

Risk Analysis & Management

Source:

1. Roger S. Pressman, Software Engineering – A Practitioner's Approach, 5th Edition, ISBN 0-07-365578-3, McGraw-Hill, 2001 (Chapter 6)
2. Bob Hughes and Mike Cotterell, Software Project Management, 4th edition

Risk management

This lecture will touch upon:

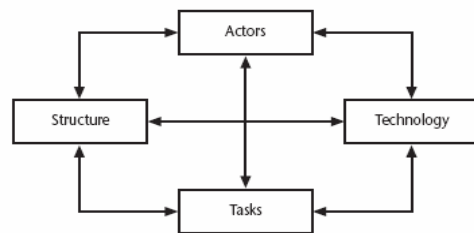
- Definition of 'risk' and 'risk management'
- Some ways of categorizing risk
- Risk management
 - Risk identification – what are the risks to a project?
 - Risk analysis – which ones are really serious?
 - Risk planning – what shall we do?
 - Risk monitoring – has the planning worked?
- We will also look at PERT risk and critical chains

Some definitions of risk

'The chance of exposure to the adverse consequences of future events'

- Project plans have to be based on *assumptions*
- *Risk* is the possibility that an assumption is wrong
- When the risk happens it becomes a *problem* or an *issue*

Categories of risk



Overview

- Risks are potential problems that might affect the successful completion of a software project.
- Risks involve uncertainty and potential losses.
- Risk analysis and management are intended to help a software team understand and manage uncertainty during the development process.
- The important thing is to remember that things can go wrong and to make plans to minimize their impact when they do.
- The work product is called a Risk Mitigation, Monitoring, and Management Plan (RMMM).

Risk Component & 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*

Risk Management

Reactive

- project team reacts to risks when they occur
- mitigation—plan for additional resources in anticipation of fire fighting
- fix on failure—resource are found and applied when the risk strikes
- crisis management—failure does not respond to applied resources and project is in jeopardy

Proactive

- formal risk analysis is performed
- organization corrects the root causes of risk
 - TQM concepts and statistical SQA
- examining risk sources that lie beyond the bounds of the software
- developing the skill to manage change

Risk Check List

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

Risk Projection & Building a Risk Table

- 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:
 - establish a scale that reflects the perceived likelihood of a risk,
 - delineate the consequences of the risk,
 - estimate the impact of the risk on the project and the product, and
 - note the overall accuracy of the risk projection so that there will be no misunderstandings.

Impact Assessment – table 1

| 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 unapproachable software | Significant financial shortages, budget overrun likely | Unachievable ICOC |
| 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 ICOC |
| 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 \$4K 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 overrun | Early achievable ICOC |

Note: (1) The potential consequence of undetected software errors or faults.

(2) The potential consequence if the desired outcome is not achieved.

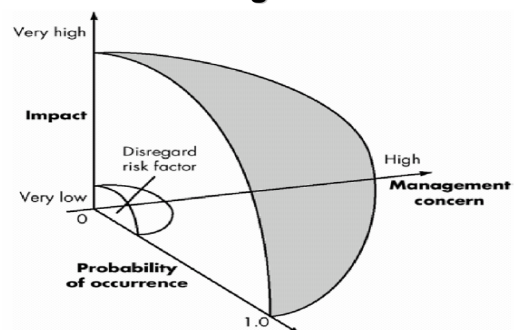
Building Risk Table – table 2

| 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:
1—catastrophic
2—critical
3—marginal
4—negligible

RMMM = Risk Mitigation, Monitoring and Management Plan

Risk and Management Concern



Assessing Risk Impact

The following steps are recommended to determine the overall consequences of a risk:

1. Determine the average probability of occurrence value for each risk component.
2. Using table 1 (slide 10) determine the impact for each component based on the criteria shown.
3. Complete the risk table and analyze the results

The overall *risk exposure*, RE, is determined using the following relationship:

$$RE = P \times C$$

where P is the probability of occurrence for a risk, and C is the cost to the project should the risk occur.



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).

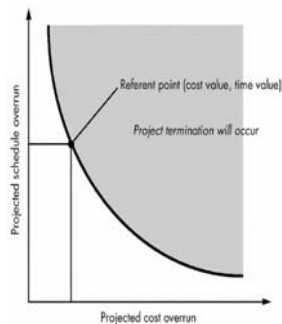
Risk impact. 60 reusable software components were planned. If only 70 percent can be used, 18 components would have to be developed from scratch (in addition to other custom software that has been scheduled for development). Since the average component is 100 LOC and local data indicate that the software engineering cost for each LOC is \$14.00, the overall cost (impact) to develop the components would be $18 \times 100 \times 14 = \$25,200$.

Risk exposure. $RE = 0.80 \times 25,200 = \$20,200$.



Risk Assessment

- For assessment to be useful, a *risk referent level* must be defined.
- In the context of software risk analysis, a risk referent level has a single point, called the *referent point* or *break point*, at which the decision to proceed with the project or terminate it (problems are just too great) are equally weighted.



In reality, the referent level can rarely be represented as a smooth line on a graph. In most cases it is a region in which there are areas of uncertainty; that is, attempting to predict a management decision based on the combination of referent values is often impossible. Therefore, during risk assessment, we perform the following steps:

1. Define the risk referent levels for the project.
2. Attempt to develop a relationship between each (r_i , l_i , x_i) and each of the referent levels. (where r_i = risk, l_i = probability of the risk, and x_i = impact of the risk)
3. Predict the set of referent points that define a region of termination, bounded by a curve or areas of uncertainty.
4. Try to predict how compound combinations of risks will affect a referent level.

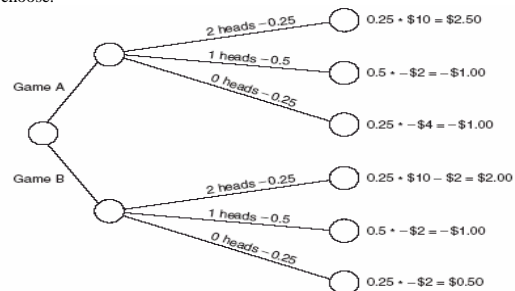


Risk Decision Tree

- A technique that can be used to visualize the risks of alternatives is to build a risk decision tree.
- The top-level branch splits based on the alternatives available. The next split is based on the probabilities of events happening. Each leaf node has the risk exposure for that event. The sum of risks exposure for all leafs under a top-level split gives the total risk exposure for that choice.
- **Example** – A friend offers to play one of two betting games with you. Game A is that you toss a coin twice. He pays you \$10 if you get two heads. You pay him \$2 for each tail you toss. Game B is that you also toss a coin twice, but it costs you \$2 to play and he pays you \$10 if you get two heads. Which game should you play?



The risk decision tree is shown below. Both games total to \$0.50. Thus, each time you play, your average gain is 50cents. No matter which game you choose.



A framework for dealing with risk

The planning for risk includes these steps:

- Risk identification – what risks might there be?
 - Risk analysis and prioritization – which are the most serious risks?
 - Risk planning – what are we going to do about them?
- Risk monitoring – what is the current state of the risk?

Risk identification

Approaches to identifying risks include:

- Use of checklists – usually based on the experience of past projects (previous slides)
- Brainstorming – getting knowledgeable stakeholders together to pool concerns
- Causal mapping – identifying possible chains of cause and effect

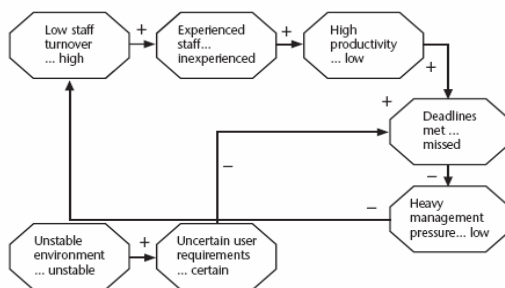
Boehm's top 10 development risks

| Risk | Risk reduction techniques |
|---|--|
| Personnel shortfalls | Staffing with top talent; job matching; teambuilding; training and career development; early scheduling of key personnel |
| Unrealistic time and cost estimates | Multiple estimation techniques; design to cost; incremental development; recording and analysis of past projects; standardization of methods |
| Developing the wrong software functions | Improved software evaluation; formal specification methods; user surveys; prototyping; early user manuals |
| Developing the wrong user interface | Prototyping; task analysis; user involvement |

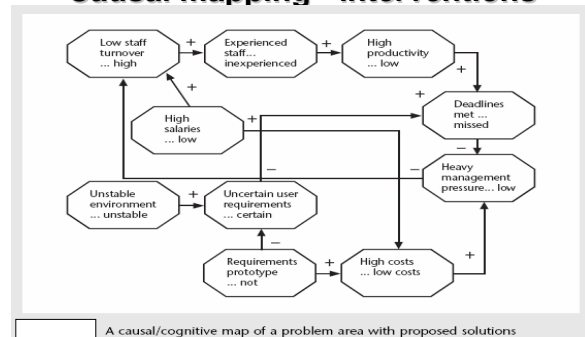
Boehm's top ten risk - continued

| | |
|--|--|
| Gold plating | Requirements scrubbing, prototyping, design to cost |
| Late changes to requirements | Change control, incremental development |
| Shortfalls in externally supplied components | Benchmarking, inspections, formal specifications, contractual agreements, quality controls |
| Shortfalls in externally performed tasks | Quality assurance procedures, competitive design etc |
| Real time performance problems | Simulation, prototyping, tuning |
| Development technically too difficult | Technical analysis, cost-benefit analysis, prototyping, training |

Causal mapping



Causal mapping - interventions



Risk prioritization

Risk exposure (RE) = (potential damage) x (probability of occurrence)

Ideally

- **Potential damage:** a money value e.g. a flood would cause \$0.5 millions of damage
- **Probability** 0.00 (absolutely no chance) to 1.00 (absolutely certain) e.g. 0.01 (one in hundred chance)
RE = \$0.5m x 0.01 = \$5,000
- Crudely analogous to the amount needed for an insurance premium

Risk Exposure Example

| Ref | Hazard | Likelihood | Impact | Risk Exposure |
|-----|---|------------|--------|---------------|
| R1 | Changes to requirements specification during coding | 8 | 8 | 64 |
| R2 | Specification takes longer than expected | 3 | 7 | 21 |
| R3 | Significant staff sickness affecting critical path activities | 5 | 7 | 35 |
| R4 | Significant staff sickness affecting non-critical activities | 10 | 3 | 30 |
| R5 | Module coding takes longer than expected | 4 | 5 | 20 |
| R6 | Module testing demonstrates errors or deficiencies in design | 4 | 8 | 32 |

Risk probability: qualitative descriptors

| Probability level | Range |
|-------------------|--------------------------------------|
| High | Greater than 50% chance of happening |
| Significant | 30-50% chance of happening |
| Moderate | 10-29% chance of happening |
| Low | Less than 10% chance of happening |

Qualitative descriptors of impact on cost and associated range values

| Impact level | Range |
|--------------|---|
| High | Greater than 30% above budgeted expenditure |
| Significant | 20 to 29% above budgeted expenditure |
| Moderate | 10 to 19% above budgeted expenditure |
| Low | Within 10% of budgeted expenditure. |

Probability impact matrix – using slide 26

| | | | | | |
|--------|-------------|----------------|------------|-------------|------|
| | | Tolerance line | | | |
| Impact | High | | R6 | | R1 |
| | Significant | | R2, R3, R5 | | |
| | Moderate | | | | R4 |
| | Low | | | | |
| | | Low | Moderate | Significant | High |
| | | Probability | | | |

Risk planning

Risks can be dealt with by:

- Risk acceptance (previous slides)
- Risk avoidance (previous slides)
- Risk reduction
- Risk transfer
- Risk mitigation/contingency measures

Risk reduction leverage

Risk reduction leverage (RRL)
 $= (RE_{\text{before}} - RE_{\text{after}}) / (\text{cost of risk reduction})$

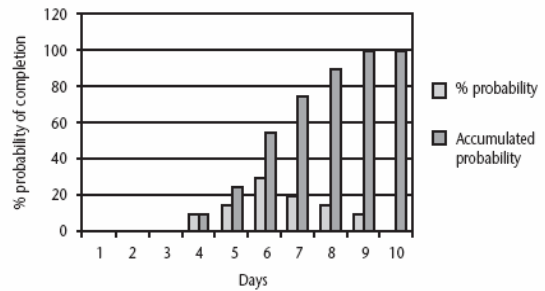
RE_{before} is risk exposure before risk reduction e.g. 1% chance of a fire causing \$200k damage

RE_{after} is risk exposure after risk reduction e.g. fire alarm costing \$500 reduces probability of fire damage to 0.5%

$RRL = (1\% \text{ of } \$200k) - (0.5\% \text{ of } \$200k) / \$500 = 2$

$RRL > 1.00$ therefore worth doing

Probability chart

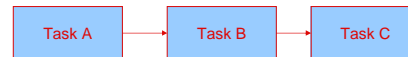


Using PERT to evaluate the effects of uncertainty

Three estimates are produced for each activity

- Most likely time (m)
- Optimistic time (a)
- Pessimistic (b)
- 'expected time' $t_e = (a + 4m + b) / 6$
- 'activity standard deviation' $S = (b-a)/6$

A chain of activities



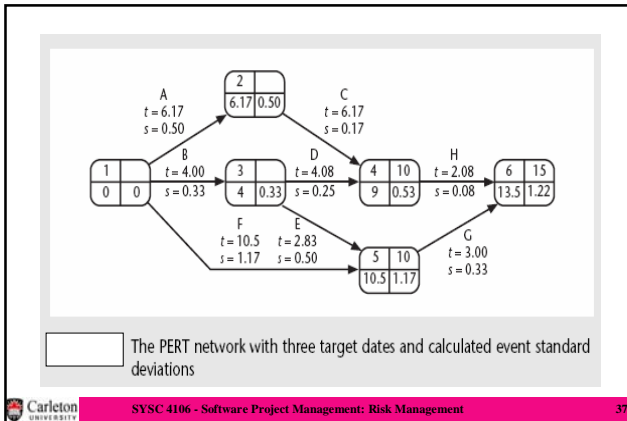
| Task | a | m | b | t_e | s |
|------|----|----|----|-------|---|
| A | 10 | 12 | 16 | ? | ? |
| B | 8 | 10 | 14 | ? | ? |
| C | 20 | 24 | 38 | ? | ? |

A chain of activities

- What would be the expected duration of the chain A + B + C?
- Answer: $12.66 + 10.33 + 25.66$ i.e. 48.65
- What would be the standard deviation for A + B + C?
- Answer: square root of $(1^2 + 1^2 + 3^2)$ i.e. 3.32

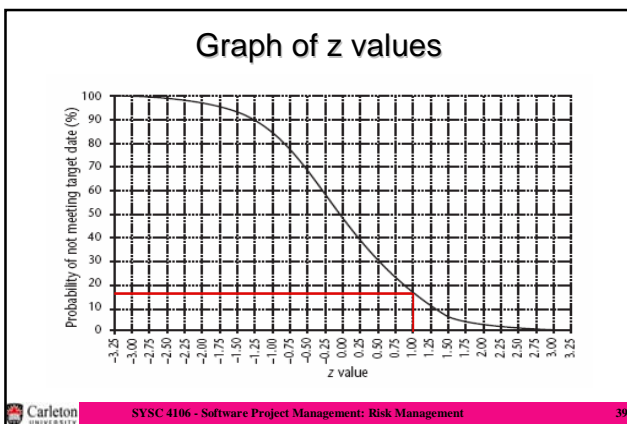
Expected Times and Standard Deviation

| Activity | Optimistic | Most likely | Pessimistic | Expected | Standard deviation |
|----------|------------|-------------|-------------|-----------|--------------------|
| | (a) | (m) | (b) | (t_e) | (s) |
| A | 5 | 6 | 8 | 6.17 | 0.50 |
| B | 3 | 4 | 5 | 4.00 | 0.33 |
| C | 2 | 3 | 3 | 2.83 | 0.17 |
| D | 3.5 | 4 | 5 | 4.08 | 0.25 |
| E | 1 | 3 | 4 | 2.83 | 0.50 |
| F | 8 | 10 | 15 | 10.50 | 1.17 |
| G | 2 | 3 | 4 | 3.00 | 0.33 |
| H | 2 | 2 | 2.5 | 2.08 | 0.08 |



Assessing the likelihood of meeting a target

- Say the target for completing A+B+C was 52 days (T) [from slide 34]
- Calculate the z value thus $z = (T - t_e)/s$
- In this example $z = (52-48.33)/3.32$ i.e. 1.01
- Look up in table of z values – see next overhead



Critical chain approach

One problem with estimates of task duration:

- Estimators add a safety zone to estimate to take account of possible difficulties
- Developers work to the estimate + safety zone, so time is lost
- No advantage is taken of opportunities where tasks can finish early – and provide a buffer for later activities

Critical chain approach

One answer to this:

- Base targets on midpoints (i.e. t_e)
- Accumulate 50% of the safety zones (between t_e and b) into a buffer at the end of the project
- Work backwards and start all activities at their latest start dates
- During project execution use relay race model

Risk Due to Product Size

Attributes that affect risk:

- estimated size of the product in LOC or FP?
- estimated size of product in number of programs, files, transactions?
- percentage deviation in size of product from average for previous products?
- size of database created or used by the product?
- number of users of the product?
- number of projected changes to the requirements for the product? before delivery? after delivery?
- amount of reused software?

Risk Due to Business Impact

Attributes that affect risk:

- effect of this product on company revenue?
- visibility of this product by senior management?
- reasonableness of delivery deadline?
- number of customers who will use this product
- interoperability constraints
- sophistication of end users?
- amount and quality of product documentation that must be produced and delivered to the customer?
- governmental constraints
- costs associated with late delivery?
- costs associated with a defective product?

Risks Due to the Customer

Questions that must be answered:

- Have you worked with the customer in the past?
- Does the customer have a solid idea of requirements?
- Has the customer agreed to spend time with you?
- Is the customer willing to participate in reviews?
- Is the customer technically sophisticated?
- Is the customer willing to let your people do their job—that is, will the customer resist looking over your shoulder during technically detailed work?
- Does the customer understand the software engineering process?

Risks Due to Process Maturity

Questions that must be answered:

- Have you established a common process framework?
- Is it followed by project teams?
- Do you have management support for software engineering
- Do you have a proactive approach to SQA?
- Do you conduct formal technical reviews?
- Are CASE tools used for analysis, design and testing?
- Are the tools integrated with one another?
- Have document formats been established?

Technology Risks

Questions that must be answered:

- Is the technology new to your organization?
- Are new algorithms, I/O technology required?
- Is new or unproven hardware involved?
- Does the application interface with new software?
- Is a specialized user interface required?
- Is the application radically different?
- Are you using new software engineering methods?
- Are you using unconventional software development methods, such as formal methods, AI-based approaches, artificial neural networks?
- Are there significant performance constraints?
- Is there doubt the functionality requested is "do-able?"