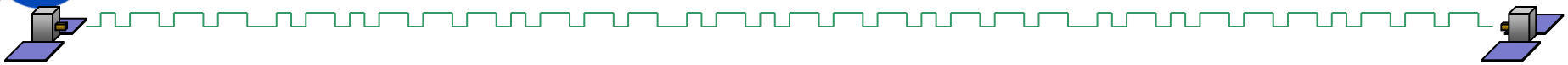

Analysis of Phoenix Anomalies and IV&V Findings Applied to the GRAIL Mission

March 5, 2012

Steve Larson
Jet Propulsion Laboratory
California Institute of Technology

Introduction

- Motivation
- GRAIL Project Overview
- Analysis of Phoenix IV&V and Post-Launch Anomalies
- Phoenix Lessons Applied to GRAIL
- Phoenix/GRAIL Post IV&V Comparison
- GRAIL Post-Launch Experience

Motivation

- GRAIL is a cost-capped mission
  - IV&V is recognized as a contributor to mission reliability, but funds were limited
  - Improving the efficiency of the IV&V effort by learning from earlier missions would increase the value added

GRAIL Project Overview

- GRAIL = Gravity Recovery And Interior Laboratory
- Launch September 2011
- Mission
  - Determine the structure of the lunar interior, from crust to core
  - Advance understanding of the thermal evolution of the Moon.
  - Extend knowledge gained from the Moon to the other terrestrial planets.
- Project Management: JPL
- PI: Dr. Maria Zuber, MIT
- Spacecraft: Lockheed Martin S&ES
- Instrument: JPL

Analysis of Phoenix IV&V Results

- NASA IV&V provided a dump of all issues written for the Phoenix project
  - 893 issues categorized into 16 bins
    - Multiple categorizations allowed
  - Analyzed according to whether the project responded by changing the affected artifacts or using as-is.
  - Looked for other patterns that would suggest areas for improvement

Analysis of Phoenix IV&V Results

- Code-related issues were much more likely to be accepted “as-is” by the project
- Issues that could be addressed by updating documentation were more likely to be fixed
- Possible explanations:
  - Code issues developed late in the development process
  - Lower effort barrier to changing documents vs. code (e.g., don’t need to regression test documents)

Description	Fix	UAI	Total
Array Bounds violation	55%	45%	1%
Conflicting Code Statements	20%	80%	2%
Coding Error	17%	83%	8%
Dead Code	30%	70%	5%
Design/Code Discrepancy	57%	43%	3%
Loss of Precision	20%	80%	1%
Memory Leak	0%	100%	0%
NULL Pointer	0%	100%	1%
Type Mismatch	25%	75%	2%
Uninitialized Variable	41%	59%	4%
Code-related Subtotal	**29%**	**71%**	**27%**
Document Discrepancy	53%	47%	17%
Design/Requirements Discrepancy	71%	29%	10%
Missing Requirement	72%	28%	6%
Requirement Not Verified	93%	7%	13%
Requirements Quality	63%	37%	20%
Requirements Trace	59%	41%	8%
Documentation-related Subtotal	**68%**	**32%**	**73%**
Combined Total	**57%**	**43%**	**100%**

Analysis of Phoenix IV&V Results

- Fix vs. Use As-Is decisions were correlated with issue severity
  - All Severity 2 issues were addressed by the project
    - IV&V agreed with the 2 UAI responses, and closed the issues
  - Severity 5 issues are the exception
    - Less numerous, may have been fixed as part of normal process or addressing higher severity issues
  - For the most part, IV&V agreed with UAI decisions
- Fix/Use as-is decisions were correlated with the analysis approach
  - Roughly 2/3 of issues found with automated tools were accepted as-is
  - The proportion was reversed in the case of manual analysis
  - Lockheed code relatively mature, often able to show that code would behave properly
    - Later in life cycle, more difficult to fix
  - Manual analysis more likely to be applied to tests, documentation
    - Earlier in life cycle and/or easier to fix

Severity	Disposition	Count	Percent
2	F	24	92.3%
2	UAI	2	7.7%
3	F	329	66.1%
3	UAI	169	33.9%
4	F	80	33.2%
4	UAI	161	66.8%
5	F	26	65.0%
5	UAI	14	35.0%

Issue Severities

1	Partial or complete mission failure
2	Adverse effect with no workaround
3	Adverse effect with a workaround
4	An inconvenience
5	Anything else

ToolUsed	Fix	UAI
Flexe-Lint, V8.00Q	26	54
Klocwork inSpect	19	49
Manual Analysis	415	239
Understand for C/C++	0	4

Analysis of Phoenix IV&V Results

- Payloads accounted for roughly ¾ of all issues, and had a somewhat higher proportion of higher severity issues
- Assuming that issue frequency is a predictor of in-flight performance, one might predict that payloads would account for the majority of post-launch FSW anomalies

CSCI	Severity				Totals	
	2	3	4	5	Count	%
GN&C	1	25	5	2	33	15.4%
I/O	0	28	5	0	33	15.4%
OS	0	16	9	1	26	12.1%
Spacecraft	5	38	50	3	96	44.9%
Telecom	0	14	12	0	26	12.1%
Spacecraft Subtotal	**8**	**124**	**85**	**11**	**214**	
Spacecraft % Distribution	**4%**	**58%**	**40%**	**5%**	**27%**	
Mardi	0	4	4	0	8	1.4%
MECA	4	130	44	4	182	30.8%
MET	0	39	9	3	51	8.6%
RA	5	48	32	7	92	15.6%
SSI/RAC	6	75	27	15	123	20.8%
TEGA	5	81	44	5	135	22.8%
Payloads Subtotal	**20**	**377**	**160**	**34**	**591**	
Payloads % Distribution	**3%**	**64%**	**27%**	**6%**	**73%**	

Phoenix Anomaly Analysis

- 369 unique post-launch ISA (Incident/Surprise/Anomaly) reports were analyzed
  - Binned into 8 categories of contributing factors
    - Factors identify where in the development & test process a defect was likely to have been introduced, or could have been corrected but was not
    - Multiple factors allowed
  - Binned according to whether issue was discovered on flight vehicle or on the ground
  - 31 ISAs determined to be in flight software (next slide)
    - 7 in spacecraft, 24 in payloads
      - » Mirrored IV&V ration of spacecraft/payload issues
    - Most ISAs concerned ground software and hardware, and were not included in the study

Contributing Factor Distribution

GRAIL Discovery

	Contributing Factors									
	Complexity	Heritage Process	Missing Requirement	Design	Inadequate Testing	Implementation	System Engineering	Insufficient Information	Total # Incidents	Occurred in Flight?
Spacecraft-	0%	29%	43%	57%	43%	29%	57%	14%	23%	86%
Payload-related	13%	4%	38%	17%	46%	58%	29%	13%	77%	50%
Combined	10%	10%	39%	26%	45%	52%	35%	13%	100%	58%

- Most ISAs had more than 1 contributing factor
  - For example, the heritage process introduced a defect in spacecraft battery control that testing should have caught, had fidelity to flight conditions been adequate
- Systems Engineering and S/W Design were leading causes of spacecraft issues
  - Inverse of what might be predicted from IV&V distribution (58% for Implementation vs. 37% for Requirements & Design)
- Code & Test were leading causes of payload issues
  - Also inverted from IV&V issue distribution(68% for Requirements & Design vs. 16% for Implementation)
- Effects of complexity is an industry-wide concern, but did not appear to be a dominant factor
  - Complexity-related issues in the payloads did not fit the classic “Normal Accidents” model

Predictive Skill

- IV&V broadly predicted (based on the data) that payloads would be the primary source of in-flight anomalies
  - This was borne out in flight
- Inverse relationship between distribution of IV&V issues and contributing factors for flight anomalies suggests additional analysis needs to be done to understand this relationship.

Phoenix Lessons Applied to GRAIL

- "Newness" is a risk
  - Payload issues dominated, and were all either first-of-a-kind or modifications to previous products
  - PBRA process used on GRAIL emphasized "newness" as a risk
- Product Line FSW should be approached differently than first-of-a-kind/low heritage software
  - Discussion of the effects of heritage allowed us to close issues more easily
- Unnecessary issues can be avoided by waiting for products to mature
  - Structured discussion of potential issues generated by review of early versions of FSW and requirements allowed us to resolve a large number of issues without excess formality
- Problems that escape both the developer and IV&V tend to be "difficult"
  - Hardware interfaces of greatest concern, received thorough IV&V review
  - However, proprietary nature of source data limited the analysis
- Augmenting the normal IV&V process with less formal analysis improves understanding and effectiveness
  - GRAIL able to vet a much larger number of potential issues and focus on important findings

IV&V Results—GRAIL/Phoenix

- Upon disposition of the last GRAIL issue, Phoenix (spacecraft only) and GRAIL issue distribution was analyzed
  - Overall roughly 10% drop in number of issues
    - Change could simply be due to different IV&V team or bundling strategies
  - Overall increase in Fix rate
    - Previous IV&V work on code base likely reduced number of false positives in code
    - Better communication eliminated more false positives in all categories
  - Big drops in Dead Code, Design/Code Discrepancy, Document Discrepancy categories
    - Better communication helped eliminate false positives and issues due to examining immature products
    - Dead code reduction may be due to prior IV&V work, but not analyzed for cause
  - A new category (Code/Requirements Discrepancy) was introduced for the GRAIL analysis
    - Better alignment with the way IV&V does their work
    - On Phoenix, these would have shown up in either Design/Code Discrepancy or Coding Error
  - Increase in Requirements Quality, Requirements Trace, Missing Requirements categories
    - IV&V changed approach, started with modeling and top-down requirements assessment (many more documents examined)
    - Different IV&V personnel may have also contributed

GRAIL Post-launch Experience

- No flight software anomalies in either spacecraft or instrument since launch
  - Instrument has been off most of the time
- One FSW patch for issue discovered prelaunch
  - Conformance with BAE RAD750 errata on register usage
  - No issues in 1000+ boot cycles prelaunch
  - Fixed out of an abundance of caution
- Two minor bugs found, but not fixed
  - Pyro firing
  - ACS parameter usage
- All three would have been very difficult for IV&V to catch
  - RAD750 errata nonconformance would have required expert level knowledge of the board
  - Pyro firing code error conformed to C language standard, subtle error in use of enums
  - ACS parameter usage would have required expert level knowledge of ACS algorithms and access to proprietary information

Conclusions

- Analysis of patterns in IV&V findings and their correlation with post-launch anomalies allowed GRAIL to make more efficient use of IV&V services
  - Fewer issues
  - Higher fix rate
  - Better communication
  - Increased volume of potential issues vetted, at lower cost
- Hard to make predictions of post-launch performance based on IV&V findings
  - Phoenix made sound fix/use as-is decisions
    - Things that were fixed eliminated some problems, but hard to quantify
  - Broad predictive success in one area, but inverse relationship in others