2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops

Toward Baselining Software Anomalies in NASA Missions

Lucas Layman[1], Marvin Zelkowitz[12], Victor Basili[123]
[1]Fraunhofer Center for Experimental Software Eng.
[2]University of Maryland, College Park
[3]King Abdulaziz University
College Park, MD 20740
llayman@fc-md.umd.edu; {mvz,basili}@cs.umd.edu

Allen P. Nikora
Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, CA 91109
Allen.P.Nikora@jpl.nasa.gov

***Abstract*—In this fast abstract, we provide preliminary findings from an analysis of 14,500 spacecraft anomalies from unmanned NASA missions. We provide some baselines for the distributions of software vs. non-software anomalies in spaceflight systems, the risk ratings of software anomalies, and the corrective actions associated with software anomalies.**

Keywords-anomaly; software defect;NASA;baseline

I. INTRODUCTION

The importance of software quality, safety, and reliability in spaceflight is increasing. Avionics, guidance, command handling, and fault detection are increasingly pushed into software systems to lighten spacecraft while infusing them with increasingly complex capabilities. In spaceflight systems (as in all software systems), *baselines* for software quality and reliability are needed to establish if system behavior falls within acceptable performance parameters, and to help guide development efforts intent on meeting software quality and reliability requirements.

An integral component to creating baselines for software quality and reliability are post-release failures. At NASA, *incident surprise anomalies (ISAs)* are erroneous or unusual observed behaviors from spacecraft (e.g. satellites, planetary explorers) that are reported by mission operations teams. These anomalies typically indicate an error in the spacecraft operation due to a component failure (hardware or software on the spacecraft or ground) or a procedural error by the operations team. These anomalies are recorded in *anomaly databases*, triaged, and addressed by the project team. We examined the anomaly databases from the NASA Jet Propulsion Lab (JPL) and Goddard Space Flight Center (GSFC), which contain over 39,000 anomalies. We analyzed a subset of 14,500 anomalies to baseline the number of software-related anomalies and to characterize them according to their corrective actions and risk ratings (mission impact and severity).

II. ANOMALY DATASET DESCRIPTION

The 14,500 anomalies selected for analysis are based on the quality of the anomaly information available for their respective missions in the database. Anomalies recorded prior to 1989 for JPL and prior to 1997 for GSFC were recorded either on paper or in legacy systems, often resulting in too much missing information for meaningful analysis. A summary of the datasets is provided in Table 1.

TABLE 1. ANOMALY DATA SET SUMMARY

Center	Total anomalies[a]	Subset analyzed	# Missions	Launch dates
JPL	13696	4621	9	1989-2005
GSFC	25320	9921	29	1997-2010

a. These numbers represent only the anomalies exposed to us in the current anomaly databases.

Each anomaly is indexed by a unique anomaly number for its database. The anomaly reports contain descriptions of the anomalous event, a description of the root cause, and a description of the corrective action. The anomalies also contain basic information, including the spacecraft they were observed in, the date the anomaly was observed, and the date the anomaly was closed due to either resolution or inactivity. The anomaly reports also contain fields categorizing the anomaly *cause* (e.g. software, hardware, unknown), however, these fields are not standardized and sometimes contain missing data. JPL anomaly reports also contain a *risk rating* (unacceptable risk, accepted risk, no significant risk, no risk), while GSFC anomalies contains a *mission impact rating* (catastrophic, major, minor, no impact).

III. SOFTWARE ANOMALY ANALYSIS

Identifying anomalies that are either caused or corrected by software is not trivial since the categorizations of the anomaly cause in the report were often missing or inaccurate. As such, we created a naïve search algorithm to look for software keywords in the anomaly reports, e.g. software, fsw (flight software), gsw (ground software), s/w, etc. Eliminating the false positives returned by the search algorithm is a manual process. Analysis suggests that the false positive rate for the search algorithm may be as high as 29% for GSFC and 6% for JPL. We ignore false negatives for now due to the size of the anomaly databases.

A. *How many anomalies are software-related?*

The number of software anomalies found in our analysis subsets is summarized in Table 2. The number of software anomalies is the number of anomalies containing a software search term in the cause description. For both data sets, the causes of approximately 25% of all anomalies appear to

978-0-7695-4928-6 2012

DOI 10.1109/ISSREW.2012.49
IEEE computer society

involve software. “Software-related anomalies” are anomaly reports where software is part of the corrective action, but not identified as a cause. These software-related anomalies may indicate a hidden software problem; if the software preventing the problem fails, then software may also be considered a cause of the anomaly.

TABLE 2. SOFTWARE ANOMALIES

Center	Subset analyzed	# S/W anom.	# s/w related anom.
JPL	4621	994	257
GSFC	9921	2328	634[a]

a. Partial estimate – several anomalies still being studied. Actual number probably around 300-400.

B. *How are software anomalies corrected?*

We manually inspected the 994 software anomalies from JPL and a random sample of 339 GSFC software anomalies to categorize the corrective action as one of the following:

- software – update, configuration change;
- hardware – replacing a broken ground component;
- operations – actions taken by the mission operations team, such as a procedural workaround or resetting connections;
- none – no action was taken by the mission team;
- unknown – the anomaly report did not specify a corrective action.

The distributions of the corrective action categories are presented in Table 3.

TABLE 3. SOFTWARE ANOMALY CORRECTIVE ACTIONS

Center	S/W	H/W	Ops	None	Unk.
JPL	577 (58%)	22 (2%)	200 (20%)	116 (12%)	79 (8%)
GSFC	112 (33%)	0	148 (44%)	27 (8%)	52 (15%)

These distributions highlight two interesting points. First, in ~10% of software anomalies, no action is taken when a software anomaly occurs. Either the anomaly was deemed unimportant, or the risks introduced by attempting to correct the problem outweighed the perceived residual risk. Second, operations fixes are common. Often the perception is that, if the operations team can find a workaround either through procedures or in resetting/rebooting the ground software, this is an acceptable if a software patch is perceived as risky.

C. *What is the risk distribution of software anomalies?*

The risk ratings for JPL anomalies (Table 4) indicate the residual risk, while the mission impact ratings for GSFC anomalies (Table 5) indicate the criticality without considering probability. The risk ratings are from the subsets of anomalies analyzed in Table 2.

In the JPL data, software anomalies account for 27% (44/164) of “Unacceptable” risk anomalies, and software accounts for ~30% of all anomalies with a risk rating. In the GSFC data, software anomalies account for 10% (27/261) of “Major” impact anomalies (Table 5), though software accounts for ~20% of all anomalies with a risk rating.

TABLE 4. RISK RATING DISTRIBUTIONS FOR JPL ANOMALIES

Risk rating	Un-acceptable	Accepted	Not significant	None	Total[a]
S/W anom.	44	251	540	45	880
% of S/W	5.7%	34.4%	53.7%	6.2%	
All anom.	164	995	1554	179	2892
% of all	5.00%	28.5%	61.4%	5.1%	

a.Not all anomalies specified a risk rating

TABLE 5. MISSION IMPACT RATING DISTRIBUTIONS FOR GSFC ANOMALIES

Mission impact	Catastrophic	Major	Minor	None	Total[a]
S/W anomalies	0	27	784	679	1490
	0%	1.8%	52.6%	45.6%	
All anomalies	0	261	4164	2853	7278
	0%	3.6%	57.2%	39.2%	

a.Not all anomalies specified a mission impact rating

IV. SUMMARY

This preliminary analysis raises some interesting points. First, software is the cause of approximately 23% of all anomalies (Table 2). Furthermore, software accounts for 27% of “unacceptable risk” anomalies for JPL and 10% of “major impact” anomalies for GSFC. Second, the software-related anomalies in Table 2 represent potentially hidden software risk should the software preventing the problem fail. Third, we need better classification of the anomaly reports to accurately identify the true numbers of software vs. non-software-related anomalies. This could be accomplished through mission staff training or through mission requirements for recording anomalies.

As with any bug tracking database, the central question is how to leverage the data to the benefit of the project team. We are exploring natural language processing techniques for triaging new anomaly reports according to risk rating and identifying potential root causes based on the anomaly descriptions, similar to those used in previous NASA datasets [1][2].

ACKNOWLEDGMENT

The work is sponsored by the NASA Office of Safety and Mission Assurance (OSMA) as part of the Software Assurance Research Program (SARP), grant #NNX08AZ60G. Part of this work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] T. Menzies and A. Marcus, “Automated severity assessment of software defect reports,” in 2008 IEEE International Conference on Software Maintenance, 2008, pp. 346–355.

[2] T. Menzies, R. Lutz, and C. Mikulski, “Better analysis of defect data at NASA,” in Proceedings of Fifteenth International Conference on Software Engineering and Knowledge Engineering, 2003.