How Software Errors Contribute to Satellite Failures -

Challenges Facing the Risk Analysis Community

15 May 2003

SCSRA Annual Workshop

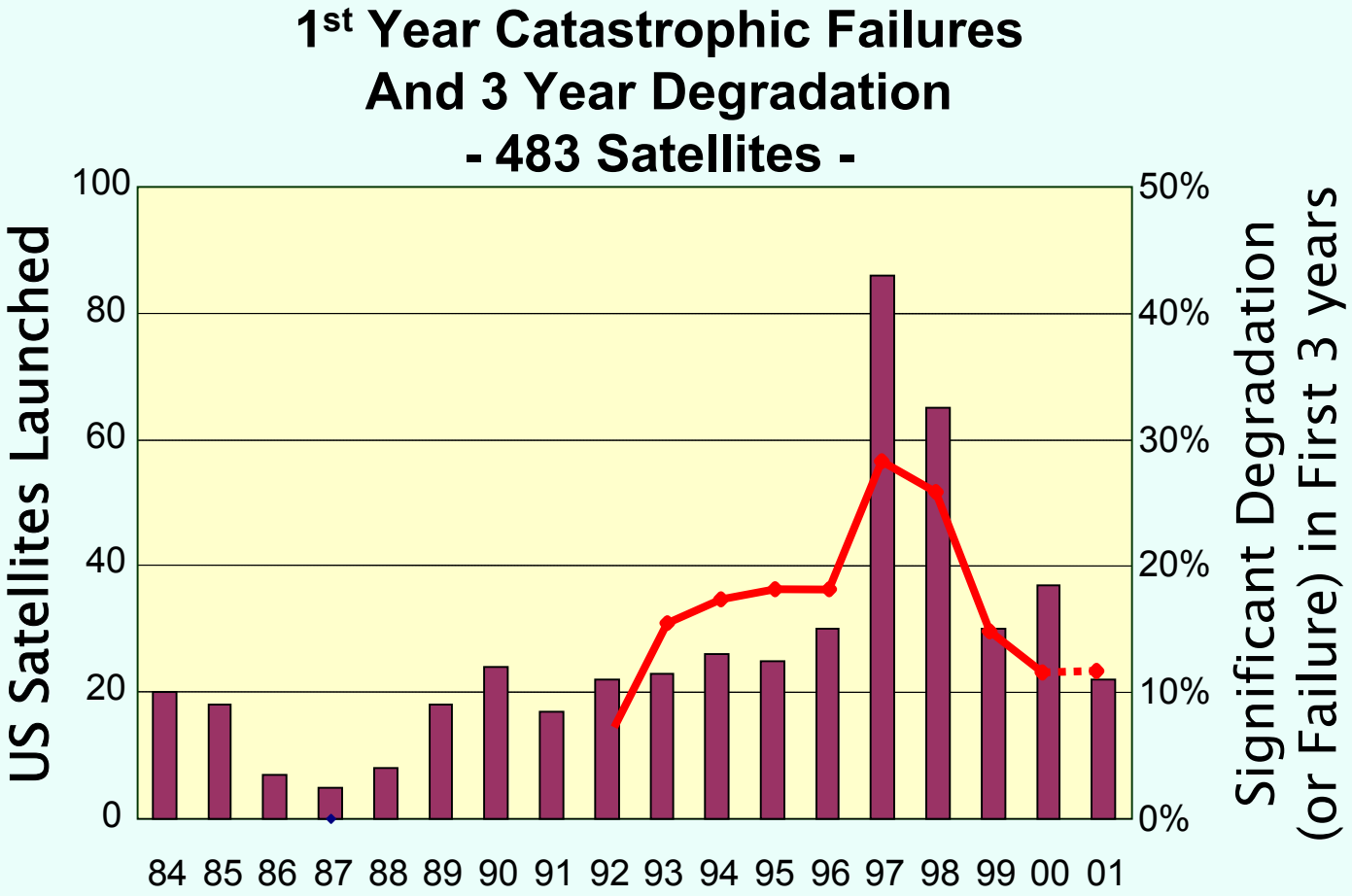
Paul G. Cheng

Risk Assessment & Management Subdivision

Systems Engineering Division

THE AEROSPACE CORPORATION

We Have Been Going to Space for Over 40 Years Yet Our Reliability Record Remains Spotty

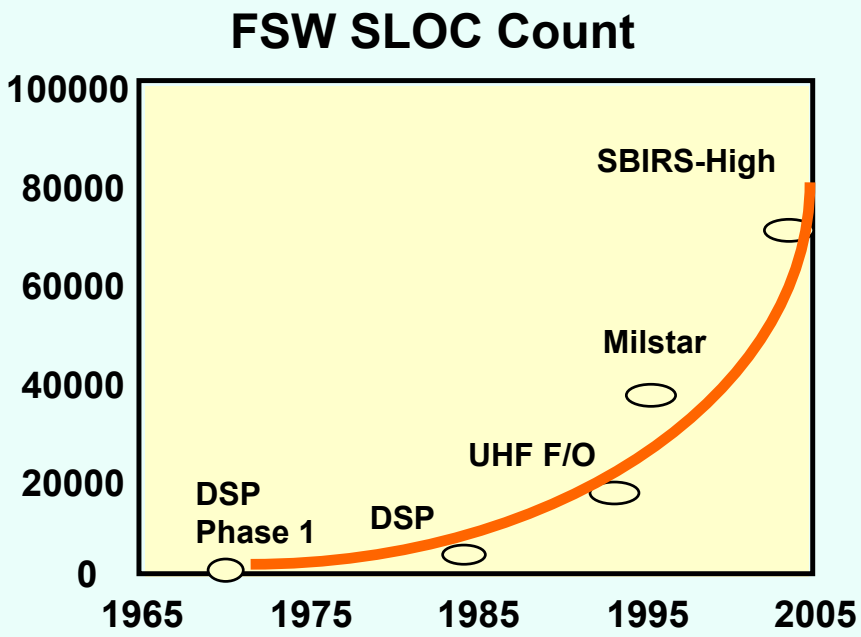


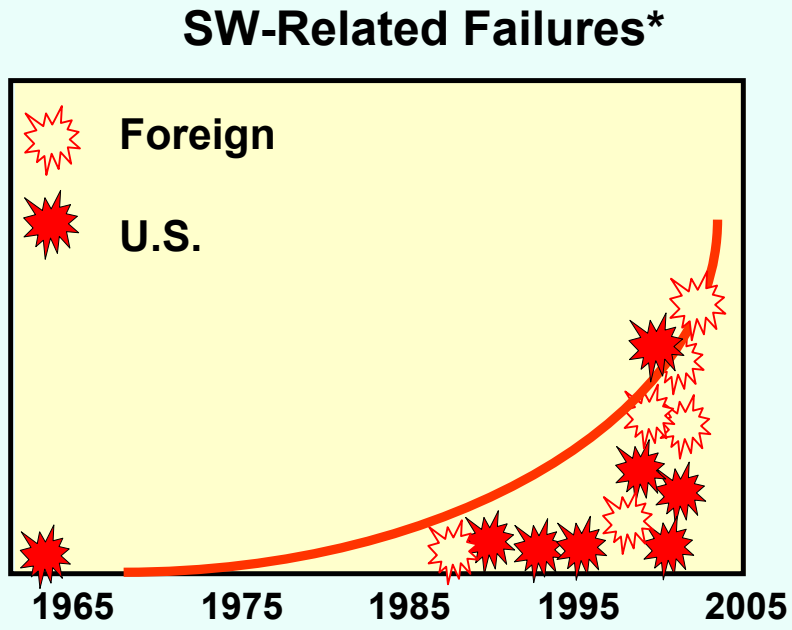
Why Do Satellites Fail?

Unclassified U.S. Government Satellites, 1990–Present

Date	Program	Problem/Outcome	Engineering Mistake	Technology Surprise
04/90	Hubble	Severe mirror aberration due to a defect in the instrument used both in manufacturing and in QA.	X	
07/92	TSS-1	Deployment mechanism jammed by a bolt added after I&T.	X	
09/92	Mars Observer	Lost contact after repressurization of the propulsion system, probably due to oxidizer leak.		X
08/93	NOAA 13	An overly long screw shorted the battery charger	X	
10/93	Landsat F	Pyrovalve blowby ignited fuel.		X
01/94	Clementine	CPU froze due to data handling overload, allowing the thruster to continuously fire, depleting the fuel.		X
05/94	MSTI 2	Contact lost, probably due to micrometeoroid/debris impact, charging, or combinations thereof.		X
12/95	Skipper	Wiring for solar arrays configured in reverse due to drawing interpretation error. The battery was drained.	X	
02/96	TSS-1R	Severe arcing due to contamination within the insulation layers burned the tether.	X	
08/97	Lewis	A technically flawed GN&C design caused tumbling -not promptly arrested due to inadequate monitoring.	X	X
10/97	STEP-4	Damaged by launch vibration – ground test deemed inadequate	X	
10/98	STEX	Solar array ran too hot, causing solder joint fatigue and severe performance loss. Thermal analysis done on wrong configuration.	X	
12/98	MCO	Metric/English unit mix-up in flight software, coupled with vulnerable navigation scheme, caused probe loss.	X	
01/99	Mars Polar Lander	Touchdown sensors not protected from deployment shock, causing premature engine shutdown.	X	
03/99	WIRE	A starting transient from the pyro electronics controller prematurely ejected the telescope cover.		X
		Count	**10**	**6**
		Since 1995	**7**	**2**

Software Increasingly Matters





Over half of failures* between 1998 and 2000 involved software

FSW SLOC = Flight Software Source Lines of Codes

*: Includes Launch Vehicles

1962	Mariner I (Atlas)
1990	Intelsat 6 (Titan CT2)
1991	Orbcomm X
1994	Clementine
1999	Milstar 2-1
1999	MCO
1999	Terriers
1999	MPL
1988	Phobos 1
1996	Cluster (Ariane 501)
1998	SOHO
2000	STRV x 2
2000	ICO F1 (Sea Launch)
2000	QuickBird (Cosmos 3M)

NEAR and Phobos 2 not counted

Software Risks Are Underappreciated

- Small error can be fatal.
- Redundancy is ineffective.
- Risks do not necessarily decrease over time.
- More human factors are involved.

Challenges Facing the Risk Community:

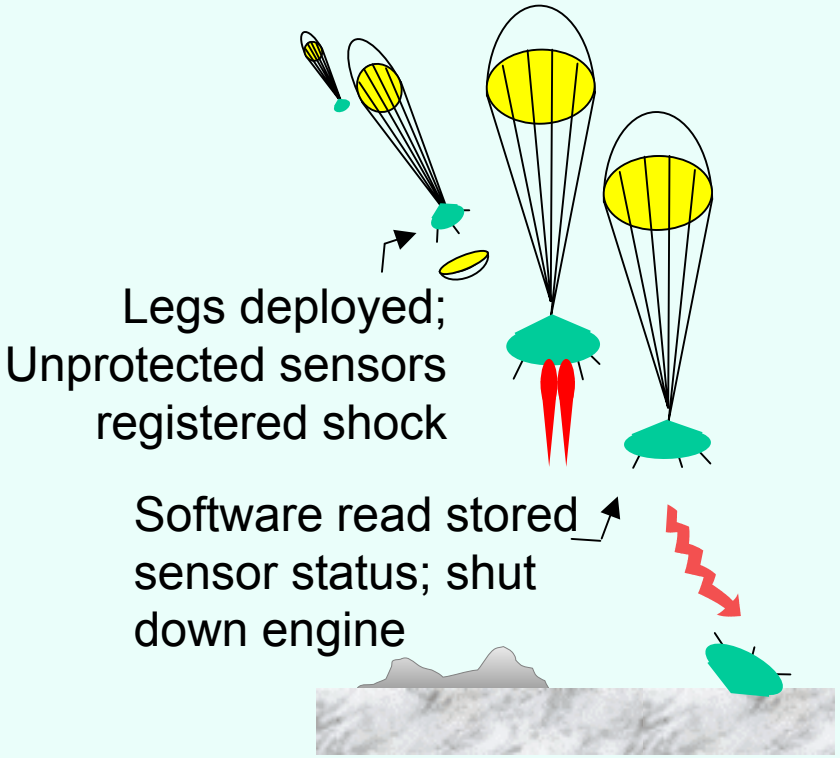
- **Do risk analysts know how to model human errors?**
- **How should the system safety goal be met?**

Mars Polar Lander Failure

Systems Requirement stated:

"The touchdown sensors shall be sampled at 100-Hz. The sample process shall initiate to keep processor demand constant.

However, sensor data shall not begin until 12 m above the surface. "



This requirement did not flow down to S/W

One requirement, one statement

Mars Climate Orbiter Failure

- Thruster firing model, reused from a previous mission, was in metric. Thruster vendor supplied data in English units.
- In the previous mission, engineers correctly inserted a 4.45 factor to convert lb-force to Newton.
- A new thruster was used, and the vendor's new (English unit) equation was pasted into the model without the 4.45 factor:
  - Spec was overlooked
  - Original code had no warning remarks
  - Ground software viewed as non-critical
  - Truth table, manually computed, had the same mistake
  - Tests not thorough

Compounded by navigation inadequacy, mistake turned deadly

More Examples

Terriers

- A torque coil had to be installed with a phase opposite of that of the other two coils.
- Software, reused from another mission, was not changed.
- The satellite could not power up because its solar arrays were steered away from the Sun.

STRV

- Software spec did not incorporate a requirement to command status relays by pulse.
- The constant voltage burned out a relay coil, disabling both receivers.
- Flaw not found on ground because the test set software correctly drove the relays with pulsed signals.

Examples from the Launch World

Titan IVB-32/Centaur

- A parameter manually entered into Centaur's avionics database had a missing exponent (i.e., misplacing a decimal point).
- Milstar Flight II-1 was lost after the upper stage became out of control.

Ariane 501

- The flight software, reused from Ariane 4, encountered an "operand error" because the new Ariane 5 flew a different trajectory.
- Avionics design presumed faults as due to random hardware problems—best handled by an equipment swap.
- The same error shut down the backup controller, and the perfectly flying rocket.

Clementine and NEAR

- Clementine failed to rendezvous with an asteroid:
  - Available processor was inadequate.
  - Hardware watchdog not used.
  - An overflow error fired the thrusters and disabled the CPU:
    * Software watchdog unable to function.
    * All fuel depleted, ending mission.
- NEAR, another attempt, studied Clementine history.
  - An engine abort triggered massive anomalies.
  - Thrusters fired thousands of times.
  - Hardware watchdog limited fuel loss and saved the mission.

Lesson Learning Saves Missions

Fools say that they learn by experience. I prefer to profit by others' experience.

Otto Bismarck