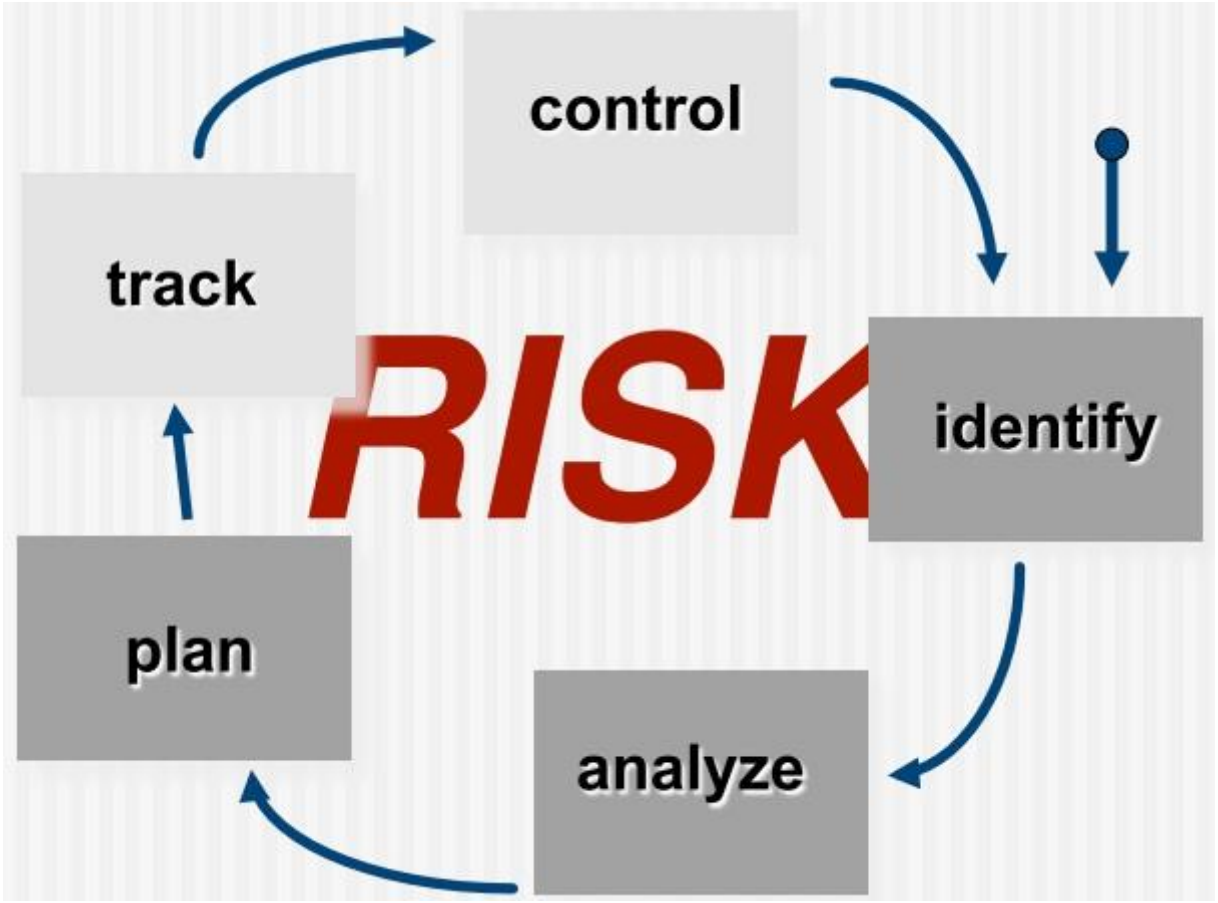
Risk Management

Roger Pressman

- What can go wrong -> Risk Identification
- What is the likelihood -> Risk Analysis
- What will the damage be? -> Risk Analysis
- What can we do about it -> Risk Mitigation

- Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, stakeholder, and requirements problems and their impact on a software project.


control
track
RISK
identify
plan
analyze

Risk Management Strategies

- Reactive
- Proactive

Type of Risks

- Technical Risk
  - If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification, and maintenance problems. In addition, specification ambiguity, technical uncertainty, technical obsolescence, and “leading-edge” technology are also risk factors. Technical risks occur because the problem is harder to solve than you thought it would be.

- Business Risk
  - Candidates for the top five business risks are

1. building an excellent product or system that no one really wants (market risk).
2. building a product that no longer fits into the overall business strategy for the company (strategic risk)
3. building a product that the sales force doesn't understand how to sell (sales risk).
4. losing the support of senior management due to a change in focus or a change in people (management risk).
5. losing budgetary or personnel commitment (budget risks).

Another categorization

- Known risks
  - Are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources (e.g., unrealistic delivery date, lack of documented requirements or software scope, poor development environment).
- 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

- General risks
- Product specific risks

Risk Identification generic checklist

- **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.
- **Stakeholder characteristics**—risks associated with the sophistication of the stakeholders and the developer's ability to communicate with stakeholders 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.

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 its customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end users have realistic expectations?
6. Is the 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?

Risk Components and Drivers

- The U.S. Air Force [AFC88] has published a pamphlet that contains excellent guidelines for software risk identification and abatement. The Air Force approach requires that the project manager identify the risk drivers that affect software risk components:
- Performance
- Cost
- Support
- Schedule.

- *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.

Risk Projection or Risk Estimation

- the likelihood or probability that the risk is real
- the consequences of the problems associated with the risk, should it occur.

Risk Projection

- Perform 4 steps:

1. Establish a scale that reflects the perceived likelihood of a risk.
2. Delineate the consequences of the risk.
3. Estimate the impact of the risk on the project and the product.
4. Assess the overall accuracy of the risk projection so that there will be no misunderstandings.

Risk Impact

- The impact of each risk driver on the risk component is divided into one of four impact categories:
- Negligible
- Marginal
- Critical
- Catastrophic.

Components / Category		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 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	
Σ				
Σ				
Σ				

Impact values:
1—catastrophic
2—critical
3—marginal
4—negligible

Risk Impact Assessment

- **Risk Exposure = Probability of Risk x Cost if the Risk occurs**
- **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 percent (likely).
- **Risk impact**. Sixty 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 = 0.80 x 25,200 ~ $20,200.

Risk Mitigation

- For example, employee turnover:
- Meet with current staff to determine causes for turnover (e.g., poor working conditions, low pay, competitive job market).
- Mitigate those causes that are under your control before the project starts.
- Once the project commences, assume turnover will occur and develop techniques to ensure continuity when people leave.
- Organize project teams so that information about each development activity is widely dispersed.
- Define work product standards and establish mechanisms to be sure that all models and documents are developed in a timely manner.
- Conduct peer reviews of all work (so that more than one person is “up to speed”).
- Assign a backup staff member for every critical technologist.

Table 3: Student Top 10 Risk Items

Risk Item	Risk Management Technique
Overriding other people's work, not having the latest versions of code	Use a configuration management tool effectively.
Lack of exposure to and/or experience with technologies	Take time to learn tools and technologies, seek help from teaching staff.
Being overwhelmed by work in other classes	Have a project management plan with deadlines and ownership, update the project management plan frequently.
Common meeting times	In the beginning of the project, determine all possible common times to meet based on class schedules and other commitments.
Requirements understanding	Meet with, e-mail, or phone customer.
Lack of communication	Set up a group Web page, group e-mail accounts, trade instant messaging IDs, meet regularly.
Project organization	Assign each team member a role, break down work in project management plan.
Loss of a team member	Assure files are uploaded and integrated consistently, use knowledge management strategies such as pair programming to understand each other's work.
Difficulty integrating work	Increase communication, integrate often.
Planning taking up too much time, not enough time to work on product	Don't get more detailed than necessary with the planning.

Ref: http://agile.csc.ncsu.edu/SEMaterials/RiskManagement.pdf

Table 4: Sample Student Risk Table

Rank	Risk	Probability	Impact	Rank Last Week/ Weeks on list	Action
1	None of us knows how to use the technology.	frequent	critical	1/5	Read. Do tutorials.
2	Integration problems.	frequent	critical	2/5	Integrate all work Sunday nights.
3	Someone drops the class.	improb	critical	4/5	Pair programming for all work.
4	Team members missing important team meetings.	improb.	marginal	5/4	Person who misses meeting has to supply Sunday night pizza the next week.
5	Overriding each other's work	improb	marginal	3/5	Continue using CVS.

Ref: http://agile.csc.ncsu.edu/SEMaterials/RiskManagement.pdf

Risk Contingency

- Plan what to do if risk occurs even after all mitigation efforts.

Risk information sheet

Risk ID: P02-4-32	Date: 5/9/09	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 7/1/09.

Current status:
5/12/09: Mitigation steps initiated.

Originator: D. Gagne	Assigned: B. Laster

Boehm's prioritized top-ten list of software risk items:

Risk item	Risk Management techniques
1. Personnel shortfalls	Staffing with top talent, job matching; teambuilding; cross-training; pre-scheduling; key people; morale building
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	High change threshold; information hiding; incremental development (defer changes to later increments)
7. Shortfalls in externally furnished components	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