Do TypeScript Applications Show Better Software Quality than JavaScript Applications? A Repository Mining Study on GitHub

Bachelor Thesis

# Organization

|  |  |
| --- | --- |
| Examiner: | Prof. Dr. Stefan Wagner |
| Supervisors: | Dr. Justus Bogner |
| Student: |  |
| Timeframe: | 2021-05-10 – 2021-11-10 |

# Context & Motivation

In 1995, JavaScript [1] was developed and introduced for small client-side tasks in the browser. JavaScript spread rapidly with the growth of the internet and is now the most widely used programming language according to a Stackoverflow survey [2]. Reasons for this include its versatility, flexibility, and ease of use. TypeScript [3], a superset of JavaScript, is also becoming more popular. The main difference is its static type system and its class-based object orientation [4]. However, there is a lack of sufficient empirical evidence to support the claim that TypeScript leads to better software quality than JavaScript. The difference between dynamically typed and statically typed languages has been studied occasionally, but results are not conclusive. While some studies claim that statically typed languages have a positive impact on code quality [5], others claim the opposite [6]. There is a lack of studies that directly compare JavaScript and TypeScript projects in terms of software quality on a larger scale.

# Objectives

The goal of this study is therefore to empirically analyze a large set of JavaScript and TypeScript applications. With the help of this data collection, comparative insights into the software quality of the two programming languages should be possible. Additionally, potential influencing factors on the analyzed properties in TypeScript projects should be identified. The quality aspects to be analyzed and the detailed research questions will be defined by the student.

# Methods

The research should be conducted as a mining software repository (MSR) study [7][8] using a large number of open-source projects, e.g. on GitHub [9]. Data collection should be conducted with appropriate tools (e.g. static analysis tools) and automated as much as possible to achieve the best possible reproducibility. For the analysis, suitable techniques could be hypothesis testing, correlation, or regression. The detailed study design will be created by the student.

# References

[1] <https://www.javascript.com>

[2] <https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages>

[3] [https://www.typescriptlang.org](https://www.typescriptlang.org/)

[4] Remo H. Jansen, “Learning TypeScript”, Packt Publishing, 2015, pp. 2 f.

[5] Z. Gao, C. Bird and E. T. Barr, "To Type or Not to Type: Quantifying Detectable Bugs in JavaScript," 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), 2017, pp. 758-769, doi: 10.1109/ICSE.2017.75.

[6] Stefan Hanenberg, “An experiment about static and dynamic type systems: doubts about the positive impact of static type systems on development time” SIGPLAN Not. 45, 10 (October 2010), 2010, 22–35, DOI: <https://doi.org/10.1145/1932682.1869462>

[7] A. E. Hassan, “The road ahead for Mining Software Repositories,” in 2008 Frontiers of Software Maintenance, 2008, pp. 48–57.

[8] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, “The promises and perils of mining GitHub,” in Proceedings of the 11th Working Conference on Mining Software Repositories - MSR 2014, 2014, pp. 92–101.

[9] <https://github.com>