Lecture 1: Why Does Software Fail?

- **Some background**
  - What is Software Engineering?
  - What causes system failures?
  - The role of good engineering practice
- **Are software failures like hardware failures?**
  - Shuttle flight STS51-L (Challenger)
  - Ariane-5 flight 501
- **Some conclusions**
  - e.g. Reliable software has very little to do with writing good programs
  - e.g. Humans make mistakes, but good engineering practice catches them!

Defining Software Engineering

- **"Engineering…**
  - …creates cost-effective solutions to practical problems by applying scientific knowledge to building things in the service of humankind"
- **Software Engineering:**
  - the "things" contain software (??)
- **BUT:**
  - pure software is useless!
    - …software exists only as part of a system
  - software is invisible, intangible, abstract
  - there are no physical laws underlying software behaviour
  - there are no physical constraints on software complexity
  - software never wears out
    - …traditional reliability measures don't apply
  - software can be replicated perfectly
    - …no manufacturing variability

Failures and Catastrophes

- **System Components often fail**
  - Parts wear out
  - Wires and joints come loose
  - Cosmic rays scramble your circuits!
  - Components get used for things they weren't designed for
  - Designs don't work the way they should
- **Point failures typically don't lead to catastrophe**
  - backup systems
  - fault tolerant designs
  - redundancy
  - certification using safety factors (eg 2x)
- **Good Engineering Practice prevents accidents**
  - failure analysis
  - reliability estimation
  - checks and balances

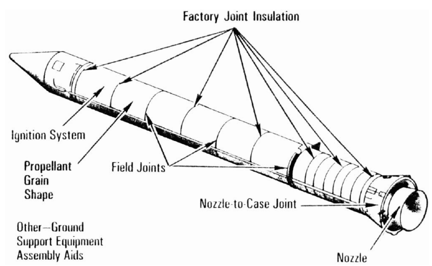
But how does this work in *Software* Engineering???

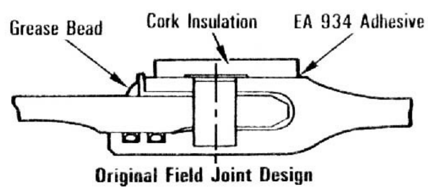
Shuttle Flight 51-L (Challenger)

- **Contracts for shuttle awarded 1972:**
  - Rockwell - Orbiter
  - Martin Marietta - external tank
  - Morton Thiokol - Solid Rocket Boosters (SRBs)
  - Rocketdyne - Orbiter Main engines
- **3 NASA centers provide management:**
  - JSC - Manage the orbiter
  - Marshall - Manage engines, tank and SRBs
  - KSC - Assembly, checkout and launch
- **4 orbiters were built:**
  - flights began in '81;
  - declared operational July '92 after STS-4
  - 24 flights over 57 months up to Dec 1995

Challenger Disaster

- Technical cause:
  - failure of a pressure seal ("O-ring") in the aft field joint of the right solid rocket motor
  - Solid rocket motor assembled from four cylindrical sections, 25 feet long, 12 feet diameter, containing 100 tons of fuel
  - 2 O-rings seal gaps in the joints caused by pressure at ignition
- Factors:
  - temperature: cold reduces resiliency of the O-ring
  - chance of O-ring failure increased by test procedures causing blow holes in the putty used to pack the joint
- But this was just the point failure…





What really happened?

- 1977: Tests show rotation of joints causes loss of secondary O-ring as a backup seal
- 1980: SRB joint classified as criticality 1R
- 1981-82 Anomalies in O-rings found in initial flights
  - but not entered into Marshall's problem assessment system
- Dec 82: Tests show secondary O-ring no longer functional under 40% of max operating pressure.
  - Criticality changed to 1
  - Paperwork after this time still shows SRB joints as 1R

1985

- Jan 24: STS 51-C launched in lowest ever temperature: 53°F (≈11°C)
  - O-ring erosion worst yet.
- Feb 8: Analysis by Thiokol noted risk of O-ring failure
  - concluded risk should be accepted because of secondary O-ring.

Leading up to the launch

1985 (cont.)

- April 29: STS 51-B:
  - primary O-ring never sealed, secondary eroded beyond predicted limits
  - as a result, Marshall placed a launch constraint on 51-F and all subsequent flights
  - Thiokol were unaware of this constraint (which was waived for each flight thereafter)
- July:
  - Thiokol engineers set up task force to solve the O-ring problem
  - Oct: task force complains of lack of cooperation from management.
  - Dec: Thiokol management recommends closure of O-ring problem
- Oct/Nov: 61-A & 61-B both experience O-ring problems

1986

- 51-L Launch originally scheduled for Jan 23rd
  - Jan 23: Flight 51-L re-scheduled for 25th
  - Jan 25: Unacceptable weather forecast
  - Jan 27: countdown halted - jammed exit hatch
- Launch re-scheduled for Jan 28th, at 9:38am
  - temperature of 27°F (≈-3°C) predicted for launch time
  - previous coldest launch: 53°F (≈11°C)

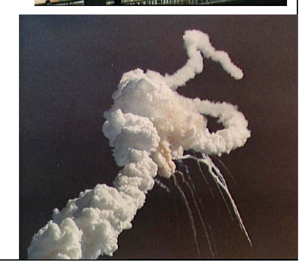
The Launch decision

Jan 27, 1986

- 2:30pm
  - Thiokol engineers express concern at predicted low temp.
- 5:45pm
  - Thiokol presents its concerns to Marshal
  - recommends launch should be delayed
- 8:45pm
  - Thiokol re-presents its conclusions to larger meeting
  - Marshall criticizes it for changing the launch criteria
- 10:30pm
  - meeting recessed for Thiokol discussion
  - engineers express strong objections to launch
- 11:00pm meeting reconvened
  - Thiokol management withdrew objections to launch

Jan 28, 1986

- 11:39am: flight 51-L launched
  - 73 seconds later, Challenger explodes



Rogers' report findings

- Lack of trend analysis
- Management Structure:
  - safety, reliability and QA placed under the organizations they were to check
  - organizational responsibility for safety was not adequately integrated with decision-making
  - No safety representative at the meetings on 27 Jan.
- Problem reporting and tracking
- Complacency:
  - Escalating risk accepted
  - Perception that less safety reliability and QA activity needed once Shuttle missions became routine
- Program Pressures were a factor
  - Pressure on NASA to build up to 24 missions per year
    - Shortened training schedules, lack of spare parts, and dilution of human resources.
    - Customer commitments may have obscured engineering concerns
  - Reduction of skilled personnel

Ariane-5 flight 501

- Background
  - European Space Agency's reusable launch vehicle
  - Ariane-4 a major success
  - Ariane-5 developed for larger payloads
- Launched
  - 4 June 1996
- Mission
  - $500 million payload to be delivered to orbit
- Fate:
  - Veered off course during launch
  - Self-destructed 40 seconds after launch
- Cause:
  - Unhandled floating point exception in Ada code

Ariane-5 Events

- Locus of error:
  - Platform alignment software (part of the Inertial Reference System, SRI)
  - This software only produces meaningful results prior to launch
  - Still operational for 40 seconds after launch
- Cause of error:
  - Ada exception raised and not handled:
    - Converting 64-bit floating point to 16-bit signed integer for Horizontal Bias (BH)
  - Requirements state that computer should shut down if unhandled exception occurs
- Launch+30s: Inertial Reference Systems fail
  - Backup SRI shuts down first
  - Active SRI shuts down 50ms later for same reason
- Launch+31s: On-board Computer receives data from active SRI
  - Diagnostic bit pattern interpreted as flight data
  - OBC commands full nozzle deflections
  - Rocket veers off course
- Launch+33s: Launcher starts to disintegrate
  - Self-destruct triggered

Why did this failure occur?

- Why was Platform Alignment still active after launch?
  - SRI Software reused from Ariane-4
  - 40 sec delay introduced in case of a hold between -9s and -5s
    - Saves having to reset everything
    - Feature used once in 1989
- Why was there no exception handler?
  - An attempt to reduce processor workload to below 80%
    - Analysis for Ariane-4 indicated overflow was not physically possible
    - Ariane-5 had a different trajectory
- Why wasn't the design modified for Ariane-5?
  - Not considered wise to change software that worked well on Ariane-4
- Why did the SRIs shut down?
  - Assumed faults are random hardware failures, hence should switch to backup
- Why was the error not caught in unit testing?
  - No trajectory data for Ariane-5 was provided in the requirements for SRIs
- Why was the error not caught in integration testing?
  - Full integration testing considered too difficult/expensive
  - SRIs were considered to be fully certified
  - Integration testing used simulations of the SRIs
- Why was the error not caught by inspection?
  - The implementation assumptions weren't documented
- Why did the OBC use diagnostic data as flight data?
  - They assumed this couldn't happen???

Summary

- Failures can usually be traced to a single root cause
- System of testing and validation designed to catch such problems
  - Catastrophes occur when this system fails
- In most cases, it takes a failure of both engineering practice and of management
- Reliable software depends not on writing flawless programs but on how good we are at:
  - Communication (sharing information between teams)
  - Management (of Resources and Risk)
  - Verification and Validation
  - Risk Identification and tracking
  - Questioning assumptions

Readings

- Van Vliet, chapter 1
  - Read all of it, especially the part about a code of ethics
- Challenger (& Space Shuttle in general)
  - Current info about the shuttle:
    - http://spaceflight.nasa.gov/shuttle/
  - Info about Challenger:
    - http://www-pao.ksc.nasa.gov/kscpao/shuttle/missions/51-l/mission-51-l.html
  - Rogers Commission Report (see especially appendix F, by Richard Feynman)
    - http://science.ksc.nasa.gov/shuttle/missions/51-l/docs/rogers-commission/table-of-contents.html
  - A Succinct summary of the key factors and issues with Challenger:
    - http://ethics.tamu.edu/ethics/ethics/shuttle/shuttle1.htm
- Ariane-5
  - Info about ESA's launchers:
    - http://www.esa.int/export/esaLA/launchers.html
  - Flight 501 inquiry report & Press release:
    - http://www.esrin.esa.it/htdocs/tidc/Press/Press96/press33.html