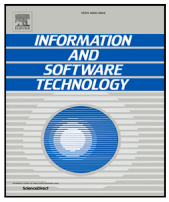
Information and Software Technology 155 (2023) 107105

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



An empirical study on bugs in JavaScript engines

Ziyuan Wang *, Dexin Bu, Nannan Wang, Sijie Yu, Shanyi Gou, Aiyue Sun

School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

ARTICLE INFO

Keywords:
Empirical study
JavaScript engine
Software bug
SpiderMonkey
Chakra
V8

ABSTRACT

Context: JavaScript is a prototype-based dynamic type scripting language. The correct running of a JavaScript program depends on the correctness of both the program and the JavaScript engine.
Objective: An in-depth understanding of the characteristics of bugs in JavaScript engines can help detect and fix them.
Methods: We conduct an empirical study on the bugs in three mainstream JavaScript engines: V8, SpiderMonkey, and Chakra. Such an empirical study involves 19,019 bug reports, 16,437 revisions, 805 test cases, and root causes of randomly selected 540 bugs.
Results: (1) The Compiler and the DOM are the most buggy component in V8 and SpiderMonkey, respectively. Most of the source files contain only one bug. (2) The scales of the testing programs that reveal bugs are usually small. Most bug fixes involve only limited modifications since the number of modified source files and lines of code modified are small. (3) Most bugs can be fixed within half a year (80.33% for V8 and 91.9% for SpiderMonkey). Only 4.33% of SpiderMonkey bugs need more than a year to fix. Bugs in SpiderMonkey are usually fixed faster than bugs in V8. (4) High priority tends to be assigned to Infrastructure bugs in V8 and Release Automation bugs in SpiderMonkey. The duration of bugs is not strictly correlated with their priorities. (5) Semantic bugs are the most common root causes of bugs. And among semantic bugs, the processing bugs, missing features bugs and function call bugs are more than others.
Conclusion: This study deepens our understanding of bugs in JavaScript engines, and empirical results could indicate some potential problems during the detecting and fixing of bugs in JavaScript engines, assist developers of JavaScript engines in improving their development quality, assist maintainers in detecting and fixing bugs more effectively, and suggest users of JavaScript evade potential risks.

1. Introduction

JavaScript is a prototype-based dynamic type scripting language. It supports multi-paradigms, including object-oriented, imperative, declarative, and functional programming. Its ease of learning and widespread use in many fields have made it in the top ten popular programming languages for a long time, according to TIOBE (https://www.tiobe.com/tiobe-index/). Although it is best known as a scripting language for developing Web pages, JavaScript is also used in many non-browser environments. JavaScript is an interpreted programming language, so the engines should load and interpret code at runtime. Therefore, the correct running of a JavaScript program depends on the correctness of both the program and the JavaScript engine. There are inevitably many bugs in JavaScript engines, which are more harmful than bugs in normal applications. Understanding and fixing the bugs in JavaScript engines is of great significance for ensuring the correct execution of JavaScript applications and even promoting the healthy development of the JavaScript programming language.

JavaScript engine is a virtual machine that specializes in JavaScript scripting and is typically released and embedded in a Web browser. Popular JavaScript engines include V8, SpiderMonkey, and Chakra, with 84% of the market. If the JavaScript application does not run correctly due to bugs in the JavaScript engine, these problems are difficult for developers of JavaScript applications to solve at the application level. It is very challenging to distinguish whether the root cause of application failure is a bug in the application code or a bug in the JavaScript engine. Therefore, exploring the bugs in JavaScript engines can help the developers, testers, and maintainers of the JavaScript engines to avoid, detect and fix bugs effectively and efficiently, and also help the developers of JavaScript applications avoid the risks caused by bugs in the engine.

The open-source projects could provide necessary conditions for the empirical study of the bugs in those projects. There have been a number of empirical studies on software bugs [1–4], but none focuses on the bugs in JavaScript engines. For example, Sun et al. studied the bugs of

* Corresponding author.
E-mail address: wangziyuan@njupt.edu.cn (Z. Wang).

https://doi.org/10.1016/j.infsof.2022.107105
Received 6 March 2022; Received in revised form 4 November 2022; Accepted 7 November 2022
Available online 12 November 2022