Lecture 10: Risk

- → **General ideas about Risk**
- → **Risk Management**
  - ↳ Identifying Risks
  - ↳ Assessing Risks
- → **Case Study:**
  - ↳ Mars Polar Lander

University of Toronto
Department of Computer Science

Risk Management

- → **About Risk**
  - ↳ Risk is "the possibility of suffering loss"
  - ↳ Risk itself is not bad, it is essential to progress
  - ↳ The challenge is to manage the amount of risk
- → **Two Parts:**
  - ↳ Risk Assessment
  - ↳ Risk Control
- → **Useful concepts:**
  - ↳ For each risk: Risk Exposure
    - ➢ RE = p(unsat. outcome) X loss(unsat. outcome)
  - ↳ For each mitigation action: Risk Reduction Leverage
    - ➢ RRL = (RE_{before} - RE_{after}) / cost of intervention

University of Toronto
Department of Computer Science

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
  - ↳ Value the individual voice
    - ➢ Unique knowledge and insights
- → **Integrated Management**
  - ↳ Project management is risk management!
- → **Continuous Process**
  - ↳ Continually identify and manage risks
  - ↳ Maintain constant vigilance
- → **Shared Product Vision**
  - ↳ Everybody understands the mission
    - ➢ Common purpose
    - ➢ Collective responsibility
    - ➢ Shared ownership
  - ↳ Focus on results
- → **Teamwork**
  - ↳ Work cooperatively to achieve the common goal
  - ↳ Pool talent, skills and knowledge

Source: Adapted from SEI Continuous Risk Management Guidebook

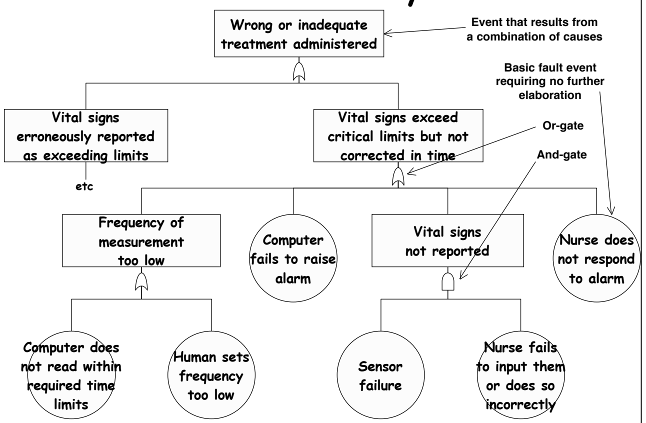
University of Toronto
Department of Computer Science

Continuous Risk Management

- → **Identify:**
  - ↳ Search for and locate risks before they become problems
    - ➢ Systematic techniques to discover risks
- → **Analyse:**
  - ↳ Transform risk data into decision-making information
  - ↳ For each risk, evaluate:
    - ➢ Impact
    - ➢ Probability
    - ➢ Timeframe
  - ↳ Classify and Prioritise Risks
- → **Plan**
  - ↳ Choose risk mitigation actions
- → **Track**
  - ↳ Monitor risk indicators
  - ↳ Reassess risks
- → **Control**
  - ↳ Correct for deviations from the risk mitigation plans
- → **Communicate**
  - ↳ Share information on current and emerging risks

Source: Adapted from SEI Continuous Risk Management Guidebook

Fault Tree Analysis



Risk Assessment

- **Quantitative:**
  - Measure risk exposure using standard cost & probability measures
  - Note: probabilities are rarely independent
- **Qualitative:**
  - Develop a risk classification matrix:

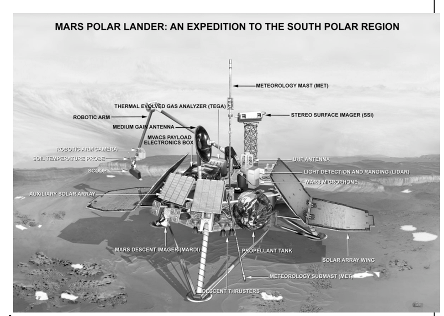
		Likelihood of Occurrence		
		Very likely	Possible	Unlikely
Undesirable outcome	(5) Loss of Life	Catastrophic	Catastrophic	Severe
	(4) Loss of Spacecraft	Catastrophic	Severe	Severe
	(3) Loss of Mission	Severe	Severe	High
	(2) Degraded Mission	High	Moderate	Low
	(1) Inconvenience	Moderate	Low	Low

Top 10 Development Risks (+ Countermeasures)

- **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
- **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

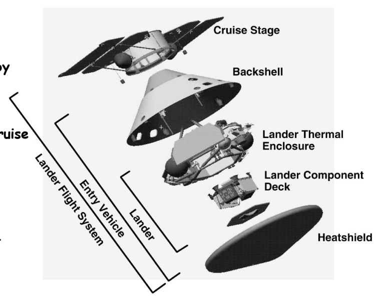
Case Study: Mars Polar Lander

- **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:**
  - Lander failed to separate from cruise stage (plausible but unlikely)
  - Landing site too steep (plausible)
  - Heatshield failed (plausible)
  - Loss of control due to dynamic effects (plausible)
  - Loss of control due to center-of-mass shift (plausible)
  - Premature Shutdown of Descent Engines (most likely!)
  - Parachute drapes over lander (plausible)
  - Backshell hits lander (plausible but unlikely)



Premature Shutdown Scenario

- **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
- **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
- **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

SYSTEM REQUIREMENTS

1) The touchdown sensors shall be sampled at 100-Hz rate.

The sampling process shall be initiated prior to lander entry to keep processor demand constant.

However, the use of the touchdown sensor data shall not begin until 12 meters above the surface.

X

2) Each of the 3 touchdown sensors shall be tested automatically and independently prior to use of the touchdown sensor data in the onboard logic.

The test shall consist of two (2) sequential sensor readings showing the expected sensor status.

If a sensor appears failed, it shall not be considered in the descent engine termination decision.

3) Touchdown determination shall be based on two sequential reads of a single sensor indicating touchdown.

FLIGHT SOFTWARE REQUIREMENTS

3.7.2.2.4.2 Processing

a. The lander flight software shall cyclically check the state of each of the three touchdown sensors (one per leg) at 100 Hz during EDL.

b. The lander flight software shall be able to cyclically check the touchdown event state with or without touchdown event generation enabled.

c. Upon enabling touchdown event generation, the lander flight software shall attempt to detect failed sensors by marking the sensor as bad when the sensor indicates "touchdown state" on two consecutive reads.

d. The lander flight software shall generate the landing event based on two consecutive reads indicating touchdown from any one of the "good" touchdown sensors.

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

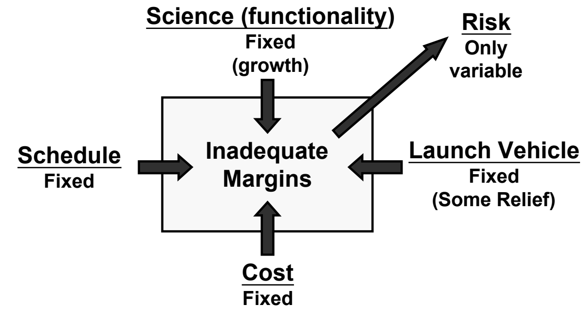
- **Understand the Causality**
  - Never a single cause; usually many complex interactions
  - Seek the set of conditions that are both necessary and sufficient…
    - …to cause the failure
- **Causal reasoning about failure is very subjective**
  - Data collection methods may introduce bias
    - e.g. 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

"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." (Leveson, Safeware)

Is there an existing "Safety Culture"?

- **Are overconfidence and complacency common?**
  - the Titanic effect - "it can't happen to us!"
  - Do managers assume it's safe unless someone can prove otherwise?
- **Are warning signs routinely ignored?**
  - What happens to diagnostic data during operations?
  - Does the organisation regularly collect data on anomalies?
  - Are *all* anomalies routinely investigated?
- **Is there an assumption that risk decreases?**
  - E.g. Are successful missions used as an argument to cut safety margins?
- **Are the risk factors calculated correctly?**
  - E.g. What assumptions are made about independence between risk factors?
- **Is there a culture of silence?**
  - What is the experience of whistleblowers? (Can you even find any?)

Failure to manage risk



Adapted from MPIAT - Mars Program Independent Assessment Team Summary Report, NASA JPL, March 14, 2000.
See http://www.nasa.gov/newsinfo/marsreports.html

Summary

- **Risk Management is a systematic activity**
  - Requires both technical and management attention
  - Requires system-level view
  - Should continue throughout a project
- **Techniques exist to identify and assess risks**
  - E.g. fault tree analysis
  - E.g. Risk assessment matrix
- **Risk and Requirements Engineering**
  - Risk analysis can uncover new requirements
    - Especially for safety-critical or security-critical applications
  - Risk analysis can uncover feasibility concerns
  - Risk analysis will assist in appropriate management action