Exploring Defect Categories for Infrastructure as Code

Akond Rahman
North Carolina State University
Raleigh, NC, USA
aarahman@ncsu.edu

Laurie Williams
North Carolina State University
Raleigh, NC, USA
lawilli3@ncsu.edu

1. INTRODUCTION

Infrastructure as code (IaC) technology refers to the process of automated and reproducible management of infrastructure that include operating systems, and software dependencies [1]. IaC technologies, such as Ansible, and Puppet aim to reduce deployment errors and deployment overhead through automation. Practitioners have observed multiple benefits of using IaC technologies including rapid delivery of software changes and fewer configuration errors [2]. Yet, IaC scripts can be complex in nature containing hundreds of lines of code, exposing IaC scripts to defects that are potentially difficult to debug. For example, a small change in infrastructure configuration incorrectly enabled Microsoft Azure blob storage, causing a large-scale system outage for Microsoft Azure [3]. Systematic investigation of what categories of defects occur can help in understanding the nature of IaC defects and take appropriate actions to mitigate those defects. *The goal of this poster is to aid software practitioners in understanding of infrastructure as code (IaC)-related defects through categorization of defects that occur in IaC scripts.*

In this research study we investigate the categories of defects that occur in Puppet scripts. Puppet is a configuration management tools to implement IaC. We first collect 564 commit messages of 11 open source repositories that contain IaC scripts. Next, we use the orthogonal defect classification (ODC) technique [4] on 564 commit messages extracted from the 11 repositories.

2. METHODOLOGY

Initially, we collected 23 repositories used by Wikimedia [5]. Of the 23 repositories we included the repositories for which we observed two commits per month, and for which at least 11.1% of all the files were Puppet scripts. Altogether, we included 11 repositories that included 564 commit messages where at least one Puppet file was changed. For categorization we selected all 564 messages to which we applied nine defect categories. Eight of the nine defect categories originally came from ODC, and we added the 'Other' category to categorize the messages that do not belong to any of the eight categories. If the message does not correspond to an error then we mark it as 'No error'. We briefly describe the nine categories as following:

Algorithm (AL) errors include efficiency or correctness problems that affect the task and can be fixed by implementing an algorithm or local data structure. **As**signment (AS) errors refer to syntax errors and missing configurations. **B**uild/package/merge (B) error occurs due to mistakes in library systems, management of changes, or version control. **C**hecking (C) error address program logic that has failed to properly validate data and values before they are used. **D**ocumentation (D) errors affect publications and maintenance notes. **F**unction (F) error is an error that requires a formal design change. **I**nterface (I) errors are errors in interacting with other components, modules or device drivers via macros, and call statements. **T**iming or serialization (T) errors are those that are corrected by improved management of shared and real-time resources. If an error that does not belong to the above-mentioned eight categories is categorized as **o**ther error (O).

In our research study we used 57 voluntary raters who were students of a graduate level course. The 564 messages were randomly distributed amongst the 57 raters in a manner such that each of the commit messages was rated by at least two raters. The raters categorized their assigned messages using the nine categories mentioned in this section. For a message if the raters disagreed then the first author of the paper resolved the disagreement.

3. RESULTS

We first identified which of the 564 commit messages are related to defects. Of the 564 messages, we identified 178 messages to be related to defects. We applied the nine categories on the 178 messages. We present the categories of the 178 messages in Figure 1. As shown in Figure 1, the highest and lowest percentages of messages belong to the categories AS, and O, respectively.

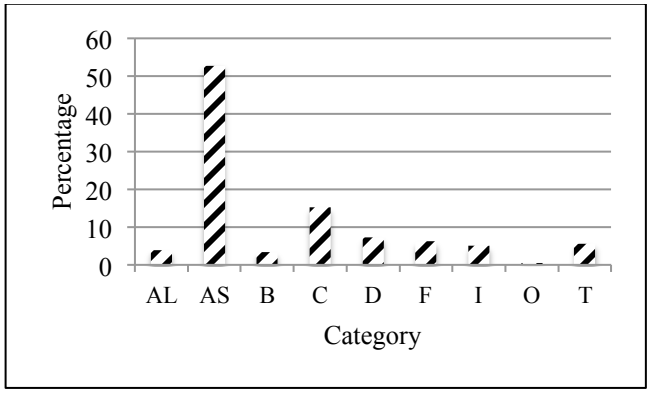


Figure 1: Defect categories of the 564 messages

4. REFERENCES

[1] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*, 1st ed. Addison-Wesley Professional, 2010.

[2] Gartner, "The Value of IT Automation," *The Value of IT Automation*, 2016. .

[3] J. Zander, "Final Root Cause Analysis and Improvement Areas: Nov 18 Azure Storage Service Interruption," *Final Root Cause Analysis and Improvement Areas: Nov 18 Azure Storage Service Interruption*, 17-Dec-2014. .

[4] R. Chillarege *et al.*, "Orthogonal defect classification-a concept for in-process measurements," *IEEE Trans. Softw. Eng.*, vol. 18, no. 11, pp. 943–956, 1992.

[5] Wikimedia, "Wikimedia Code Review," *Wikimedia Code Review*. .

Copyright Information: Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. *Symposium and Boot Camp on the Science of Security HotSoS'17*, April 4-5, 2017, Hanover, MD, USA.
Copyright 2010 ACM 1-58113-000-0/00/0010 …$15.00.