Lecture 5, Part 1: Risk

Jennifer Campbell
CSC340 - Winter 2007

Risk

- The possibility of suffering loss
- Risk involves uncertainty and loss:
  - **Uncertainty**: The degree of certainty about whether the risk will happen.
  - **Loss**: If the risk becomes a reality, unwanted consequences or losses will occur.

Risk Categories

- Project Risk (aka Development Risk)
- Technical Risk
- Business Risk

[Pre97]

Risk Categories: Project (Development) Risks…

Schedule problems

Budget problems

Personnel problems

… threaten the project plan.

Requirements problems

Customer problems

[Pre97]

Example: Project Risk

- A company introduced OO technology into its organization, using a well-defined project "X" as the pilot.

- Many project "X" personnel were familiar with OO, but it had not been part of their development process (had very little experience and training in application of OO).

-It is taking project personnel longer than expected to climb the learning curve.

- Some personnel are concerned that the modules implemented to date might be too inefficient to satisfy project "X" performance requirements.

[SEI]

Risk Categories: Technical Risks…

design problems

implementation problems

… threaten the quality and timeliness of the software.

interface problems

maintenance problems

verification problems

[Pre97]

Risk Categories:
Business Risks…

no demand for product

lose management support

… threaten the (economic) success of the project.

lose resource commitments

lose budget commitments

[Pre97]

Reactive Risk Management

[Pre97]

Proactive Risk Management

"The purpose of risk management is to identify potential … problems before they occur so that action can be taken to reduce or eliminate the likelihood and/or impact of these problems should they occur."

[Pre97]

Principles of Risk Management

- **Global Perspective**
  - View software in context of a larger system
  - For any opportunity, identify both:
    - Potential value
    - Potential impact of adverse results
- **Forward Looking View**
  - Anticipate possible outcomes
  - Identify uncertainty
  - Manage resources accordingly
- **Open Communications**
  - Free-flowing information at all project levels

[DWA96]

Principles of Risk Management [2]

- **Integrated Management**
  - Project management is risk management!
- **Continuous Process**
  - Continually identify and manage risks
- **Shared Product Vision**
  - Everybody understands the mission
    - Common purpose
    - Collective responsibility
    - Shared ownership
- **Teamwork**
  - Work cooperatively to achieve the common goal
  - Pool talent, skills and knowledge

[DWA96]

Continuous Risk Management

5. Correct for deviations from the risk mitigation plans.

1. Produce list of project-specific risks.

4. Monitor risk indicators and reassess risks.

2. For each risk, identify probability, loss magnitude, and timeframe. Prioritize risks.

3. Choose and implement risk mitigation actions.

[DWA96]

1. Risk Identification

- **Identify product-specific risks**
  - Risks that can only be identified by those with a clear understanding of the specifics (technology, people, environment) of the project
- **Risk Identification Checklists**
  - Method of identifying risks
  - Focuses on some known and predictable risks

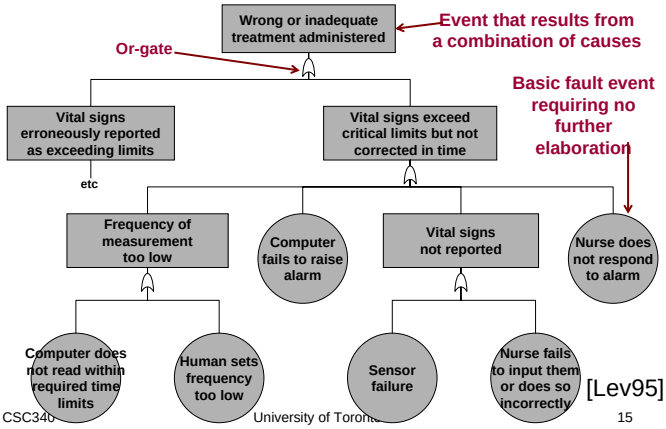
[Boe91], [Pre97]

1. Risk Identification: Top 10 Software Risks

- Personnel Shortfalls
- Unrealistic schedules/budgets
- Developing the wrong software functions
- Developing the wrong user interface
- Gold plating
- Continuing stream of requirements changes
- Shortfalls in externally furnished components
- Shortfalls in externally performed tasks
- Real-time performance shortfalls
- Straining computer science capabilities

[Boe91]

1. Risk Identification: Fault Tree



[Lev95]

2. Risk Analysis

- Determine the probability and loss magnitude for each identified risk.
- Prioritize risks
- Some important concepts:
  - **Risk Exposure -** For each risk:
    - RE = p(unsat. outcome) X loss(unsat. outcome)
  - **Risk Reduction Leverage -** For each mitigation action:
    - RRL = (RE_{before} - RE_{after}) / cost of intervention

[Boe91]

2. Risk Analysis: Measurement

- **Quantitative:**
  - Measure risk exposure using standard cost & probability measures
  - Note: probabilities are rarely independent
- **Qualitative:**
  - Assess the risk probabilities and losses on a relative scale

2. Risk Analysis: Qualitative Measurement

Undesirable outcome	**Likelihood of Occurrence**		
	Very likely	**Possible**	**Unlikely**
Loss of life	Catastrophic	Catastrophic	Severe
Loss of Spacecraft	Catastrophic	Severe	Severe
Loss of Mission	Severe	Severe	High
Degraded Mission	High	Moderate	Low
…			

3. Risk Mitigation:
Top 10 Risks
(with management techniques)

- **Personnel Shortfalls**
  - use top talent
  - team building
  - training
- **Unrealistic schedules/budgets**
  - multisource estimation
  - designing to cost
  - requirements scrubbing
- **Developing the wrong software functions**
  - better requirements analysis
  - organizational/operational analysis
- **Developing the wrong user interface**
  - prototypes, scenarios, task analysis
- **Gold plating**
  - requirements scrubbing
  - cost benefit analysis
  - designing to cost
- **Continuing stream of reqts changes**
  - high change threshold
  - information hiding
  - incremental development

[Boe91]

3. Risk Mitigation:
Top 10 Risks [2]
(with management techniques)

- **Shortfalls in externally furnished components**
  - early benchmarking
  - inspections, compatibility analysis
- **Shortfalls in externally performed tasks**
  - pre-award audits
  - competitive designs
- **Real-time performance shortfalls**
  - targeted analysis
  - simulations, benchmarks, models
- **Straining computer science capabilities**
  - technical analysis
  - checking scientific literature

[Boe91]

4. Risk Monitoring and 5. Risk Control

- Periodically review risk items.
- Address problems with resolving risks.
- Re-rank the risks.

Risk item	This Month	Last Month	Num months	Risk-resolution progress
Replacing sensor-control software developer	1	4	2	Top replacement candidate unavailable
Target hardware delivery delays	2	5	2	Procurement procedural delays
Sensor data formats undefined	3	3	3	Action items to sw, sensor teams; due next month

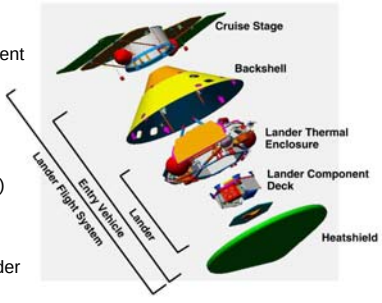
[Boe91]

Case Study: Mars Polar Lander

- **Launched**
  - 3 Jan 1999
- **Mission**
  - Land near South Pole
  - Dig for water ice with a robotic arm
- **Fate**:
  - Arrived 3 Dec 1999
  - No signal received after initial phase of descent
- **Cause:**
  - Several candidate causes
  - Most likely is premature engine shutdown due to noise on leg sensors

What happened?

- **Investigation hampered by lack of data**
  - spacecraft not designed to send telemetry during descent
  - This decision severely criticized by review boards
- **Possible causes:**
  - Landing site too steep (plausible)
  - Heatshield failed (plausible)
  - **Premature Shutdown of Descent Engines (most likely!)**
  - Parachute drapes over lander (plausible)
  - Backshell hits lander (plausible but unlikely)



Premature Shutdown

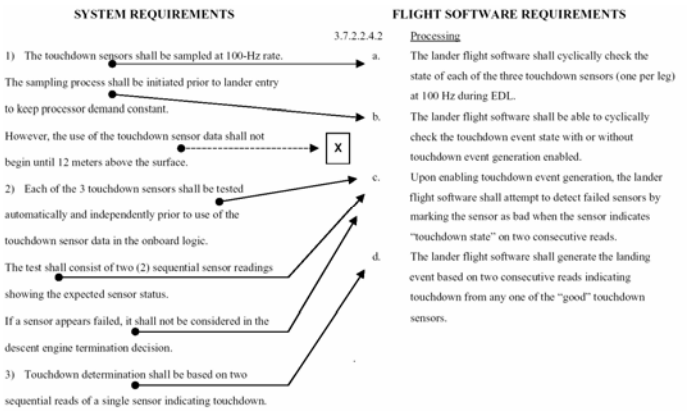
- **Cause of error**
  - Magnetic sensor on each leg senses touchdown
  - Legs unfold at 1500m above surface
    - transient signals on touchdown sensors during unfolding
    - software accepts touchdown signals if they persist for 2 timeframes
    - transient signals likely to be long enough on at least one leg

Premature Shutdown [2]

- **Factors**
  - *System* requirement to ignore the transient signals
    - But the *software* requirements did not describe the effect
    - s/w designers didn't understand the effect, so didn't implement the requirement
  - Engineers present at code inspection didn't understand the effect
  - Not caught in testing because:
    - Unit testing didn't include the transients
    - Sensors improperly wired during integration tests (no touchdown detected!)
    - Full test not repeated after re-wiring

Premature Shutdown [3]

- **Result of error**
  - Engines shut down before spacecraft has landed
    - When engine shutdown s/w enabled, flags indicated touchdown already occurred
    - estimated at 40m above surface, travelling at 13 m/s
    - estimated impact velocity 22m/s (spacecraft would not survive this)
    - nominal touchdown velocity 2.4m/s



Adapted from the "Report of the Loss of the Mars Polar Lander and Deep Space 2 Missions -- JPL Special Review Board (Casani Report) - March 2000". See http://www.nasa.gov/newsinfo/marsreports.html

Learning the Right Lessons

- **Understand the Causality**
  - Never a single cause; usually many complex interactions
  - Seek the set of conditions that are both necessary and sufficient…
    - …to cause the failure
- **Causal reasoning about failure is very subjective**
  - Data collection methods may introduce bias
    - e.g. failure to ask the right people
    - e.g. failure to ask the right questions (or provide appropriate response modes)
  - Human tendency to over-simplify
    - e.g. blame the human operator
    - e.g. blame only the technical factors

Preventing Accidents

"In most of the major accidents of the past 25 years, technical information on how to prevent the accident was known, and often even implemented. But in each case… [this was] negated by organisational or managerial flaws."

[Lev95]

References

[Boe91] Boehm, B. W. 1991. *Software Risk Management: Principles and Practices.* IEEE Software, Vol. 8, no.1, p.32-41, Jan. 1991.

[DWA96] Dorofee, A. J., Walker, J. A., Alberts, C. J., Higuera, R. P., Murray, T. J., and Williams, R. J. *Continuous Risk Management Guidebook.* Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1996.

[Lev95] Leveson, N.G. *SAFEWARE: System Safety and Computers*. Addison-Wesley, 1995.

[Pre97] Pressman, R. S. 1997. *Software Engineering: A Practitioner's Approach*. New York, USA: McGraw Hill.

References [2]

[SEI] Sofware Engineering Institute: http://www.sei.cmu.edu/sei-home.html