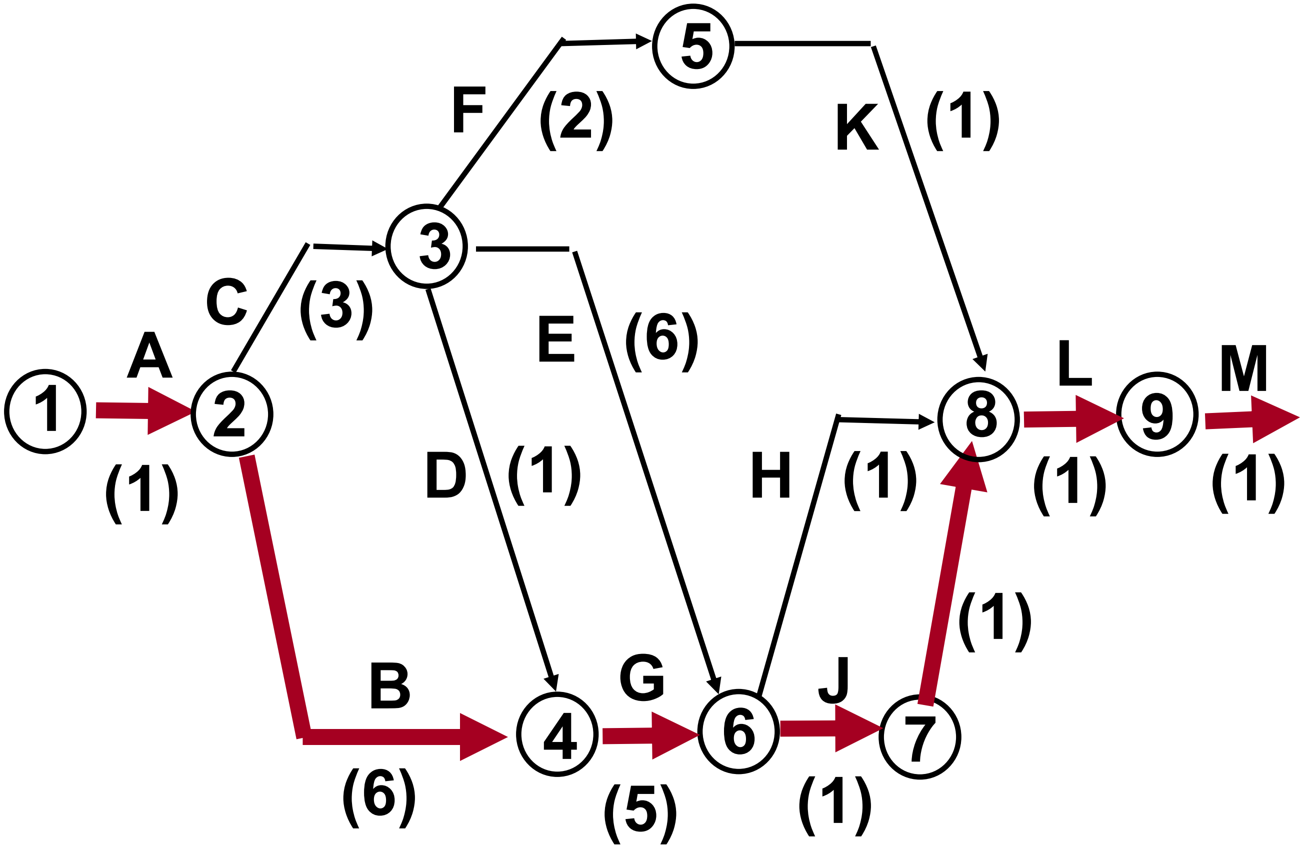
Constraints and Uncertainty (1)

Constraints and uncertainty in resources, requirements, and schedule are the root causes of project risk:



- Uncertainty results from lack of information
- An overly constrained project has no constraint space

Analyzing Schedule Risk



The PERT Technique for Determining Schedule Risk

- In a PERT network, three estimates of duration are provided for each activity:

  m - the most likely duration

  a - the most optimistic estimate

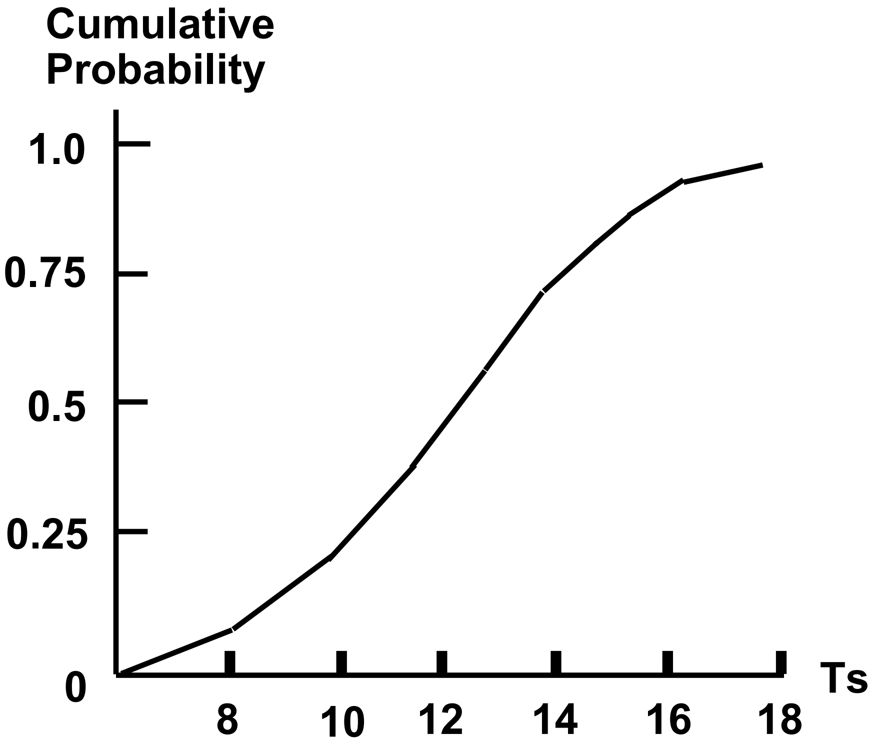
  b - the most pessimistic estimate

- A probability distribution for each activity duration is constructed from (a,m,b):

  $E = (a+4m+b) / 6; \qquad s = (b-a) / 3.2$

- The activity durations are aggregated into an overall probability distribution for project completion

A Cumulative PERT
Probability Distribution for a Schedule

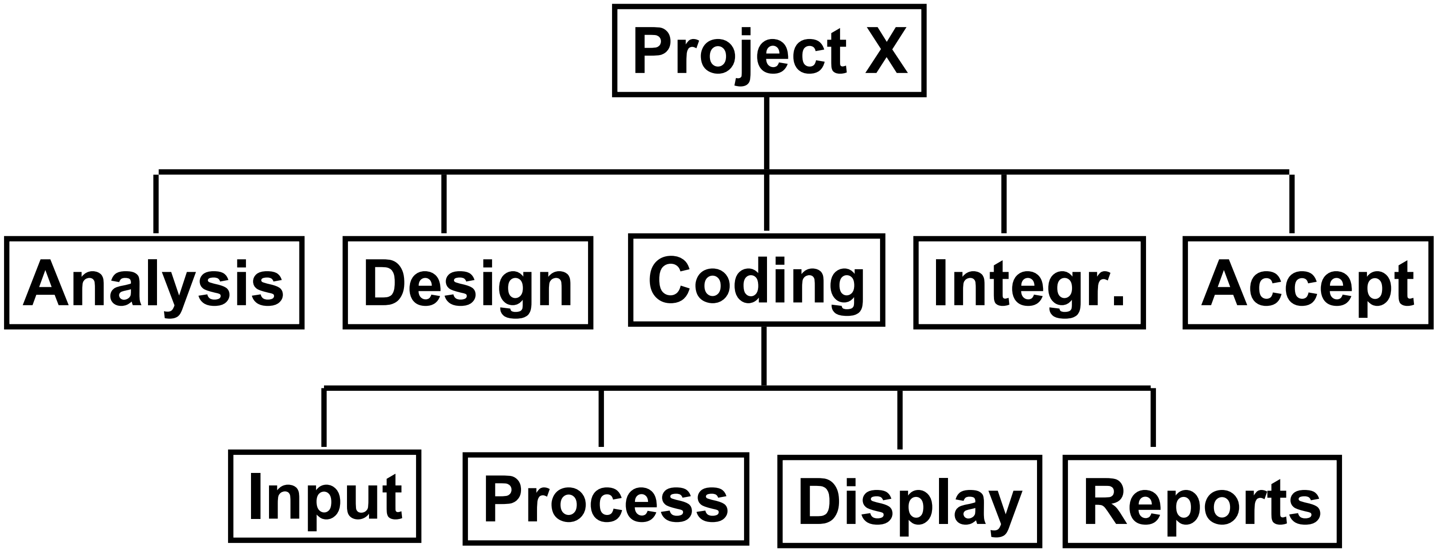


Probability of Completion in time t ≤ Ts
(probability of achieving milestone 10)

Additional Risk Considerations for Activity Networks

- Nodes with a high degree of "fan-out"
- Nodes with a high degree of "fan-in"
- Paths that are "almost critical"

Using a WBS to Determine Size and/or Effort Risk



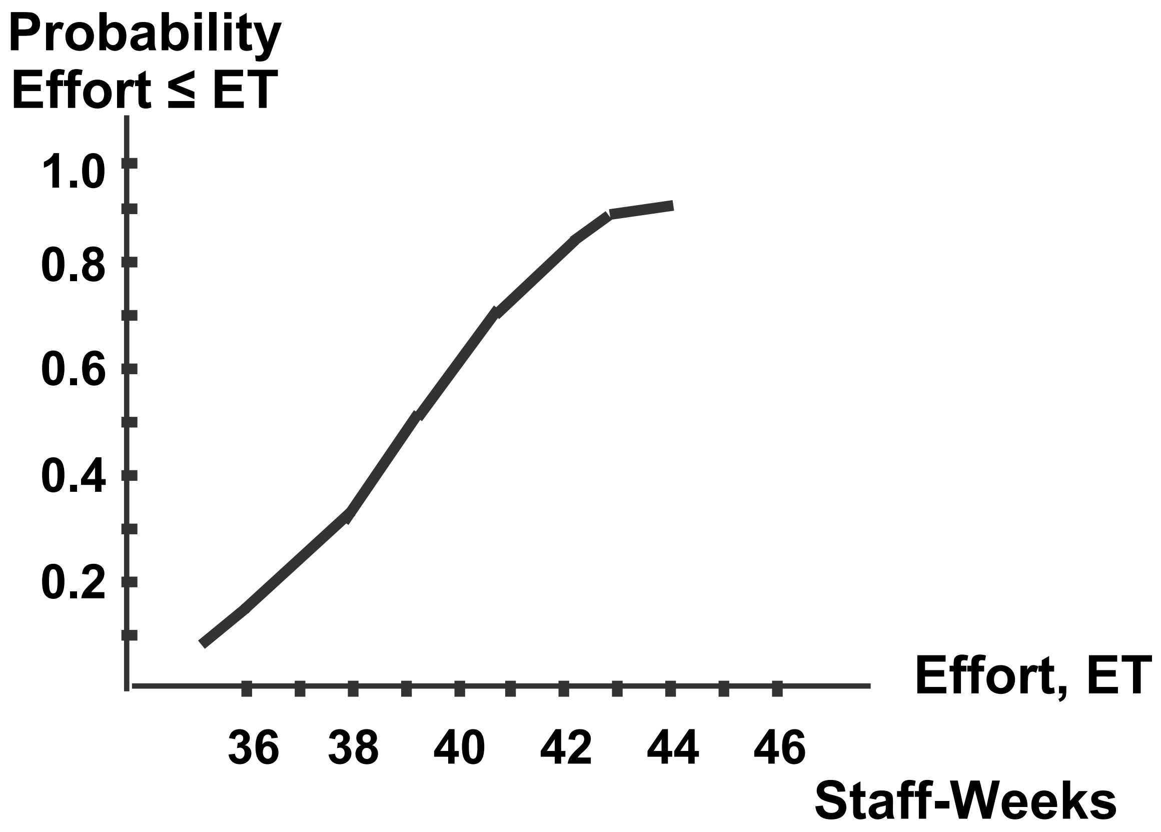
Determining Effort Risk

Component	a_i	m_i	b_i	E_i	σ_i
INPUT	6.0	10.0	20.0	11.0	2.33
PROCESS	4.0	7.0	13.0	7.5	1.5
DISPLAY	8.0	12.0	19.0	12.5	1.83
REPORTS	4.0	8.0	12.0	8.0	1.33

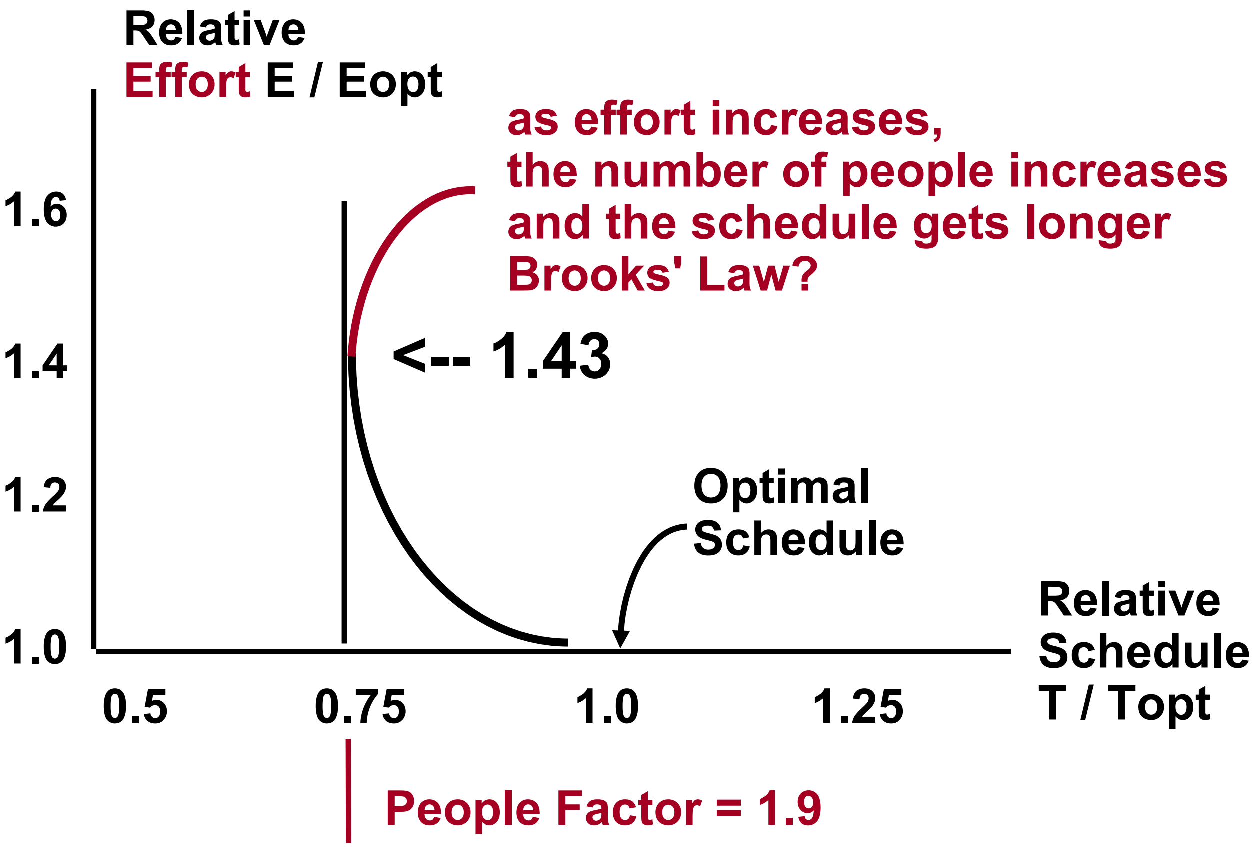
$E = 39$ SW and $\sigma = 3.6$ SW

68% probability range:
$E – \sigma = 35.4$ and $E + \sigma = 42.6$ SW

Cumulative Distribution of Effort Risk



The Risk of Schedule Compression



as effort increases,
the number of people increases
and the schedule gets longer
Brooks' Law?

People Factor = 1.9

Brooks' Law

- Brooks' Law warns of limitations on effort - schedule tradeoffs*

  “Adding manpower to a late software project makes it later.”

* page 25, *The Mythical Man-Month, Anniversary Edition* by Fred Brooks, Addison Wesley 2000

The Risk of Schedule Compression

- The COCOMO II effort multiplier for compressing the schedule by 25% is 1.43
- Suppose we have a 20 person, 24 month project

  effort = 20 x 24 = 480 staff-months
- Suppose we compress the schedule to 18 months; i.e. by 25%

  required effort: 480 x 1.23 = 686 staff-months

  required # people = 590/18 = 38 people

compressing the schedule by 25% requires increasing the number of people by 90% (38/20)

failing to compensate for schedule compression by adding enough people increases the risk of project failure

Quality Risk

- When schedule is compressed and projects are inadequately staffed:
  - defects increase exponentially
- This increases the risk of software that might be:
  - unsafe
  - unsecure
  - unreliable
  - unusable
  - unmaintainable

Risk Management Procedures

Risk management procedures include:

1. Risk Identification: what are the risk factors?
2. Impact Analysis: what are the probabilities, and costs?
3. Prioritization: which are most threatening?
4. Risk Mitigation: how to respond?
5. Risk Tracking: when to respond?
6. Recovery Mgmt: working plan B
7. Crisis Mgmt: when risk management fails

Risk Mitigation Strategies (1)

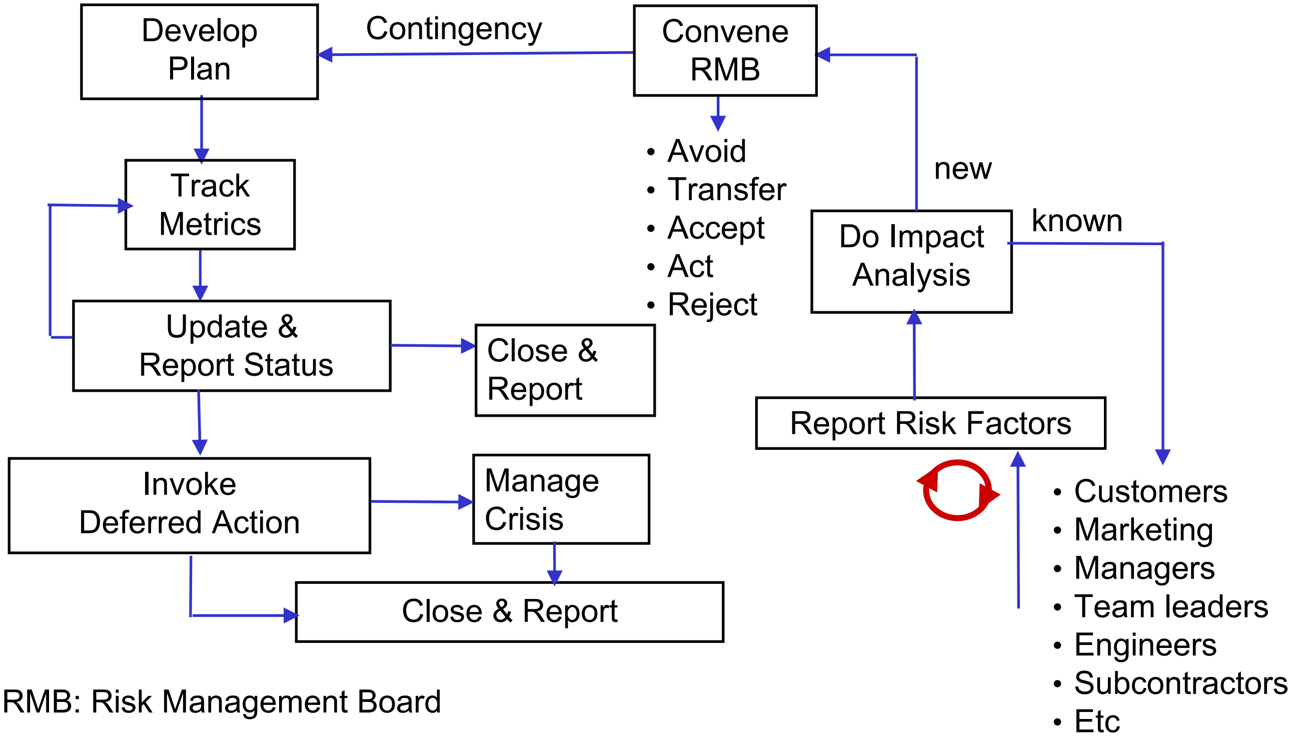
Strategy	**Approach**
Avoidance	Change the project or the product
Transfer	Reallocate requirements
Acceptance	Watch list
Immediate action	Reduce probability and/or impact
Contingent action	Delayed action, if warranted

Risk Mitigation Strategies (2)

- Avoidance
  - o e.g., modify some requirements
- Risk Transfer
  - o e.g., shift the top layers of the communication protocol to the network server
  - o - e.g., subcontract some of the work to specialists
- Risk Acceptance
  - o acknowledge the risk and proceed anyway

Q: when would risk acceptance be a reasonable alternative?

The Risk Management Process



Develop Plan

Contingency

Convene RMB

Track Metrics

- Avoid
- Transfer
- Accept
- Act
- Reject

new

known

Do Impact Analysis

Update & Report Status

Close & Report

Report Risk Factors

Invoke Deferred Action

Manage Crisis

Close & Report

- Customers
- Marketing
- Managers
- Team leaders
- Engineers
- Subcontractors
- Etc

RMB: Risk Management Board

A Format for Reporting Risk Factors

- Risk Report Number
- Submitter (name & contact information)
- Risk Category (schedule, resources, cost, technical, other)
- Severity Level
- Description
- Potential Impact
- Time frame
- Recommended Disposition (avoid, transfer, accept, immediate action, contingent action)

Immediate Action Plan

- Documentation of an immediate-action plan includes:
  - an identifier,
  - the individual who is responsible for seeing that the plan is executed,
  - responsibilities of others involved in implementing the plan
  - a description of the risk factor to be mitigated,
  - the actions to be completed,
  - the resources needed,
  - the planned duration of the indicated actions,
  - the progress milestones to be achieved, and
  - the success criteria that will indicate successful completion of the planned activities.

Format and Example of an Immediate Action Plan

Action Plan Number & Name: AP#3, Java Training

Risk factor to be mitigated: Lack of sufficient Java skill

Actions to be completed: Training class and lab exercise for 20 programmers

Responsible party: Sue Smith

Resources Needed: Instructor, classroom with work stations, release time for attendees

Duration of plan: 4 weeks

Milestones:

Week 1: find instructor, reserve classroom, identify attendees

Week 2: load software on computers, obtain/reproduce class materials

Week 3: conduct 5-day class

Week 4: complete lab project (1/2 time, 4 days)

Success criteria: 19 of 20 attendees successfully complete the lab project

Contingency Planning & Risk Tracking (1)

Contingency planning & risk tracking

- Itemize risk factors:
  - Software Size (256K limit)
  - Execution Time (100 u sec)

List constraints:

  - Schedule and Budget

- Develop alternatives:
  - Build prototype
  - Use memory overlays
  - Use a faster processor
  - Buy more memory
  - Pursue incremental development

Contingency Planning & Risk Tracking (2)

- Evaluate alternatives:
  - o Prototype:

    time and effort required

    how to scale up results?
  - o Memory Overlays

    execution time penalty
  - o Faster Processor

    must use existing processor
  - o Buy Memory:

    architecture won’t support it
  - o Incremental development

    best available approach

Contingency Planning & Risk Tracking (3)

- Selected approach:
  - prioritize requirements
  - pursue incremental development

    (based on prioritized requirements)
  - use Technical Performance Measurement (TPM) on the memory and execution time budgets
- Designate a Responsible Party:
  - Joe Smith

Contingency Tracking Using TPM*

- Identify the critical resource(s) to be tracked
- Allocate the resource(s) to system components
- When a component is satisfactorily completed*, compare the budgeted amount to actual amount:

  AA: Actual Amount of Resource Used

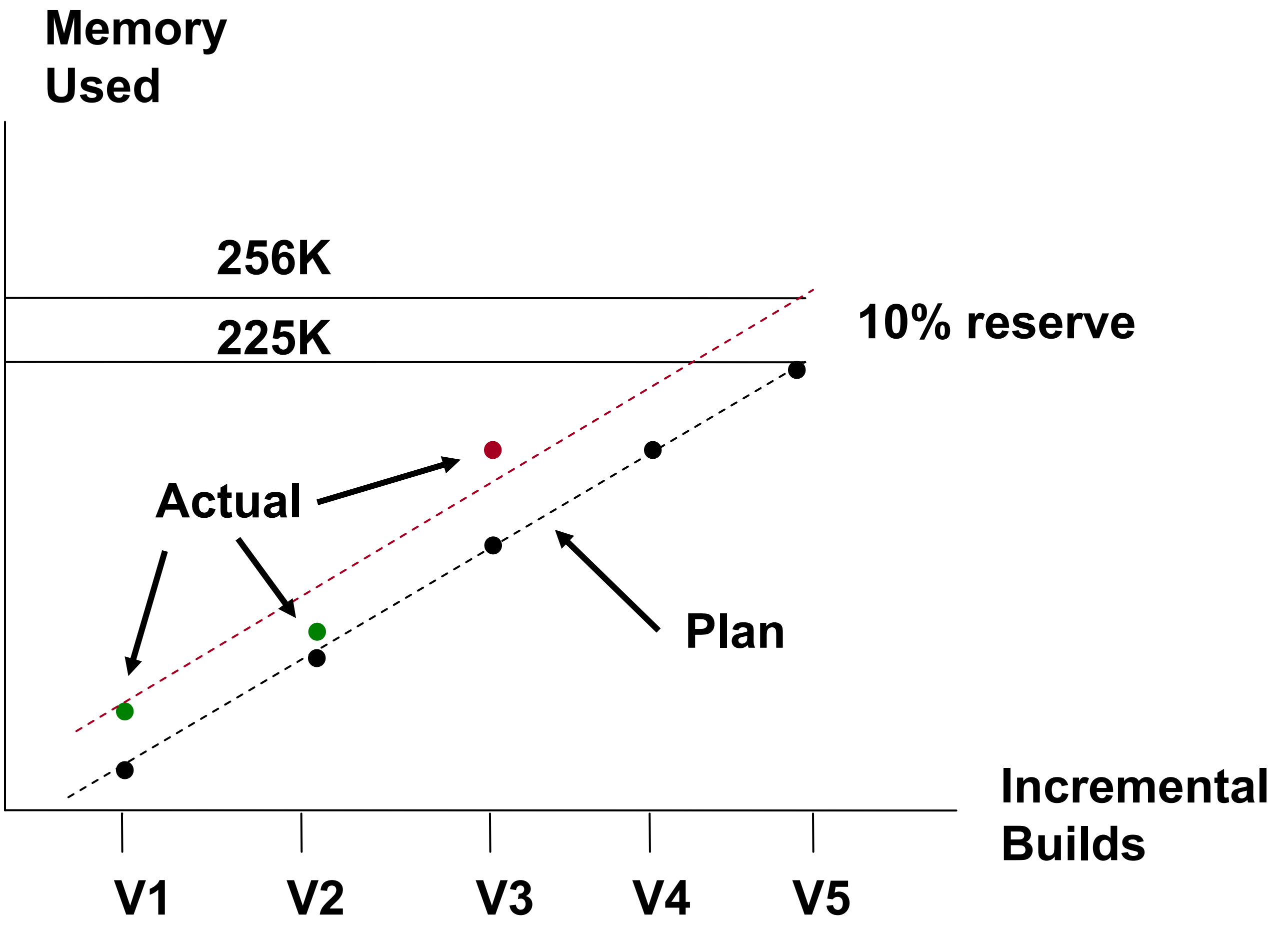
  BA: Budgeted Amount of Resource Planned
- Variance: TPI = Σ (AA - BA) / BA

  TPI: Technical Performance Index

*satisfactory: work package completion criteria

* Technical Performance Measurement

Tracking Memory Usage with TPM



Contingency Plans

- A contingency plan specifies:
  - the risk indicator to be measured,
  - the frequency of measurement,
  - the threshold value for contingent action (i.e., the problem trigger),
  - the contingent-action plan, and
  - the maximum duration for the contingent actions to resolve the problem.
- A project enters crisis mode if the contingent actions do not achieve the success criteria specified in the plan within the specified duration.

Format of a Contingency Plan

Contingency Plan Number & Name: CP #5; Track Memory

Risk factor to be mitigated: Lack of sufficient memory in the microprocessor

Risk indicator to be measured: Planned vs actual memory used in successive incremental builds

Frequency of measurement: weekly measurement of memory usage for weekly incremental builds

Threshold value: 10% over plan on any incremental build

Responsible party: Joe Williams

Contingency Actions:

- *Actions:* Re-engineering of the software to fit within allocated memory
- *Resources:* Unlimited paid overtime for Joe Williams and Sue Smith for a maximum of two weeks
- *Milestones:* no intermediate milestones
- *Success criteria:* Memory usage reduced to not more than 5% over allocated amount

Status Monitoring Techniques to be Used for the Example

- Technical Performance Tracking
- Daily Progress Reports
- Weekly Status Reports
- Demonstrations of Incremental Product Builds

Tracking and Reporting Risk Factors

- Two approaches
  1. Top-N Risk Reporting
  2. Risk Registers

Top-N Risk Tracking (N ≤ 10)

Project:	www					
Date:	xx/yy/zz					
Rank this week	Rank last week	Weeks on list	Risk factor	Potential impact	Current action	Time frame for resolution
1	4	2	Replacement for sensor-control team leader	Delay in completion of coding; lower quality code	Meeting with dept mgr Monday	Immediate
2	6	1	Requested changes in the user interface	Delayed delivery date	Assigned 2 more people	Must complete changes by next Friday
3	2	5	Compiler bugs	Delay in completing the coding of hardware drivers	Validation tests in progress	New release must be validated by this Friday

Top-N Risk Tracking

- Different Top-N lists are prepared and updated at different levels of the organization
  - appropriate stakeholders work together to generate consensus lists
- The lists are updated frequently
- Top-N lists are publicly displayed

Risk Register (1)

- A risk register contains the following information for each identified risk factor
  - Risk factor identifier
  - Revision number & revision date
  - Responsible party
  - Risk category (schedule, resources, cost, technical, other)
  - Description
  - Status (Closed, Action, Monitor)

continued on next chart

Risk Register (2)

- If closed: date of closure and disposition

  (disposition: avoided, transferred, removed from watch list, immediate action or contingent action completed, crisis managed)

- If active: action plan number of contingency plan number & status of the action)

  (status: on plan; or deviating from plan and risk factors for completing the plan)

continued on next chart

Risk Register (3)

- If monitored:
  - Top N rank
  - Previous rank
  - weeks on list
  - potential impact
  - current action
  - time frame for resolution
  - relationship to other risk factors
  - related contingency plan

Crisis Situations

A crisis is a "show-stopper"

Examples:

- the system crashes every 15 minutes
- a key employee has just quit and his design in not written down
- the delivered system does not meet its performance requirements
- marketing has made a major change to requirements
- "Plan B" for the memory budget didn't work

How Do Projects get into Crisis Mode?

Projects get into crisis in the following ways:

1. Lack of attention to risk management
2. An unanticipated situation
3. A foreseen, but deferred, situation
4. A failed contingency plan

Crisis Management Procedures (1)

1. Announce / publicize the problem:

- We have implemented 50% of the functions; the memory budget is overrun by 15%;

- Plan B has not fixed the problem

2. Assign responsibilities and authorities

- Williams, Jones, Smith, and Fairley are to stop all other work and concentrate on this problem

- Fairley will lead the team and will have access to all necessary project resources, subject to coordination with the project manager

Crisis Management Procedures (2)

3. Update status frequently

- standup strategy meetings will be held at 11:00 AM and 6:00 PM until further notice

4. Relax resource constraints

- all necessary project resources are dedicated to solving this problem;

- two additional target machines will be flown in from San Jose;

- meals will be catered;

- sleeping facilities will be established in Room 319

Crisis Management Procedures (3)

5. Operate in Burnout mode

- all project personnel are requested to be on call 24 hours per day until further notice

6. Establish a "drop-dead" date

- this effort will continue *for a maximum of 30 days*;

- if the problem is not solved by then the project will undergo major re-evaluation by marketing and upper level management

7. Stay out of the way

- all personnel not assigned to this effort are requested to continue with their normal work activities and to be available on an as-needed basis

Crisis Recovery (1)

1. Conduct a crisis post-mortem
   - assess damage
   - fix any systemic problems
2. Calculate cost-to-complete the project, or cost to cancel the project
3. Update plans, schedules, and work assignments
4. Time-compensate workers for extraordinary efforts

Crisis Recovery (2)

5. Provide recognition to outstanding performers (and their families):
   - letters of appreciation
   - dinners
   - bonuses
   - time off
   - conference travel
   - new workstations
   - . . . .

Risk Management at the Organization Level (1)

- Factors that result in successful risk management at the organizational level include:
  - explicit definition of development and management practices to be tailored to each project
  - communication based on risk management
  - risk reporting to senior managers

Risk Management at the Organization Level (2)

- A corporate policy for risk management of software projects includes:
  - risk management plans are to be developed at the planning stage of each project and incorporated into the overall project plan
  - project-specific tailoring of the development process and the risk management methods, tools, and techniques to be used on each project
  - explicit review of risk factors on a regular, on-going basis.

Joint Risk Management (1)

- Some risk factors may require mitigation strategies that involve an external customer or a subcontractor,
- For example,
  - reducing risk factors created by ambiguous operational requirements may require greater involvement of representative users;
  - mitigation of technical risk factors may require increased allocation of resources (i.e., money) from the customer, increased involvement with a hardware group, or de-scoping of the requirements.

Joint Risk Management (2)

- Effective risk management at this level requires a great deal of trust and cooperation between the customer and the supplier (your organization).
  - o On the other hand, if there is no trust or cooperation between acquirer and supplier the project will probably fail in any case.

The Main Points of Chapter 9 (1)

- a risk factor is a potential problem; a problem is a risk factor that has materialized
- the goal of risk management is to identify and mitigate risk factors with sufficient lead time to avoid crisis situations
- risk factors are characterized by probability of occurrence and potential loss
- project risk is the set of risk factors that have the potential to negatively impact a software project
- most standards and guidelines for software project management include risk management as a key process
- conventional project management techniques for planning and estimating, measuring and controlling, and leading and directing software projects are institutionalized techniques used to manage generic risk factors

The Main Points of Chapter 9 (2)

- risk management techniques are used to identify, analyze, prioritize, and mitigate project-specific risk factors
- risk mitigation strategies include avoidance, transfer, acceptance, immediate action, and contingent action
- successful risk mitigation reduces the probability and/or the cost of a potential problem to acceptable levels
- managing risk factors in your software projects will be easier, and more likely succeed, if risk management is supported at the organizational level of the organization in which your project is conducted
- you and your customer, and you and your subcontractors (if any) should engage in joint risk management