COURSE NAME

SOFTWARE ENGINEERING

CSC 3114

(UNDERGRADUATE)

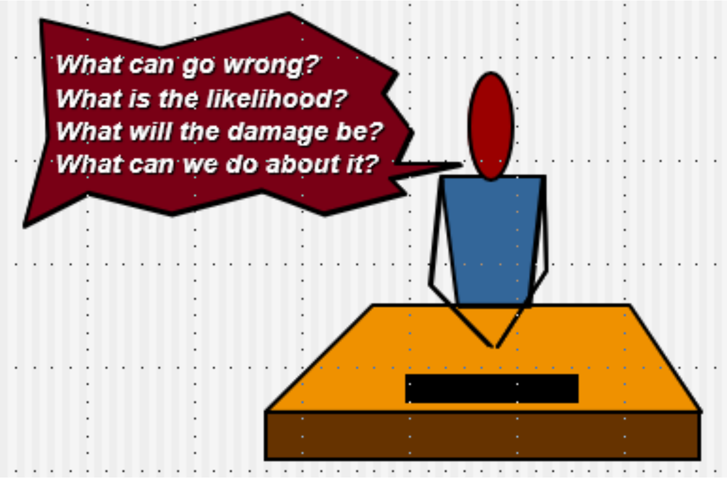
CHAPTER 16

RISK MANAGEMENT

RISK OVERVIEW

The chance of exposure to (introduce) the adverse (opposing) 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*
- 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



RISK MANAGEMENT

Reactive

- project team reacts to risks when they occur
- mitigation—plan for additional resources to reduce the severity of damages
- fix on failure—resources are found and applied when the risk strikes

Proactive

- formal risk analysis is performed
- organization corrects the root causes of risk
  - examining risk sources that lie beyond the bounds of the software (C=A/B)
  - developing the skill to manage change

RISK PROJECTION & BUILDING A RISK TABLE

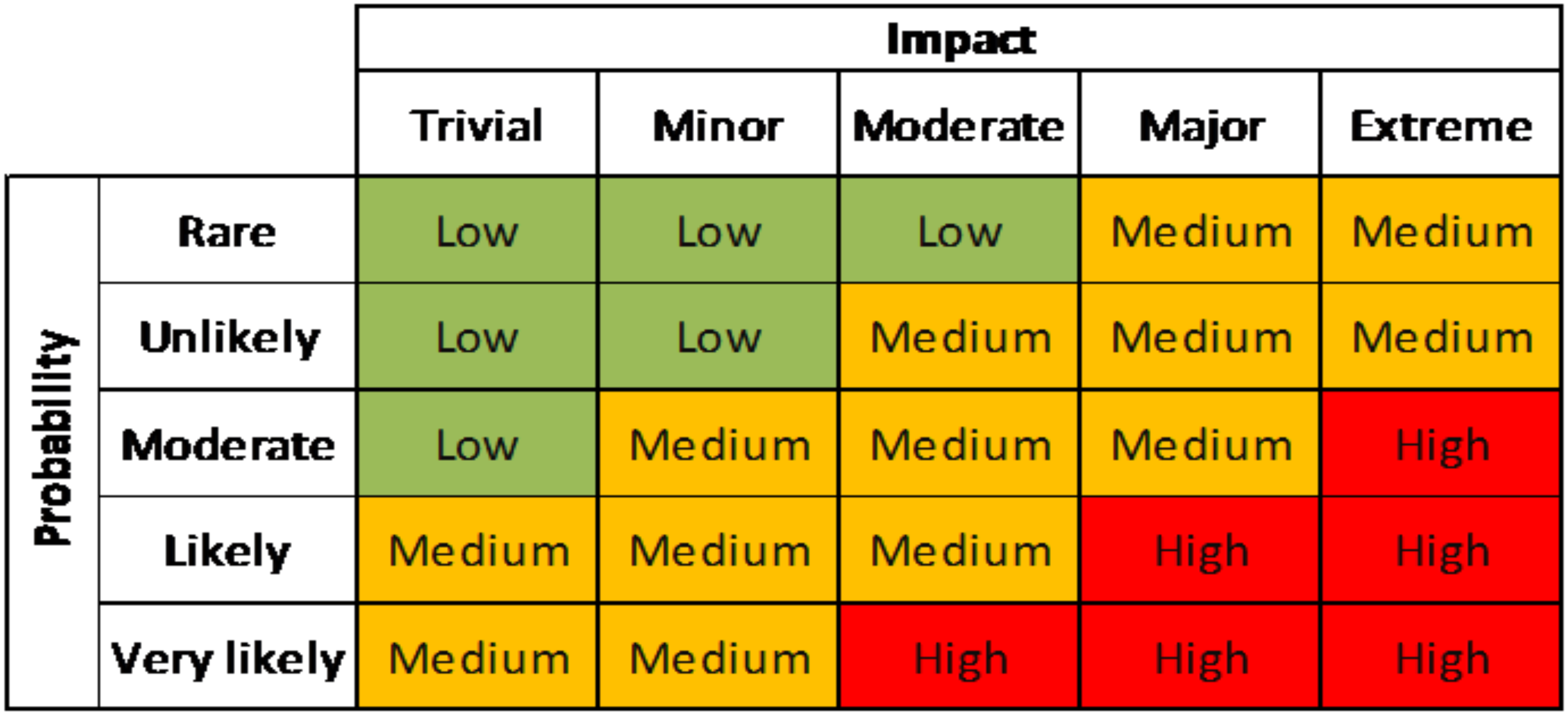
- Risk projection, also called risk estimation, attempts to rate each risk in two ways
  - Probability: the likelihood or probability that the risk is real
  - Consequences: the consequences of the problems associated with the risk, should it occur

- The project planner, along with other managers and technical staff, performs four risk projection activities:
  - Probability: establish a scale that reflects the perceived likelihood of a risk,
  - Consequences: define the consequences of the risk,
  - Impact: estimate the impact of the risk on the project and the product,
  - Accuracy: note the overall accuracy of the risk projection so that there will be no misunderstandings.

RISK COMPONENT & DRIVERS

- The major risk components (risk categories) 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*

PROBABILITY-IMPACT MATRIX

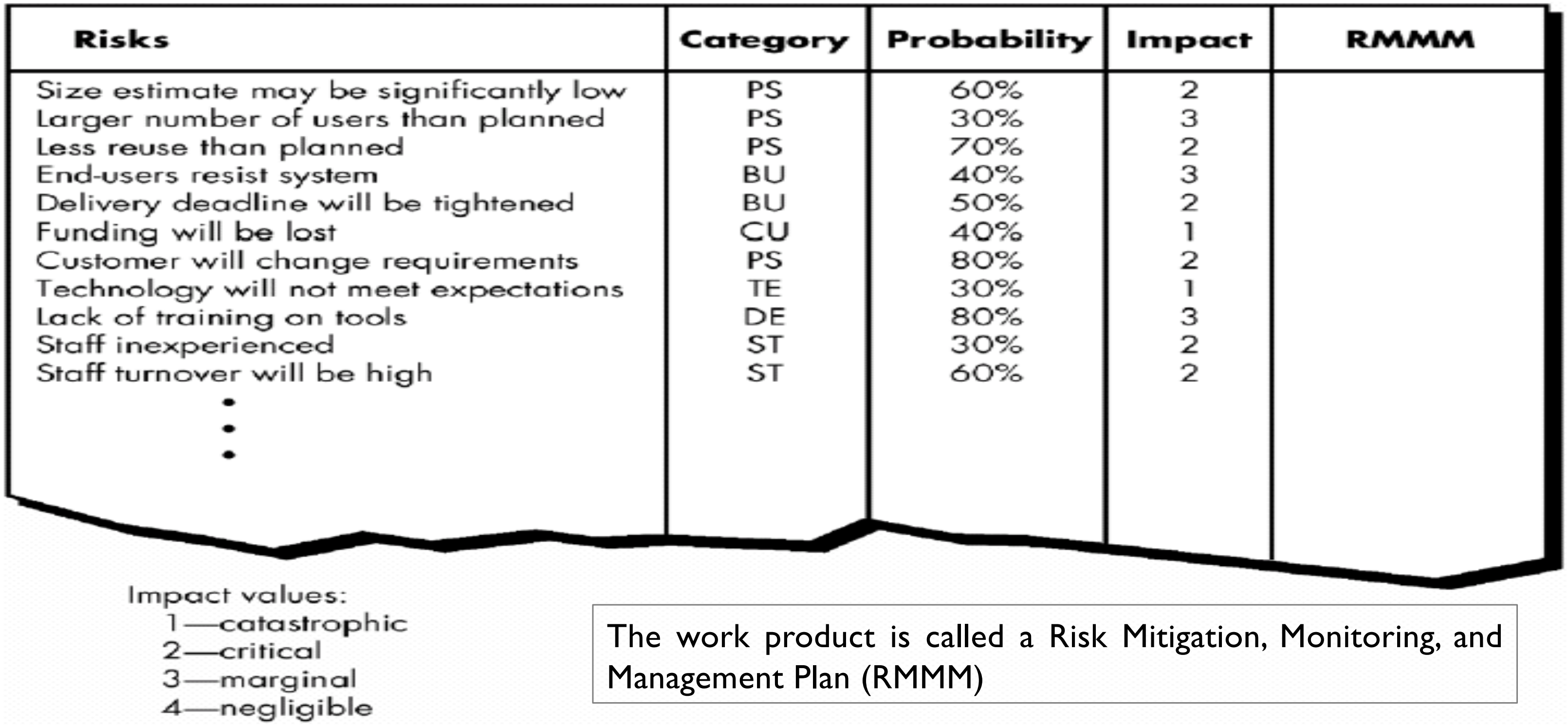


		Impact				
		Trivial	Minor	Moderate	Major	Extreme
Probability	Rare	Low	Low	Low	Medium	Medium
	Unlikely	Low	Low	Medium	Medium	Medium
	Moderate	Low	Medium	Medium	Medium	High
	Likely	Medium	Medium	Medium	High	High
	Very likely	Medium	Medium	High	High	High

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 [autopilot performance fixing with XP]
- Development environment (DE) — risks associated with the availability and quality of the tools to be used to build the product [resource allocation plan]
- 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

BUILDING RISK 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

The work product is called a Risk Mitigation, Monitoring, and Management Plan (RMMM)

ASSESSING RISK IMPACT

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.

A FRAMEWORK FOR DEALING WITH RISK - RISK MANAGEMENT

- **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? Must be an ongoing activity, as the importance and likelihood of particular risks can change as project proceeds.

RISK IDENTIFICATION

❑ Approaches of identify risks include:

- **Use of checklists** – usually based on the experience of past projects. Some risk are generic risk, they are relevant to all software projects. A standard checklist can be used to identify the risks (e.g. changing technology).
- **Brainstorming** – getting knowledgeable stakeholders together to pool concerns
- **Causal mapping** – identifying possible chains of cause and effect. For example, illness of a team member is a risk that might put the project completion date at risk and result in the late delivery of the product

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 10 DEVELOPMENT RISKS

Risk	***Risk reduction techniques***
Gold plating	Requirements scrubbing (cleaning), prototyping, design to cost
Late changes to requirements	Change control, incremental development
Shortfalls in externally supplied components	Benchmarking (evaluate by comparison with standard), inspections, formal specifications, contractual agreements, quality controls
Shortfalls in externally performed tasks	Quality assurance procedures, competitive design
Real time performance problems	Simulation, prototyping, tuning
Development technically too difficult	Technical analysis, cost-benefit analysis, prototyping , training

RISK PLANNING

Risks can be dealt with by:

- Risk prevention/avoidance – a project can, for example, be protected from the risk of overrunning the schedule by increasing duration estimates or reducing functionality.
- Risk reduction – some risk, while they cannot be prevented, can have their likelihoods reduced by prior planning. The risk of late changes to a requirements specification can, for example, be reduced by prototyping but will not eliminate the risk of late changes.
- Risk transfer – the impact of some risk can be transferred away from the project, by, for example, contracting out or taking out insurance.

RISK REDUCTION LEVERAGE (RRL)

- ❑ Risk Reduction Leverage is another Quantitative means of assessing how Risks are being managed

$$\text{Risk Reduction Leverage} = \frac{(\text{Risk Exposure Before} - \text{Risk Exposure After})}{\text{Cost of Risk Reduction}}$$

  - RE_{before} is risk exposure before risk reduction e.g. 20% chance of a fire causing $20,000 damage
  - RE_{after} is risk exposure after risk reduction e.g. fire alarm costing $1500 reduces probability of fire damage to 5%

- ❑ RRL = (0.2x20,000) – (0.05x20,000) / 1500
  = 4,000 – 1,000 / 1500 = 2

- ❑ RRL > 1.00 therefore worth doing

REFERENCES

- R.S. Pressman & Associates, Inc. (2010). *Software Engineering: A Practitioner's Approach.*
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software*, *17*(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation. *IBM Systems Journal*, *33*(1), 182-214.