Resolving Requirements Discovery in Testing and Operations

Robyn Lutz
Jet Propulsion Lab
and Iowa State University
rlutz@cs.iastate.edu

Inés Carmen Mikulski
Jet Propulsion Lab
ines.c.mikulski@jpl.nasa.gov

RE 2003

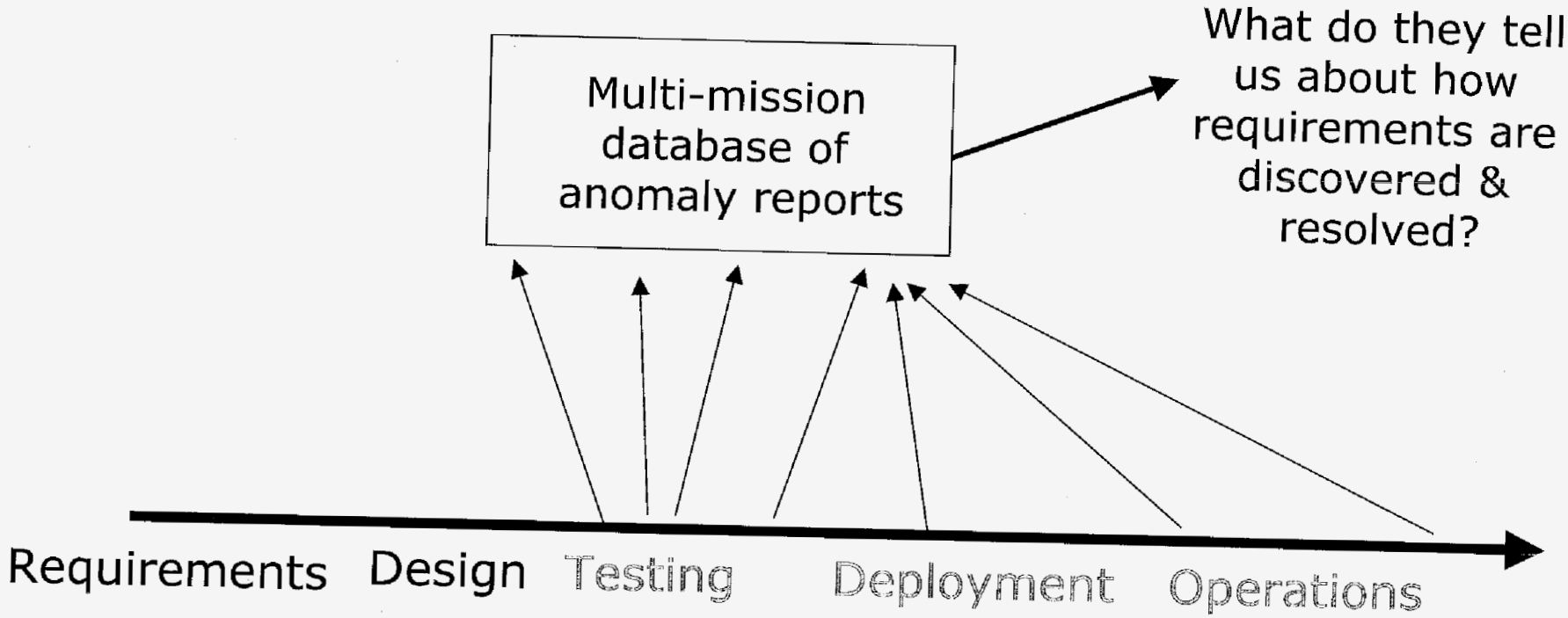
Monterey, CA September 10, 2003

The research described in this presentation was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. It was funded by NASA's Office of Safety and Mission Assurance, Center Initiative UPN 323-08. The first author's research is supported in part by National Science Foundation Grants CCR-0204139 and CCR-0205588.

Introduction

- Many anomalies found during testing and operations involve requirements discovery.
- Questions:
  1. What types of ***requirements discovery*** occur in operational systems and ***how are these discoveries resolved?***
  2. How do these mechanisms compare with those previously found in testing [Lutz & Mikulski, 2003]?

Approach


Multi-mission database of anomaly reports
What do they tell us about how requirements are discovered & resolved?
Requirements
Design
Testing
Deployment
Operations

Approach

- Analyzed anomaly reports from testing and operations
- Testing dataset: 463 anomalies from integration and system testing of twin Mars Exploration Rover spacecraft (arrive 1/04)
- Operations dataset: 189 critical anomalies from seven launched spacecraft
- 80 testing anomalies and 25 operational anomalies involved requirements discovery and resolution
- Adapted ODC (Orthogonal Defect Classification) [Chillarege et al., 1992] to characterize:
  - Activity: what was taking place when anomaly occurred?
  - Trigger: what was the catalyst?
  - Target: what was fixed?
  - Type: what kind of fix was done?

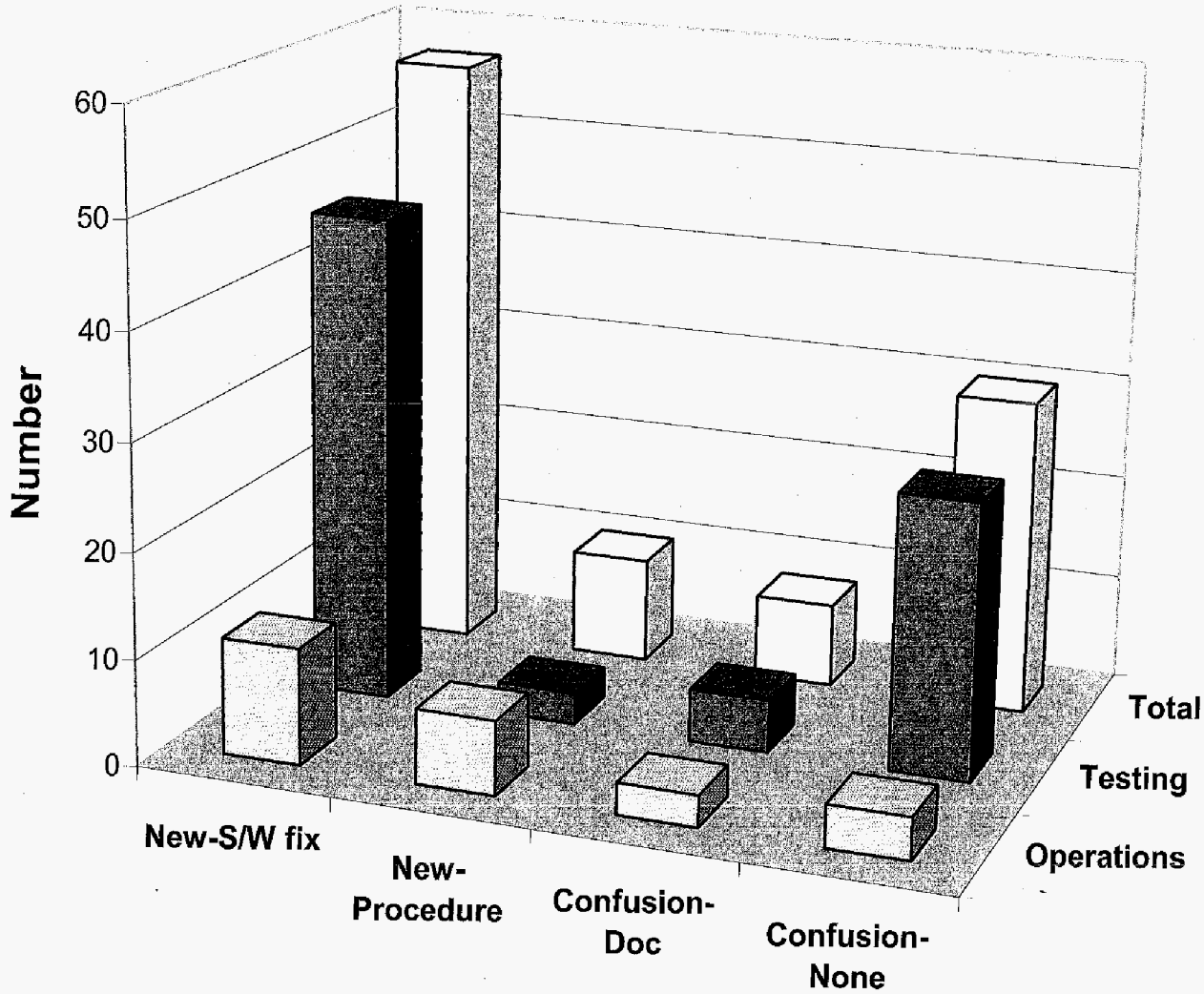
Related Work

- **"Continuous requirements management"** [Dubois & Pohl, 2003]
  - Proposals for changes to requirements continue post-deployment (updates and maintenance) [Bennett & Tajlich, 2000; Harker, Eason, & Dobson, 1992]
  - Contrast: anomaly-driven requirements changes are essential for current system (more analysis, less negotiation)
- **Defect analysis**
  - Incomplete requirements cause defects [Lutz, 1993; Leszak, Perry, & Stoll, 2002; Lauesen & Vinter, 2001]
  - Miscommunication of domain knowledge causes requirements defects in critical systems [Hanks, Knight, & Strunk, 2001; Weiss, Leveson, Lundqvist, Farid, & Stringfellow, 2001]
- **Goal-obstacle analysis** [van Lamsweerde & Letier, 2000]
  - Anomaly reports document obstacles to goals
  - Our results confirm importance of several obstacle subclasses and resolution strategies

Results

- 2 basic kinds of requirements discovery:
  - Discovery of ***new*** (i.e., previously unrecognized) requirements or requirements knowledge
  - Discovery of ***misunderstandings*** of (existing) requirements
- Reflected in ODC Target (what gets fixed) and ODC Type (nature of the fix):
  1. **Software change** (new requirement allocated to software)
  2. **Procedural change** (new requirement allocated to operational procedure)
  3. **Document change** (requirements confusion addressed via improved documentation)
  4. **No change needed**
- Very similar results for testing and operations

Results


Number
60
50
40
30
20
10
0
New-S/W fix
New-Procedure
Confusion-Doc
Confusion-None
Total
Testing
Operations

Types of requirements discovery & resolution

1. Incomplete requirements, resolved by change to software:
   - Testing anomaly: new requirement became evident for initial state of a component's state machine to wait for the associated motor's initial move to complete
   - Operational anomaly: new requirement became evident for software to compensate for noisy transducers that were causing frequent component resets

Types of requirements discovery & resolution

2. Unexpected requirements interactions, resolved by changes to operational procedures:
   - Testing anomaly: Software fault monitor issued redundant off commands from a particular state (correct but undesirable behavior). Corrective action was to prevent redundant commands procedurally by selecting limits that avoid that state in operations
   - Operational anomaly: when aerobraking maneuver erroneously performed twice, discovered that due to software being loaded to memory "too soon"; fixed by adding a procedure to prevent recurrence of configuration problem.

Types of requirements discovery & resolution

3. Requirements confusion, resolved by changes to documentation
   - Testing anomaly: Testing personnel incorrectly thought heaters would stay on as software transitioned from pre-separation to Entry/Descent mode; clarified in documentation.
   - Operational anomaly: Drop in battery power occurred when operational personnel misunderstood required behavior initiated by a command; clarified required behavior and associated command in operational flight rule.

Types of requirements discovery & resolution

4. Requirements confusion, resolved without change
   - Testing anomaly: Testers assumed commands issued when component was off would be rejected, but commands executed upon reboot. No fix needed; behavior correct.
   - Operational anomaly: Operational personnel assumed "stow" meant "close instrument cover when instrument not in use" and "deploy" meant "open instrument cover when instrument will be used." In fact, "stow" opens the cover, and "deploy" closes it. No fix needed.

- Does "no fix" suffice?
  - Mismatch between correct and expected behavior
  - Possible recurrence in operations with serious consequences
  - Long-lived systems→ more turnover, loss of requirements knowledge

Conclusions

- "False-positive" anomaly reports are invaluable
  - Insights into where software behavior is correct but unexpected can help us prevent recurrence
  - If requirement confused testers or users once, it may confuse others in future
  - Preclude by improved communication (rarely can change behavior)
- Patterns of requirements confusion exist
  - Some requirements are prone to misunderstanding, e.g., relative vs. absolute time
  - Working to understand why
- Allocating requirements to operational procedures may add risk
  - Explicit traceability from requirements should be maintained
- Product-line perspective is common
  - Anomaly reports for one system often address requirements implications for other, similar systems
  - Need better capture of these pointers from experts