ANALYZING THE IMPACT ON IDENTIFIABLE DEFECTS IN A CODE INDEPENDENT OF TYPESCRIPT

G.M. Deshan Madurajith
(199343H)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

June 2022

ANALYZING THE IMPACT ON IDENTIFIABLE DEFECTS IN A CODE INDEPENDENT OF TYPESCRIPT

G.M. Deshan Madurajith
(199343H)

Thesis/Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

June 2022

DECLARATION OF THE CANDIDATE & SUPERVISOR

I declare that this is my own work and this thesis does not incorporate without acknowledgment of any material previously submitted for a Degree or Diploma in any other universities or institutes of higher learning and to the best of my knowledge and belief does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name of Candidate: G.M. Deshan Madurajith

Signature: ..... Date: ……

The above candidate has carried out research for the Master' thesis under my supervision.

Name of the supervisor: Prof. Indika Perera

Signature: ..... Date: …….

Abstract

Javascript has become the most widespread programming language used to build software for many platforms, including web, mobile, backends, hardware, and desktop applications. It is a dynamically typed programming language that gives freedom to ignore types and build applications quickly. Still, Javascript is a poor language for identifying bugs at the compile time and maintaining large applications because of the dynamic behavior. The developers can use Typescript to add type syntaxes on top of JavaScript as a solution.

However, not enough empirical studies exhibit how Typescript impacts detecting detects in applications. We decided to follow an empirically quantified "what-if" style of experimentation. We collected merged javascript bug-fix pull requests from selected open-source projects. Then we selected candidate bugs by applying our predefined criteria such as discarded pull requests with more than five files, code-refactoring, and configuration changes. We manually added Typescript annotation to the buggy code base and checked whether Typescript could detect the defects at the compile type. We assessed 500 bug-fix pull requests over five projects and identified that using Typescript over Javascript could have prevented 22.7% of bugs.

According to our literature review, this is one of the few studies related to Typescript identifying bug impact. We believe this study will be significant evidence to consider using Typescript over Javascript in the future to reduce the significant number of bugs. This result will influence developers to adapt to the Typescript from Javascript. However, Typescript is not a silver build for Javascript because developers have to add extra code and complexity to the codebase. So, More research on Typescript is needed. In the future, we plan to explore the cognitive complexity of applying Typescript and the number of required lines to annotate. Furthermore, we plan to use a Typescript converter for annotating to increase the number of sample bugs to analyze.

Keywords: Typescript, Javascript, static typed, dynamic typed

ACKNOWLEDGEMENT

First, I am indebted to my thesis advisor, Prof. Indika Perera. He helped to finalize the scope and advised me on how the contents and research should be. I also thank all the lecturers who teach me because those subjects helped me a lot to choose this research. Finally, yet importantly, I want to mention the role of my parents in my journey. They add unconditional support to manage my career & studies in a free mindset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES