SE463

Software Requirements Specification & Analysis

Risk Management

Readings:

J. D. Kiper and M. S. Feather, "A Risk-Based Approach to Strategic Decision-Making for Software Development," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005.

Risk

A risk is an uncertain factor whose occurrence may result in some loss of satisfaction of some corresponding objective.

[van Lamsveerde, p. 3.6]

- has a likelihood to occur
- has consequences
- product-related risks
- process-related risks

Project Risk Manager

Risk Exposure

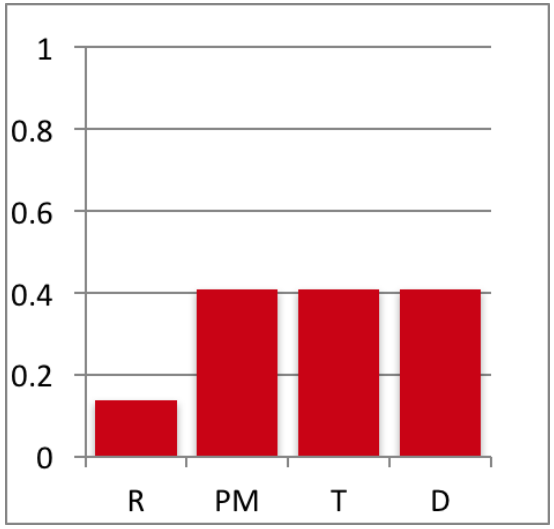
Risk Exposure is an expression of the degree of risk

RE = prob(occurrence) X cost(consequences)

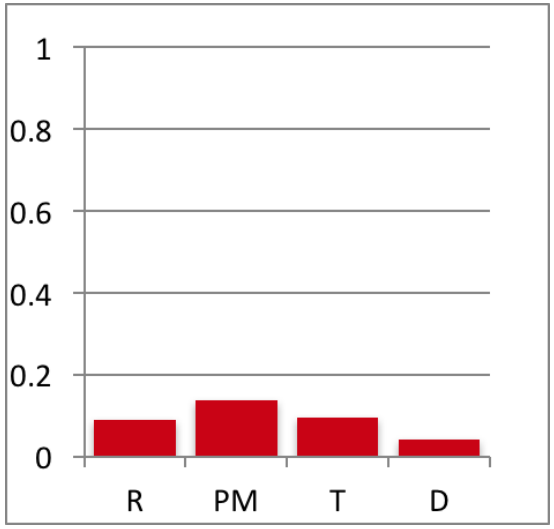
The risk exposure determines whether it makes sense to take action to reduce risk or mitigate consequences.

Risk Exposure

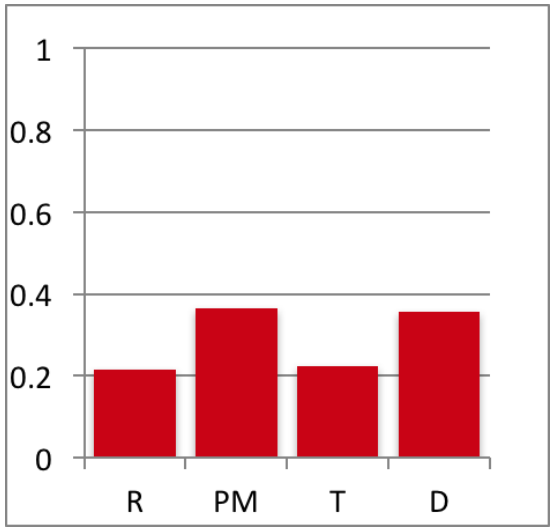
Team A



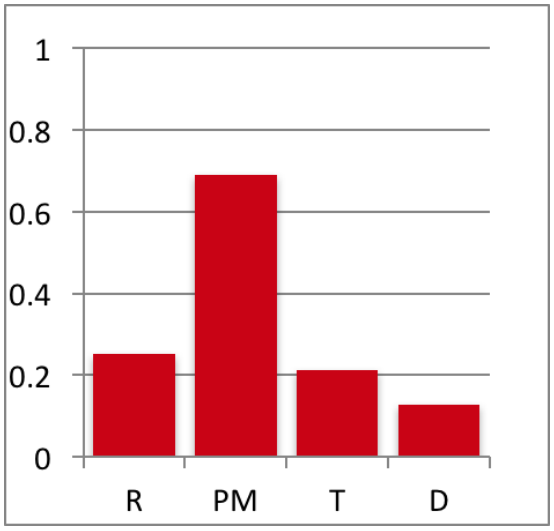
Team B



Team C



Team D



Risk Consequence Table

		Risks (Failure Modes)				
Requirements	**Weight (req)**	**Requirements** (incomplete, incorrect, ambiguous, changing)	**Project Management** (estimations, project, team management)	**Technical** (complex problem, lack of experience with technology)	**Dependencies** (on adjacent systems, components, other people)	**Loss of Objective**
Likelihood (risk)		**0.2**	**0.6**	**0.2**	**0.1**	
Creating a product that users would like	**0.7**	0.8	0.5	0.8	0.7	**0.483**
Completing the product on time	**1.0**	0.7	0.8	0.5	0.8	**0.8**
Risk Criticality		**0.252**	**0.69**	**0.212**	**0.129**	

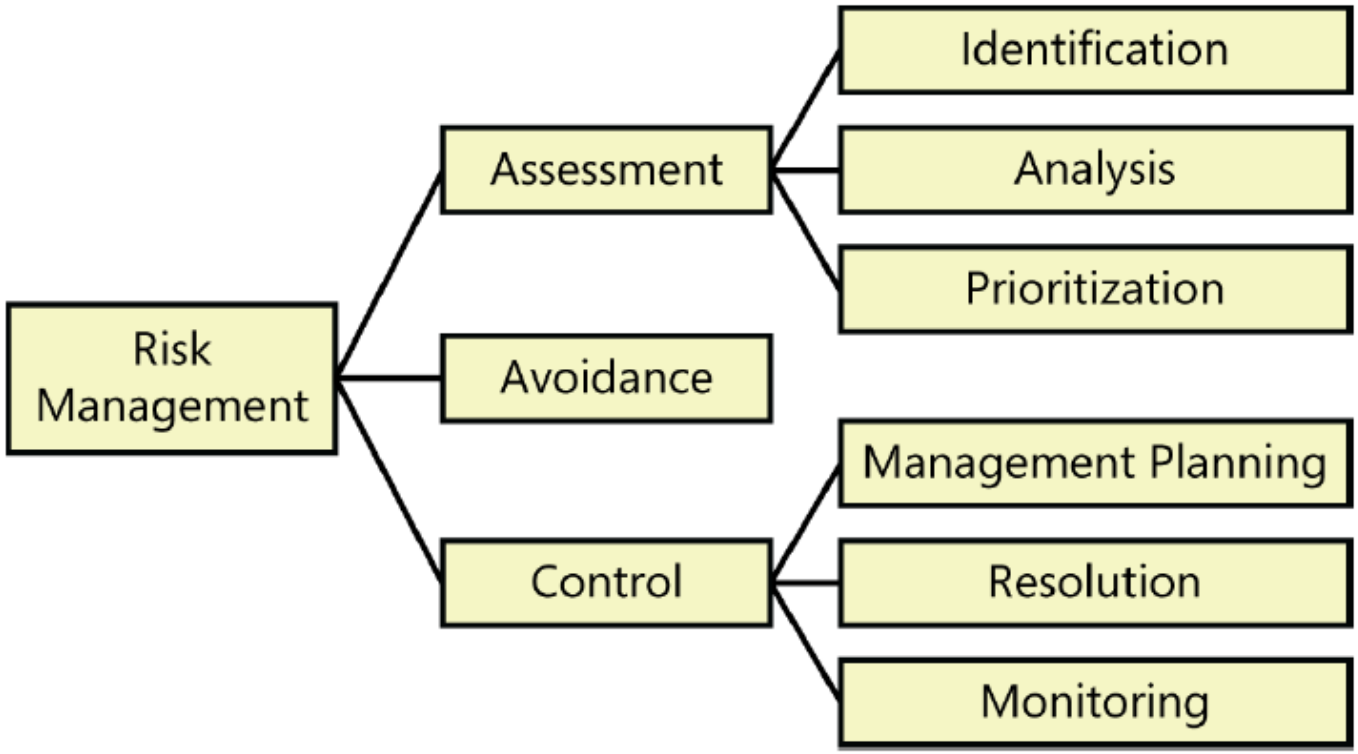
Impact(risk, req) = estimate of loss of requirement

0 = no loss

1 = total loss

Risk Management

Risk Management attempts to manage the degree to which a project is exposed to risks of quality, delay, or failure.



K. Wiegers and J. Beatty, Software Requirements 3ed, Microsoft Press, 2013.

Risk Countermeasures Table

	Risks (Failure Modes)				
Countermeasures	**Requirements** (incomplete, incorrect, ambiguous, changing)	**Project Management** (estimations, project, team management)	**Technical** (complex problem, lack of experience with technology)	**Dependencies** (on adjacent systems, components, other people)	**Overall single effect of countermeasure**
Criticality (risk)	**0.252**	**0.69**	**0.212**	**0.129**	
Collaborative elicitation process with extensive user involvement; modelling; mock-ups	0.5	0.3	0	0.1	**0.3459**
Continually estimate costs; use shorter development iterations	0	0.7	0.2	0	**0.5254**
Prototype novel or risky requirements; plan time for learning and experimentation	0.5	0.3	0.6	0.1	**0.4731**
Investigate suppliers; monitor their progress; develop backup plans	0	0	0.1	0.5	**0.0857**
Combined Risk Reduction	**0.75**	**0.853**	**0.712**	**0.595**	

Effect(cm, risk) = estimate of reduction of risk

0 = no reduction

1 = risk eliminated

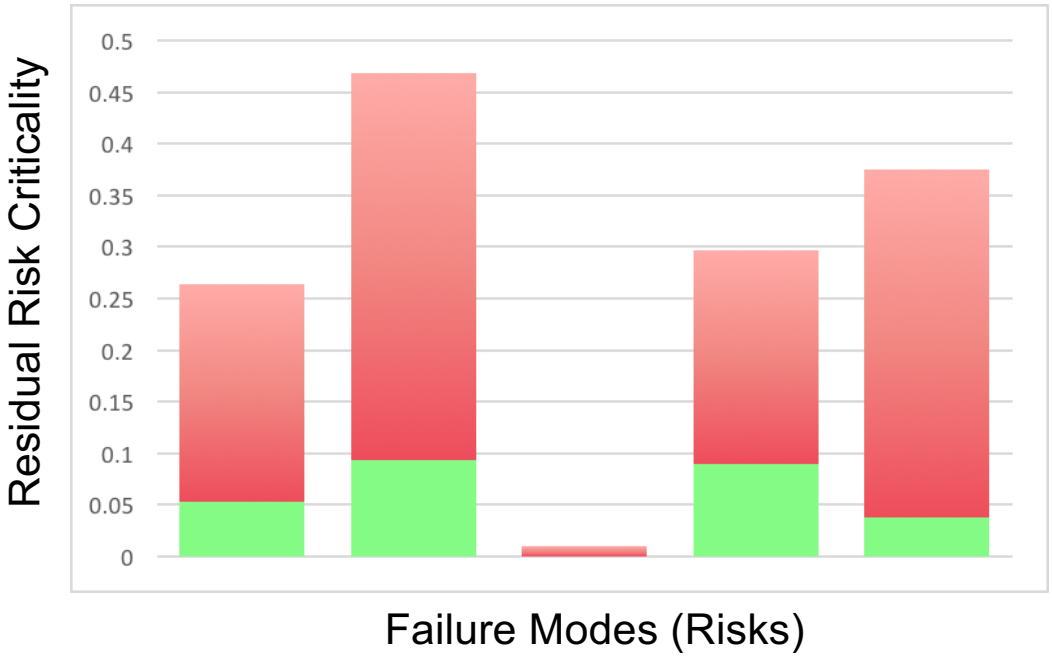
Optimizing the Residual Risk

Goal: Select the optimal combinations of countermeasures

- based on their joint effectiveness in reducing risk
- based on their associated cost of implementation

Result: Optimal (or near optimal) balance of risk exposure

- selected countermeasures become new requirements



Case Study: Medical Devices

Risk management in the medical device domain covers both the medical device and the usage of the device

- Design
- Manufacturing
- Quality control
- User training
- Interaction with other devices
- Human factors
  - User interface design
  - Job aiding
  - Personnel training
  - Cognition and perception of information

Case Study: Medical Devices

Input: user scenarios representing normal operations and special circumstances

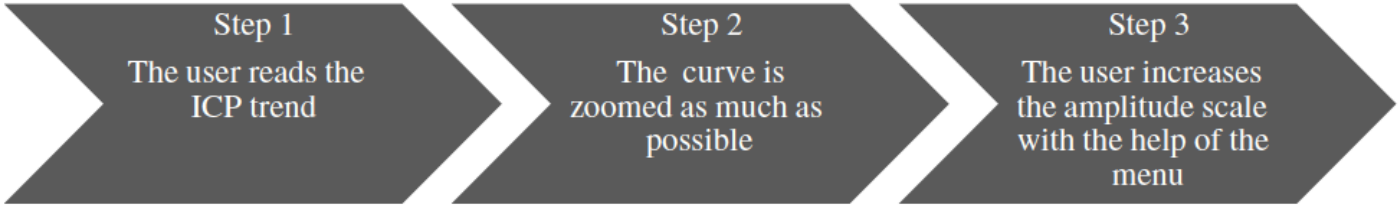


Fig. 6 User scenario A1.1

Process: risks are identified through brainstorming and assessed for probability, severity, and detectability.

Table 1 Risk identified from user scenario A1.1

ID	Conditions	Risk ID	Risk Description	S	P	R
A1.1	The amplitude scale is increased with the menu	A1.1.3 R1	Another user does not see that the amplitude scale is increased	4	4	16

Lindholm et al., "A case study on software risk analysis and planning in medical device development," Software Quality Journal, 2014.

Case Study: Medical Devices

- 15 meetings, 2-3 hours each
- 225 identified risks
  - 25 were removed (assessed to be non risks)
  - 22 user risks to be addressed
  - 11 were determined to be technical risks (to be addressed
  - 167 considered to be sufficiently managed

Summary

Software engineers and project managers are eternal optimists. We expect our next project to run smoothly, despite the history of problems on earlier projects. The reality is that dozens of potential pitfalls can delay or derail a software project. Software teams must identify and control their project risks, beginning with those related to requirements.

- Risk - What could go wrong?
- Risk Analysis
- Risk Mitigation
- Adoption of optimal countermeasures as additional requirements to be implemented.