A Mixed-Methods Study of Novice Programmer Interaction with Python Error Messages

Rachel D'souza
Dept. of Computer and Mathematical Sciences
University of Toronto Scarborough
rachel.dsouza@mail.utoronto.ca

Mahima Bhayana
Dept. of Computer and Mathematical Sciences
University of Toronto Scarborough
mahima.bhayana@mail.utoronto.ca

Marzieh Ahmadzadeh
Dept. of Computer and Mathematical Sciences
University of Toronto Scarborough
marzieh.ahmadzadeh@utoronto.ca

Brian Harrington
Dept. of Computer and Mathematical Sciences
University of Toronto Scarborough
brian.harrington@utoronto.ca

ABSTRACT

The ability to interpret error messages in order to find and fix bugs is an essential skill for novice programmers. Unfortunately, the technical language of most error messages can hinder the progress of CS1 students and can lead to feelings of confusion and frustration. A potential intervention for CS educators is the use of enhanced error messages that utilise natural-language geared towards novice learners, but the current discourse in CS Education regarding benefits of such messages is inconclusive. In this paper, we describe a planned, semi-controlled experiment running parallel to a CS1 course at the University of Toronto Scarborough. The study aims to build upon existing work in quantifying the effects of enhanced error messages by incorporating additional quantitative metrics, usability surveys, student feedback, and semi-structured interviews. The additional methods serve to measure not only the effects on error recovery, but student satisfaction, sense of frustration and overall attitude towards error messages and debugging.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; **Computer science education**; **CS1**.

KEYWORDS

CS1, error messages, debugging, usability, compilation behaviour

ACM Reference Format:
Rachel D'souza, Mahima Bhayana, Marzieh Ahmadzadeh, and Brian Harrington. 2019. A Mixed-Methods Study of Novice Programmer Interaction with Python Error Messages. In *Western Canadian Conference on Computing Education (WCCCE '19), May 3–4, 2019, Calgary, AB, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3314994.3325090

WCCCE '19, May 3–4, 2019, Calgary, AB, Canada

ACM ISBN 978-1-4503-6715-8/19/05…$15.00
https://doi.org/10.1145/3314994.3325090

1 INTRODUCTION

Error detection and correction are critical skills that novice programmers need to develop in order to succeed in programming. For students with little experience in programming, error messages can be an important source of guidance. Unfortunately, the technical syntax and brevity of most error messages render them inadequate for use as a learning tool. Thus, error messages can hinder progress and serve as a source of discouragement for novice programmers.

In this proposal, we aim to assess the effects of error messages written in a more accessible natural language on introductory students. We will evaluate the impact of enhanced error messages on student satisfaction, frustration, error correction time, number of submissions required to fix errors, and probability of repeating the same error in future. We have also implemented a web application capable of tracking these variables of interest and delivering either enhanced error messages or default Python error messages to designated users, tracking subsequent submissions, and categorizing changes made in response to specific error messages.

2 BACKGROUND

There is a great deal of research focusing on the compilation behaviour of novice programmers - the cycle of receiving feedback and making changes to the code accordingly. The ability to use such feedback effectively for debugging is crucial in the development of novice programmers [1, 7] but existing research has shown that current error messages (EMs) are inadequate for novices [3] and can act as a source of frustration [12].

Given that having a positive attitude and good debugging skills is important in the development of novices[1, 12], many researchers have attempted to investigate if "enhanced" error messages (EEMs) can better support CS1 students, with inconclusive or contradictory results. Denny et al.[6] found no statistically significant difference between students who interacted with EEMs and those interacting with EMs when looking at the number of submissions required to resolve an error. In contrast, Becker found that using less technical terms and providing solutions in EEMs reduced the likelihood of students repeating the same mistake, and reduced student frustration [4]. While Petit et al.[10], were unable to find a significant benefit of EEMs, they did receive positive feedback from both students and instructors.

Some studies postulate that the culmination of inconclusive research is rooted in the fact that the effects of EEMs can be difficult to quantify [11] and research focus should shift to measuring their usability. For example, in a more recent controlled experiment by Becker et al.[5], students exclusively interacted with EMs or EEMs for the duration of a CS1 course. Differences between the two groups were evaluated through a syntax error debugging test conducted in week 10 of the course. There was some evidence for the positive effects of EEMs in some metrics (e.g., number of errors rectified) and evidence that there is no statistically significant difference between the effects of EEMs and EMs in others (e.g., student test scores). The researchers thus proposed that the contradiction might be indicative of the subtle effects of EEMs and that focus might be shifted to how these influence the attitudes and behaviours of learners instead.

3 METHODOLOGY

To investigate the effects of EEMs on novice programmers, students will be recruited from a CS1 course at the University of Toronto Scarborough. The course is geared towards first year students intending to enter a major or specialist program in Computer Science and does not require any previous knowledge of programming. We expect to attract approximately 200-300 student volunteers from the incoming cohort.

A multi-phase study with a combination of previously described methodologies for quantifying the effects of EEMs on CS1 students [5, 8] and HCI research methods will be conducted. Initially, a pilot study will be conducted to evaluate the usability of the tool and suitability of drafted EEMs. Once suitable EEMs have been agreed upon, subjects will be randomly distributed into a control group interacting with EMs and an intervention group interacting with EEMs in separate computer laboratories. Both groups will be invited to use the application in a series of workshops running in parallel with the course every two weeks. During the workshops, students will be asked to debug erroneous code and write code in response to a prompt. In debugging questions, students will be prompted to identify the error before running the code, and then asked to repair the code after seeing an error message. During coding questions, errors will be tracked along with subsequent submissions. The changes made in response to the given error message will be assessed in accordance with the rubric described in Marceau et al. [9] with slight modification: instead of recording keystrokes following an error message call, changes will only be tracked in code after a run attempt. The application will track error correction time, number of submissions required to fix errors, total time spent on each question, total time spent on task, number of questions correctly solved, and the repetition of the same error message in subsequent problems for a given user identified by a non-descriptive login.

Data collection will also include demographic information, the user's perceived satisfaction with interacting with the error messages and the user's overall attitude towards encountering challenges. A pre-test will allow participants to self-identify gender, cultural background (i.e international student or domestic student), prior programming experience and their first language. At the end of each workshop, a modified System Usability Scale survey [2] written for "error messages" not "systems" will be conducted. Lastly, at the conclusion of the experiment, semi-structured interviews will be conducted with samples from both groups to assess general attitudes toward programming, debugging, and computer science in general.

4 DISCUSSION

The design of this study builds on Becker et al.'s 2018 study by answering similar research questions and adding more questions about student satisfaction and attitude. We make some changes to the design of the paper by evaluating students on a bi-weekly basis rather than a single point in the semester and we will use Python instead of Java. Focus will also be given to the usability of the EMs and EEMs by conducting a simple usability survey following each user's interaction with them at a workshop, and assessing student attitudes and perceptions through general surveys and post study semi-structured interviews.

Replacing standard technical error messages with natural language enhanced error messages specifically designed for novice programmers is a low-cost intervention that could have a positive impact in the classroom. This study attempts to ascertain whether these enhanced error messages have an impact on students ability to find and fix errors, as well as their level of frustration and overall perceptions of debugging and error correction.

REFERENCES

[1] Marzieh Ahmadzadeh, Dave Elliman, and Colin Higgins. 2005. An analysis of patterns of debugging among novice computer science students. In *Acm sigcse bulletin*, Vol. 37. ACM, 84–88.

[2] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[3] Titus Barik, Justin Smith, Kevin Lubick, Elisabeth Holmes, Jing Feng, Emerson Murphy-Hill, and Chris Parnin. 2017. Do developers read compiler error messages?. In *Proceedings of the 39th International Conference on Software Engineering*. IEEE Press, 575–585.

[4] Brett A Becker. 2016. An effective approach to enhancing compiler error messages. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 126–131.

[5] Brett A Becker, Kyle Goslin, and Graham Glanville. 2018. The Effects of Enhanced Compiler Error Messages on a Syntax Error Debugging Test. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, 640–645.

[6] Paul Denny, Andrew Luxton-Reilly, and Dave Carpenter. 2014. Enhancing syntax error messages appears ineffectual. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*. ACM, 273–278.

[7] Sandy Garner, Patricia Haden, and Anthony Robins. 2005. My program is correct but it doesn't run: a preliminary investigation of novice programmers' problems. In *Proceedings of the 7th Australasian conference on Computing education-Volume 42*. Australian Computer Society, Inc., 173–180.

[8] David Liu and Andrew Petersen. 2019. Static Analyses in Python Programming Courses. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM, 666–671.

[9] Guillaume Marceau, Kathi Fisler, and Shriram Krishnamurthi. 2011. Measuring the effectiveness of error messages designed for novice programmers. In *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 499–504.

[10] Raymond S Pettit, John Homer, and Roger Gee. 2017. Do Enhanced Compiler Error Messages Help Students?: Results Inconclusive.. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, 465–470.

[11] James Prather, Raymond Pettit, Kayla Holcomb McMurry, Alani Peters, John Homer, Nevan Simone, and Maxine Cohen. 2017. On Novices' Interaction with Compiler Error Messages: A Human Factors Approach. In *Proceedings of the 2017 acm conference on international computing education research*. ACM, 74–82.

[12] V Javier Traver. 2010. On compiler error messages: what they say and what they mean. *Advances in Human-Computer Interaction* 2010 (2010).