



Lecture 10: Managing Risk

General ideas about Risk

Risk Management

Identifying Risks

Assessing Risks

Case Study:

Mars Polar Lander



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**

$$RE = p(\text{unsatisfactory outcome}) \times \text{loss}(\text{unsatisfactory outcome})$$

For each mitigation action: **Risk Reduction Leverage**

$$RRL = (RE_{\text{before}} - RE_{\text{after}}) / \text{cost of intervention}$$





Risk Assessment

Quantitative:

Measure risk exposure using standard cost & probability measures

Note: probabilities are rarely independent

Qualitative:

Develop a risk exposure matrix

Eg for NASA:

		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 SE risks (with countermeasures)

Source: Adapted from Boehm, 1989

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 requirements 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

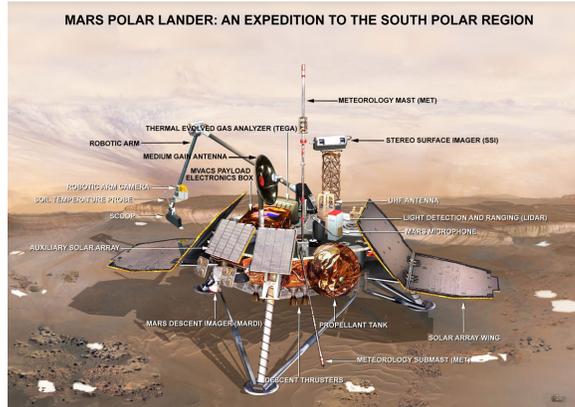
Land near South Pole
Dig for water ice with a robotic arm

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



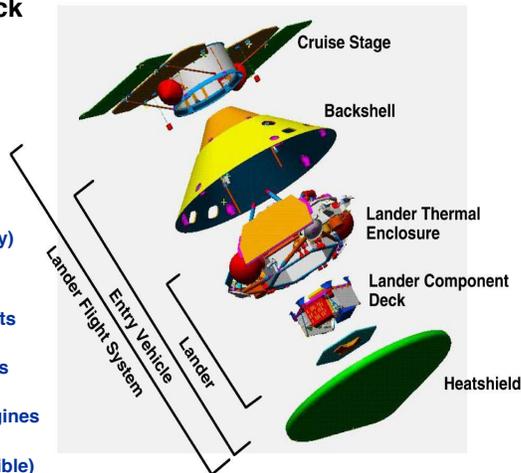
What happened?

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
- software accepts transient signals on touchdown sensors during unfolding

Factors

- System requirement to ignore the transient signals
- But the software requirements did not describe the effect
- 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!)

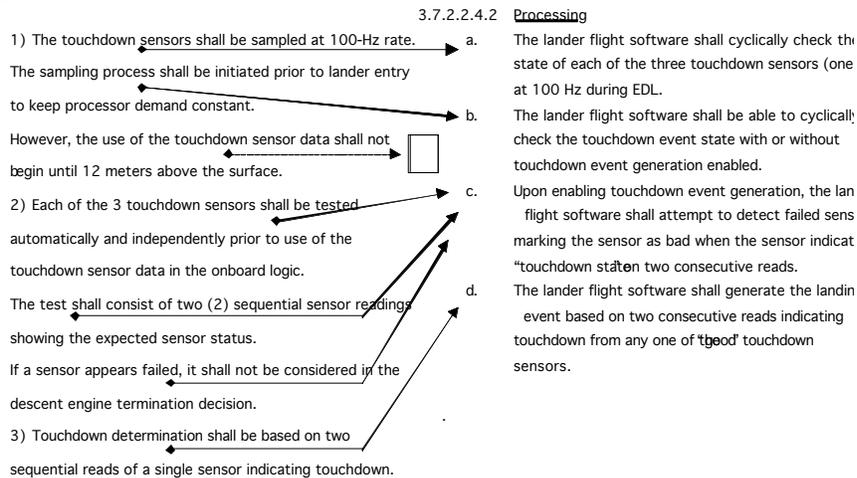
Result of error

- Engines shut down before spacecraft has landed
- 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

FLIGHT SOFTWARE REQUIREMENTS



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>



Lessons?

Documentation is no substitute for real communication

Software bugs hide behind other bugs
(full regression testing essential!)

Fixed cost + fixed schedule = increased risk



Case Study: Mars Climate Orbiter

Launched

11 Dec 1998

Mission

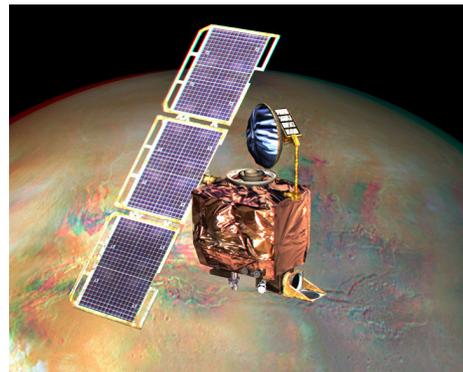
interplanetary weather satellite
communications relay for Mars Polar Lander

Fate:

Arrived 23 Sept 1999
No signal received after initial orbit insertion

Cause:

Faulty navigation data caused by failure to convert imperial to metric units





MCO Events

Locus of error

Ground software file called "Small Forces" gives thruster performance data
data used to process telemetry from the spacecraft

Angular Momentum Desaturation (AMD) maneuver effects underestimated
(by factor of 4.45)

Cause of error

Small Forces Data given in Pounds-seconds (lbf-s)

The specification called for Newton-seconds (N-s)

Result of error

As spacecraft approaches orbit insertion, trajectory is corrected

Aimed for periapse of 226km on first orbit

Estimates were adjusted as the spacecraft approached orbit insertion:

1 week prior: first periapse estimated at 150-170km

1 hour prior: this was down to 110km

Minimum periapse considered survivable is 85km

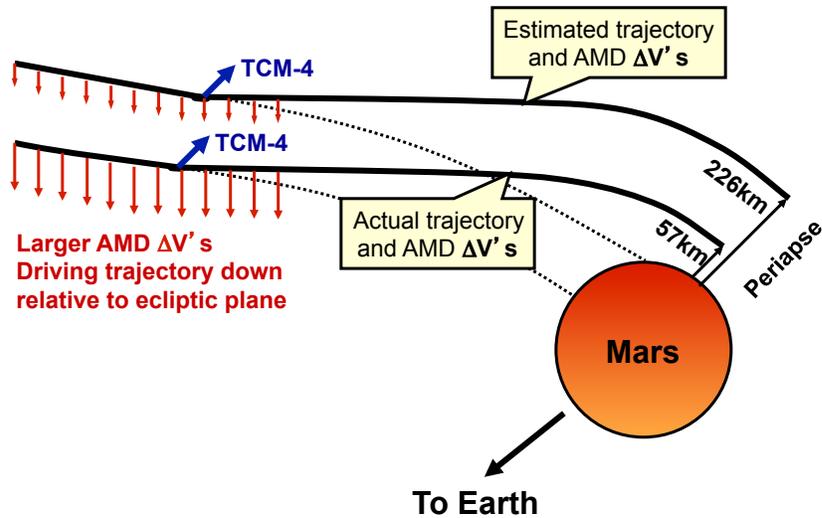
MCO entered Mars occultation 49 seconds earlier than predicted

Signal was never regained after the predicted 21 minute occultation

Subsequent analysis estimates first periapse of 57km



MCO Navigation Error





Contributing Factors

For 4 months, AMD data not used (file format errors)

Navigators calculated data by hand
File format fixed by April 1999
Anomalies in the computed trajectory became apparent almost immediately

Limited ability to investigate:

Thrust effects measured along line of sight using doppler shift
AMD thrusts are mainly perpendicular to line of sight

Poor communication

Navigation team not involved in key design decisions
Navigation team did not report the anomalies in the issue tracking system

Inadequate staffing

Operations team monitoring 3 missions simultaneously (MGS, MCO and MPL)

Operations Navigation team unfamiliar with spacecraft

Different team from development & test
Did not fully understand significance of the anomalies
Surprised that AMD was performed 10-14 times more than expected

Inadequate Testing

Software Interface Spec not used during unit test of small forces software
End-to-end test of ground software was never completed
Ground software considered less critical

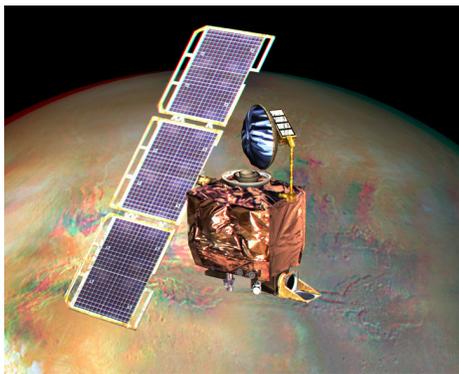
Inadequate Reviews

Key personnel missing from critical design reviews

Inadequate margins...



Mars Climate Orbiter



Mars Global Surveyor





Lessons?

If it doesn't behave how you expect, it's not safe
(yes, really!)

**If your teams don't coordinate,
neither will their software**
(See: Conway's Law)

**With software, everything is connected
to everything else -- every subsystem is critical**



Sidetrack: SNAFU principle

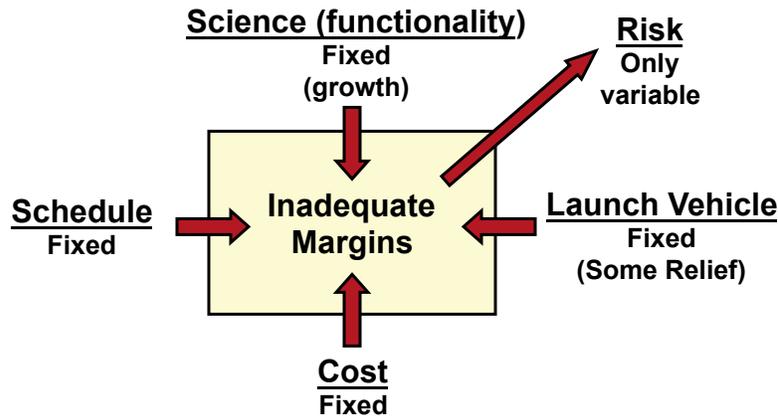
**Full communication is only possible among peers;
Subordinates are too routinely rewarded for telling
pleasant lies, rather than the truth.**

**Not a good idea to have the
IV&V teams reporting to the program office!!**





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>



Principles of Risk Management

Source: Adapted from SEI Continuous Risk Management Guidebook

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

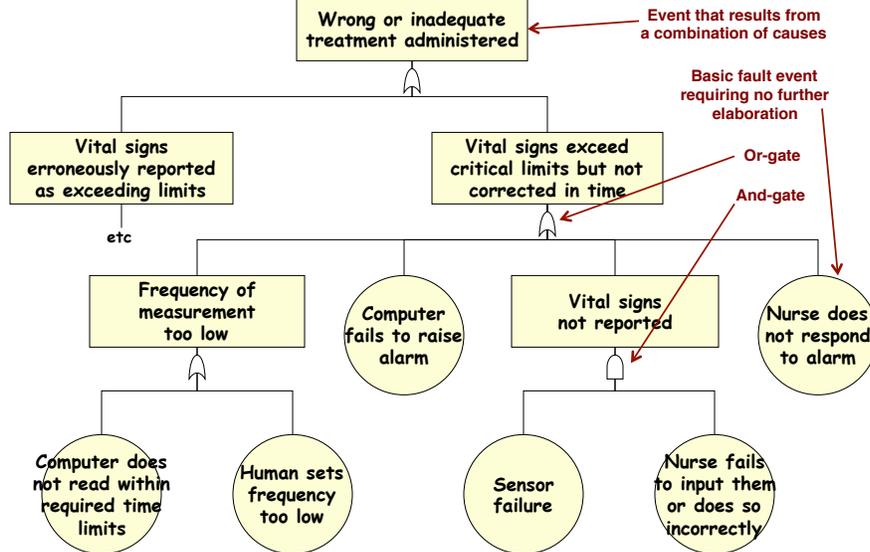
- Work cooperatively to achieve the common goal
- Pool talent, skills and knowledge





Identifying Risks: Fault Tree Analysis

Source: Adapted from Leveson, "Safeware", p321



Continuous Risk Management

Source: Adapted from SEI Continuous Risk Management Guidebook

Identify:

Search for and locate risks before they become problems
Systematic techniques to discover risks

Analyze:

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

