



Formalized Pilot Study of Safety-Critical Software Anomalies

Dr. Robyn Lutz and Carmen Mikulski

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The work was sponsored by the NASA Office of Safety and Mission Assurance under the Software Assurance Research Program led by the NASA Software IV&V Facility. This activity is managed locally at JPL through the Assurance and Technology Program Office

OSMA Software Assurance Symposium
September 4-6, 2002



Topics

- **Overview**
- **Results**
 - **Quantitative analysis**
 - **Evolution of requirements**
 - **Pattern identification & unexpected patterns**
- **Work-in-progress**
- **Benefits**



Overview

- **Goal: To reduce the number of safety-critical software anomalies that occur during flight by providing a *quantitative analysis* of previous anomalies as a foundation for process improvement.**
- **Approach: Analyzed anomaly data using adaptation of *Orthogonal Defect Classification (ODC)* method**
 - Developed at IBM; widely used by industry
 - Quantitative approach
 - Used here to detect patterns in anomaly data
 - More information at <http://www.research.ibm.com/softeng>
- **Evaluated ODC for NASA use using a *Formalized Pilot Study* [Glass, 97]**



Overview: *Status*



- Year 3 of planned 3-year study
 - Plan → Design → Conduct → *Evaluate* → *Use*
- FY'03 extension proposed to extend ODC work to *pre-launch and transition to projects* (Deep Impact, contractor-developed software, Mars Exploration Rover testing)
- Adapted ODC categories to operational spacecraft software at JPL:
 - Activity: what was taking place when anomaly occurred?
 - Trigger: what was the catalyst?
 - Target: what was fixed?
 - Type: what kind of fix was done?



Results: *ODC Adaptation*

- Adapted ODC classification to post-launch spacecraft Incident Surprise Anomalies (ISAs)

Activities	Triggers	Targets	Types
System Test	Software Configuration Hardware Configuration Start/Restart, Shutdown Command Sequence Test Inspection/Review	Ground Software	Function/Algorithm Interfaces Assignment/Initialization Timing
Flight Operations	Recovery Normal Activity Data Access/Delivery Special Procedure Hardware Failure	Flight Software	Function/Algorithm Interfaces Assignment/Initialization Timing Flight Rule
Unknown	Unknown	Build /Package	Install Dependency Packaging Scripts
		Ground Resources	Resource Conflict
		Info. Development	Documentation Procedures
		Hardware	Hardware
		None/Unknown	Nothing Fixed Unknown



Results: *Quantitative Analysis*

- **Analyzed 189 Incident/Surprise/Anomaly reports (ISAs) of highest criticality from 7 spacecraft**
 - Cassini, Deep Space 1, Galileo, Mars Climate Orbiter, Mars Global Surveyor, Mars Polar Lander, Stardust
- **Institutional defect database → Access database of data of interest → Excel spreadsheet with ODC categories → Pivot tables with multiple views of data**
- **Frequency counts of Activity, Trigger, Target, Type, Trigger within Activity, Type within Target, etc.**
- **User-selectable representation of results**
- **User-selectable sets of spacecraft for comparison**
- **Provides rapid quantification of data**

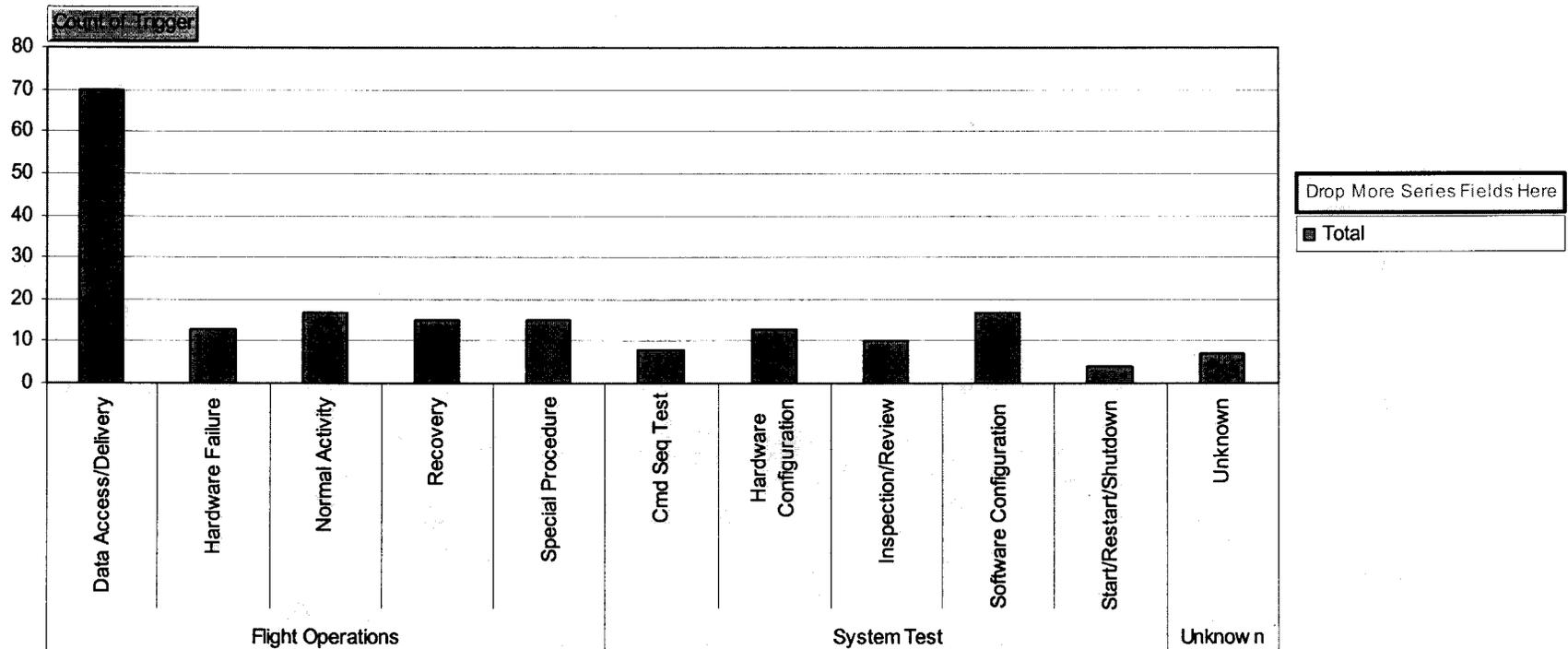


Results: Quantitative Analysis



PROJECT (All)

Distribution of Triggers within Activity



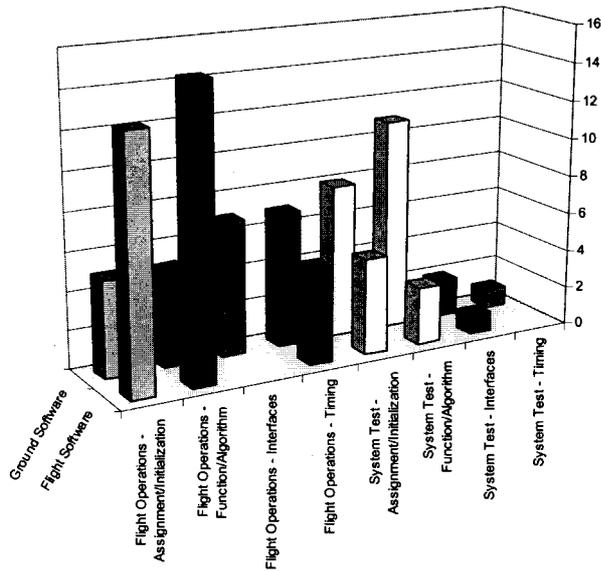
Activity Trigger



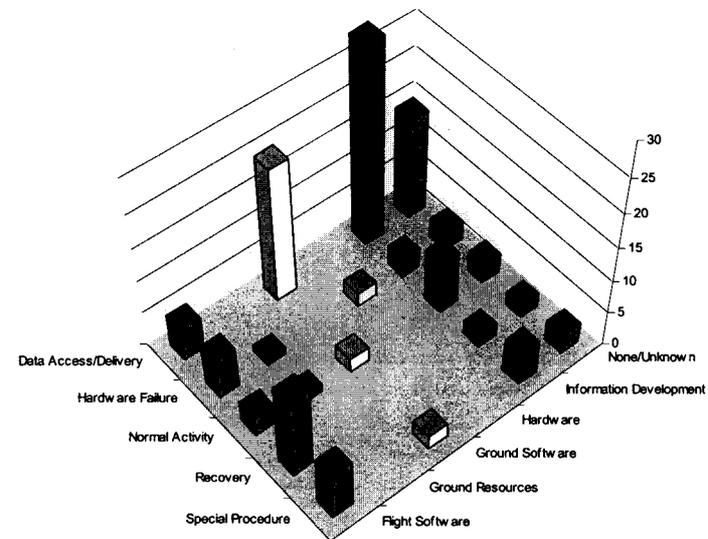
Results: *Quantitative Analysis*



Ground/Flight S/W vs. Type within Activity



Trigger vs. Target





Results: *Evolution of Requirements* **JPL**

California
Institute of
Technology

- **Anomalies sometimes result in changes to software requirements**
- **Finding:**
 - Change to handle rare event or scenario (software adds fault tolerance)
 - Change to compensate for hardware failure or limitations (software adds robustness)
- **Contradicts assumption that “what breaks is what gets fixed”**

Example: Damaged Solar Array Panel cannot deploy as planned

- Activity = Flight Operations (occurred during flight)
- Trigger = Hardware failure (Solar Array panel incorrect position--broken piece rotated & prevented latching)
- Target = Flight Software (Fixed via changes to flight software)
- Type = Function/Algorithm (Added a solar-array-powered hold capability to s/w)



Results: *Pattern Identification*

- **Sample Question: What is the typical signature of a post-launch critical software anomaly?**
- **Answer:**
 - **Activity = Flight Operations**
 - **Trigger = Data Access/Delivery**
 - **Target = Information Development**
 - **Type = Procedures**
- **Example: Star Scanner anomaly**
 - **Activity = occurred during flight**
 - **Trigger = star scanner telemetry froze**
 - **Target = fix was new description of star calibration**
 - **Type = procedure written**



Results: *Unexpected Patterns*



<i>Examples of Unexpected ISA patterns:</i>	<i>Process Recommendation:</i>	<i>Example (from spacecraft):</i>
22% of critical ISAs had <u>ground software</u> as Target (fix)	Software QA for ground software	Unable to process multiple submissions. Fixed code.
23% of critical ISAs had <u>procedures</u> as Type	Assemble checklist of needed procedures for future projects	Not in inertial mode during star calibration. Additions made to checklist to prevent in future.
Of these, 41% had <u>Data access / delivery</u> as Trigger	Better communication of changes and updates to operations	Multiple queries for spacecraft engineering and monitor data failed. Streamlined notification to operators of problems.
34% of critical ISAs involving system test had software configuration as Trigger (cause) ; 24% had hardware configuration as Trigger	Additional end-to-end configuration testing	OPS personnel did not have a green command system for the uplink of two trajectory-correction command files. Problems resulted from a firewall configuration change.



Work-In-Progress



- ***Assembling process recommendations*** tied to specific findings and unexpected patterns; in review by projects
- ***Working to incorporate ODC classifications*** into next-generation problem-failure reporting database (to support automation & visualization)
- ***Disseminating results:*** invited presentations to JPL Software Quality Improvement task, to JPL Mission Assurance Managers, to MER, informal briefings to other flight projects; at Assurance Technology Conference (B. Sigal), included in talk at Metrics 2002 (A. Nikora), at 2001 IFIP WG 2.9 Workshop on Requirements Engineering; papers in 5th IEEE Int'l Symposium on Requirements Engineering and The Journal of Systems and Software (to appear).



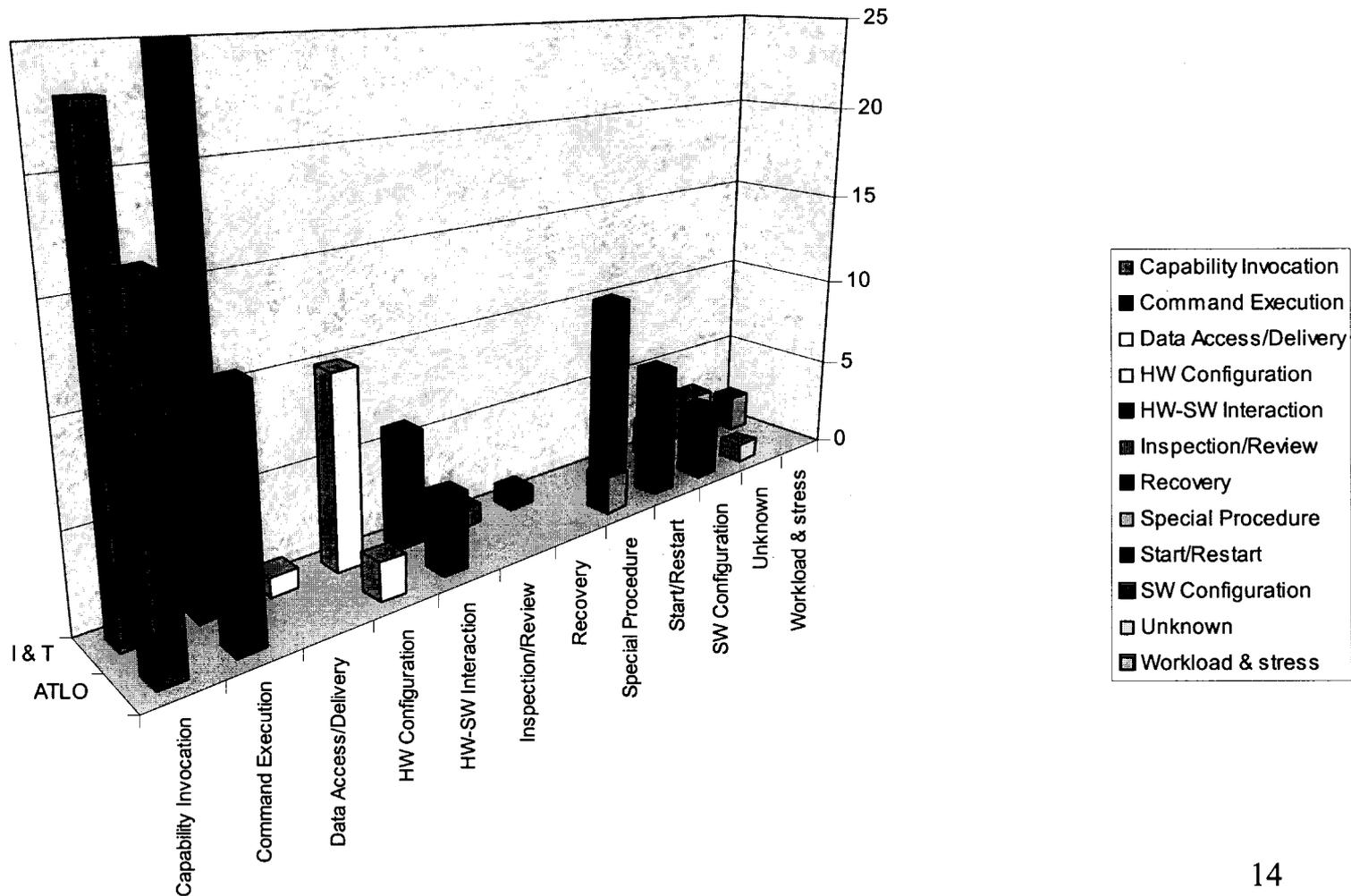
Work-In-Progress



- **Collaborating with Mars Exploration Rover to experimentally extend ODC approach to *pre-launch software problem/failure testing reports***
 - **Adjusted ODC classifications to testing phases (build, integration, acceptance)**
 - **Delivered experimental ODC analysis of 155 Problem/ Failure Reports to MER**
 - **Feedback from Project has been noteworthy**
 - **Results can support tracking trends and progress:**
 - **Graphical summaries**
 - **Comparisons of testing phases**
 - **Results can support better understanding of typical problem signatures**



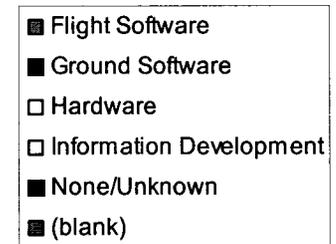
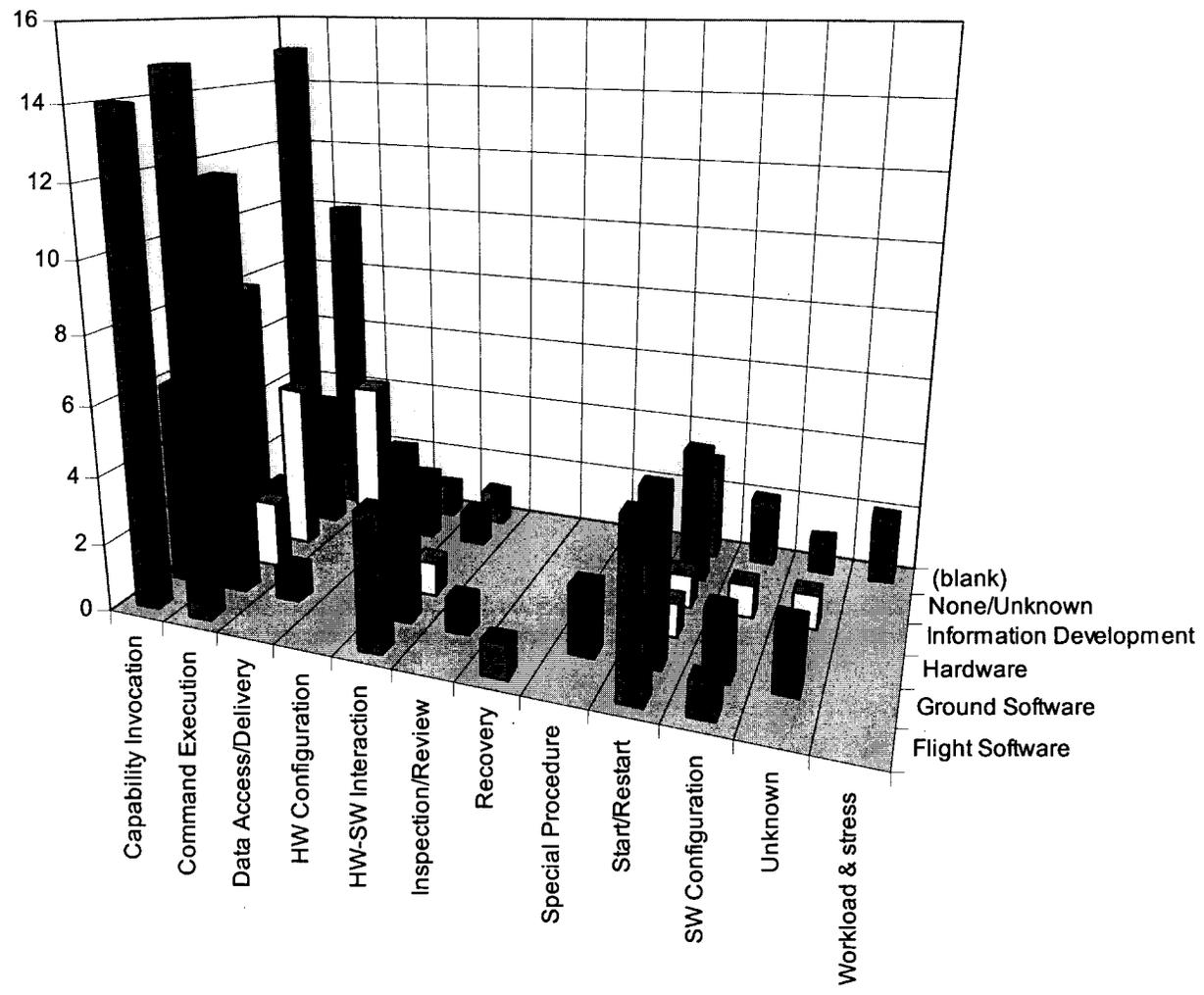
ODC Analysis (preliminary) MER Testing Problem/Failure Reports : *Trigger by Activity*





ODC Analysis (*preliminary*)

MER Testing Problem/Failure Reports: *Trigger by Target*





Benefits

- **User selects preferred representation (e.g., 3-D bar graphs) and set of projects to view**
- **Data mines historical and current databases of anomaly and problem reports to feed-forward into future projects**
- **Uses metrics information to identify and focus on problem areas**
- **Provides quantitative foundation for process improvement**
- **Equips us with a methodology to continue to learn as projects and processes evolve**