

# StackDoc - A Stack Overflow Plug-in for Novice Programmers that Integrates Q&A with API Examples

Akhila Sri Manasa Venigalla, Chaitanya S. Lakkundi, Vartika Agrahari, Sridhar Chimalakonda  
Department of Computer Science and Engineering  
Indian Institute of Technology  
Tirupati, India  
{cs18m017, cs18s502, cs18m016, ch}@iittp.ac.in

**Abstract**—There is a tremendous increase in the use of online coding platforms, courses and walkthrough tutorials to learn programming today. Stack Overflow, a Q&A website of crowd-sourced knowledge on programming is one of the popular platforms that developers and learners use to ask and answer Q&As related to programming. However, novice programmers often face difficulties in understanding the answers as they may contain new terminologies, function calls and attributes of certain technology or programming language. Researchers have proposed different ways to augment Stack Overflow in the literature, but to the best of our knowledge, there is no work that exists to augment Stack Overflow posts with definitions of API calls and relevant examples. To this end, we propose *StackDoc*, a prototype plug-in that augments Stack Overflow with definitions and examples of API calls in the questions and answers with the goal of helping novice programmers. We did a preliminary survey with 20 students of various universities, novice to Java and 85% of the users reported positive experience with the plugin.

**Keywords**- Stack Overflow; API call; Novice Programmers; plug-in; Learning

## I. INTRODUCTION

Programming is considered as one of the fundamental skills in the 21st century. With the emergence of tremendous web content and novel technologies, one can learn programming through competitive programming or through online courses or with the help of Q&A sites. Developers today extensively rely on using code snippets and answers present Q&A websites like Stack Overflow [1]. It has been observed by researchers, that developers use Q&A sites such as Stack Overflow to clarify their doubts [2]. These questions on Stack Overflow could refer to debugging a code or adding new features to a given code [3] [2]. To raise their level of understanding on the code snippets provided on Stack Overflow, users generally search for definition, usage and importance of certain function calls that are used in the code snippets [3].

It was also observed that code examples are searched in Android API documentation by programmers [4]. However, most of the novice programmers might not be aware of all the functions and attributes used in code snippets, making code reuse difficult for them [5] [6] [7]. If these Q&A sites can serve as knowledge reserves, beginners might be able to understand how to solve a problem and purpose of using different function calls. This analysis of existing literature presents the need to support novice programmers with additional information to effectively use Q&A websites. Also, we observed that tools such as *ExampleCheck* [8] have been developed to augment

Stack Overflow to support developers by reporting incorrect usage of APIs. However, to the best of our knowledge, existing work did not focus on providing information about API calls on Stack Overflow for novice programmers, motivating the need for our work. In this paper, we propose to augment Stack Overflow with *definition, usage* and *examples* for certain inbuilt API calls of Java programming language.

*StackDoc*<sup>1</sup> is a Stack Overflow prototype plug-in to help beginners for simple and rapid learning. In this preliminary version of our plug-in, a pop-up consisting of Java API calls information is shown on the webpage to the users, instead of searching on other sources for API definitions. We also conducted a survey to evaluate our plug-in and received satisfactory results with feedback of about 85% users recommending the plug-in to their peers. Fig 1 shows a high-level overview of *StackDoc*. We extract API definitions from standard Java documentation, OpenJDK 10<sup>2</sup> and to demonstrate our idea, we use JExamples<sup>3</sup> to extract API examples. This extracted information is displayed to the user on Stack Overflow. The essence of the plug-in is to extract examples and information about a certain API call from multiple sources and present this information to novice programmers.

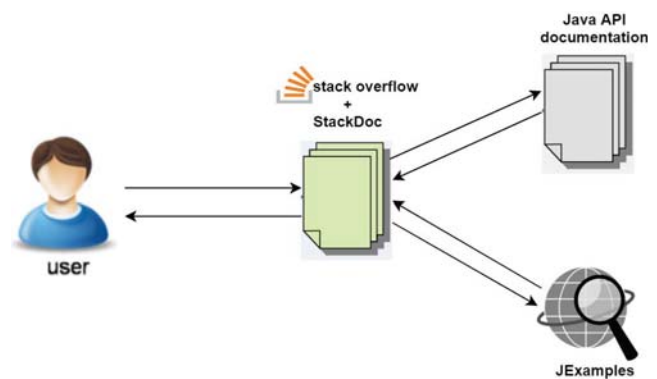


Fig. 1. Overview of *StackDoc*

The remainder of this paper is structured as follows. Section II discusses the related work followed by Section III, which focuses on design methodology and development of *StackDoc*.

<sup>1</sup><https://github.com/AkhilaSriManasa/StackDoc>

<sup>2</sup><http://cr.openjdk.java.net/~iris/se/10/latestSpec/api/>

<sup>3</sup><http://www.jexamples.com>

We present the evaluation and user survey results in Section IV and Section V. Finally, we discuss the limitations in Section VI and end the paper with conclusions in Section VII and future directions in Section VIII.

## II. RELATED WORK

The number of code snippets on websites such as Stack Overflow are increasing dramatically and most of these snippets use many API calls [9]. There are various development, debugging and learning tools to support novice programmers (as shown in Table I). *Scratch* supports beginners to learn programming by designing, creating and remixing code blocks [10]. *Alice* was developed to support novice programmers learn basic concepts of programming through 3D visualization [11]. Treude et al. have augmented API documentation with insights from Stack Overflow [12]. Tools such as *Prompter* have been developed to support users with discussions on Stack Overflow, based on the context of code in an Eclipse IDE [13]. *Exemplar* provides applications relevant to queried APIs entered by users in the tool [14]. The relevant applications are retrieved using information retrieval and program analysis techniques [14]. In *Examplore*, code snippets have been bound with the examples of API usage and definitions of API calls [15]. *Examplore* produces an interactive visualization to view general usage patterns of an API call in a code snippet [15]. *Blueprint* integrates source code examples into Adobe Flex Builder, a development environment, resulting in faster example code search [16]. Zhang et al. augmented Stack Overflow with API misuse warnings through a Google Chrome plug-in, *ExampleCheck* [8]. It displays a pop-up with API misuse alert and a suggestion to fix these misuses by providing curated examples that use the specific API call correctly [8].

TABLE I  
RELATED WORK

S.No.	Domains	References
1	Development	Examplore [15] Exemplar [14], Prompter [13] Blueprint [16]
2	Debugging	[17], [18], ExampleCheck [8]
3	Learning	Scratch [10], Alice [11]

Although *ExampleCheck* provides examples for correct usage of APIs, it does not provide definitions and examples of any other APIs used in the code snippets. *StackDoc* displays definitions and examples of all identified in-built API calls. It can thus serve as a unified solution, as it reduces the effort of searching for API usage and definitions explicitly. *Exemplar* provides a search interface to retrieve examples for queried API calls, whereas, *StackDoc* integrates API definitions and API examples inline with the code snippets in java.

## III. DESIGN AND DEVELOPMENT OF *StackDoc*

We have developed *StackDoc* as a browser extension and tested on Google Chrome and Mozilla Firefox browsers. *Stack-*

*Doc* facilitates novice programmers by parsing the Stack Overflow webpage and retrieving documentation for identified API calls from online sources such as OpenJDK and JExamples. These API calls are highlighted, for which, documentation and relevant example usages are displayed in a popup to the user.

This could help users learn about many new API calls that the user might be unaware of. This design methodology of *StackDoc* makes it distinct from existing approaches which display definitions and examples after an explicit query from the user. Whereas, we rely on standard Java documentation, OpenJDK, to retrieve API definitions and JExamples.

Fig2 shows the development process of *StackDoc* consisting of 5 steps.

In the first Step, we create a regular expression to identify function calls in Java. The regular expression we used to identify API calls is given below.

```
(([a-zA-Z$_]+[a-zA-Z0-9$_]*)\.)?  
[a-zA-Z$_]+[a-zA-Z0-9$_]*\ (
```

The defined regular expression (regex) accounts for the fact that API calls can be made using a class name directly or using an instantiated object. If the function is called directly without specifying any variable or class name, then the regex will still identify the function call. The defined regex identifies API calls of the form <variable>.<function\_call>, <class\_name>.<function\_call> and <function\_call>.

In Step 2, *StackDoc* generates a list of keywords using the regex defined previously. These keywords are matched with the existing corpus of class names and API calls of Java 10 specification. *StackDoc* then stores all the identified API calls into an array for further processing.

In the third Step, for every identified API call, we search its documentation in OpenJDK. Initially, we search the combination of <variable or class name>.<function\_call> directly in the documentation. If a matching combination is found, then its documentation comprising of the arguments and definition is fetched. If any matching combination is not found, we search OpenJDK documentation only for the <function\_call>. Once a matching API call is found, it is highlighted and made interactive so that the user will be able to click on it.

As a part of Step 4, when any of these highlighted API calls are clicked, corresponding usage examples for the particular API call are fetched from JExamples website.

Finally, in Step 5, the documentation of API calls and their usage examples are concatenated and displayed as an overlay. The overlay popup is hidden by default and becomes visible only when the user clicks on any highlighted API call. For every API usage example, a link to JExamples is provided from where the complete example can be viewed.

## IV. EVALUATION

We conducted a study to evaluate the usefulness of *StackDoc*. We aimed at assessing the extent to which *StackDoc* could be helpful to novice programmers in understanding and implementing API calls and data types in Java. Hence, we considered 20 university students novice to Java. The study

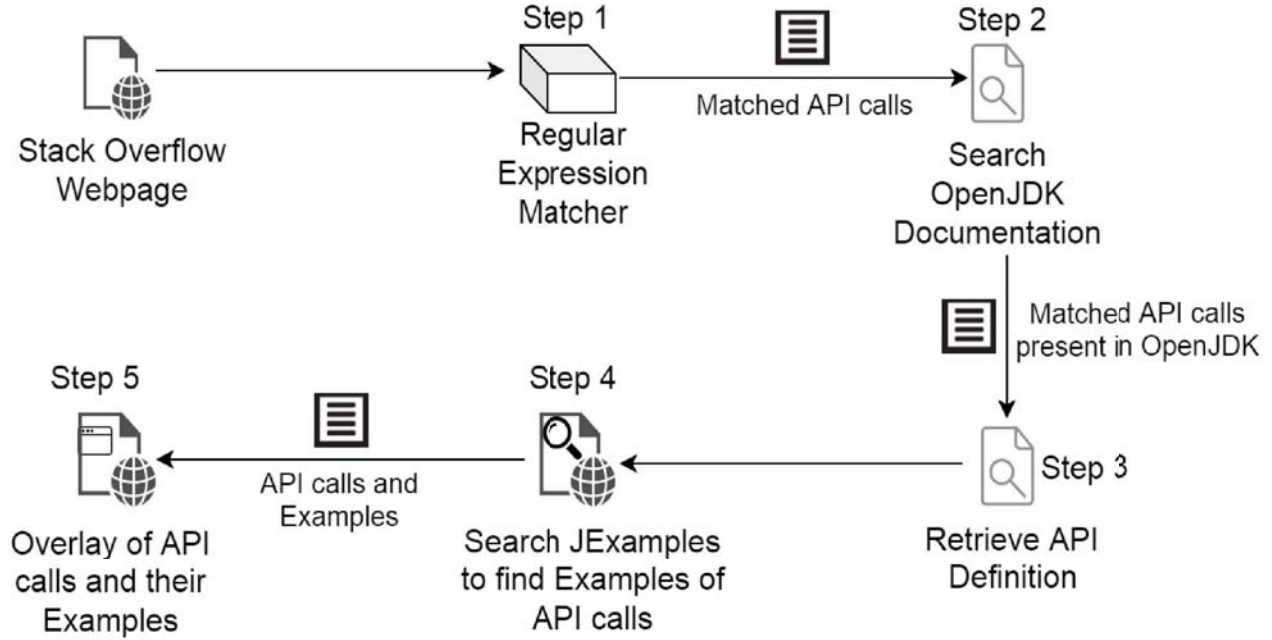


Fig. 2. Design Methodology of StackDoc

was performed with the help of a questionnaire based on Likert scale [19], on personal laptops of students.

#### A. Procedure

All the participants were requested to add *StackDoc* extension to browsers on their laptops. They were all provided with a slide-show depicting the working of *StackDoc*, that served as a basic tutorial. These 20 participants were then asked to search for questions related to Java programming language and go through at least 20 questions and answers that they have retrieved as a result of their search. They were requested to use *StackDoc* to clarify their doubts about usage and definitions of API calls that might arise in the process of their observation.

After completion of the above exercise, participants answered a questionnaire using a 5-point Likert scale. Questionnaire given to the participants is as shown in Table II.

#### B. User Scenario

Suppose *Veda* is a novice programmer working on Java and wishes to know why sorted arrays are processed faster than unsorted arrays, for which she queries on Stack Overflow.

She is then displayed with a list of posts related to this query as shown in [A] of Fig 3. She randomly selects one of the displayed posts. Once a post is selected, Java API calls present in code snippets of the post are highlighted ([B] of Fig 3). Among the answers displayed, *Veda* encounters API calls such as *System.println()*, *System.nanoTime()*, *Arrays.sort()*, *Random.nextInt()* and wishes to know their definitions and usage. *Veda* clicks on *System.println()* in the code snippet. She is then displayed with a pop-up containing definition and examples of *System.println()* as shown in [C] of Fig 3.

If *Veda* navigates to another example containing *log* and clicks on the API call, she will be able to view description and usage of *log(Level level, Supplier <String>msgSupplier)* (as represented in part [D] of Fig 3).

### V. RESULTS

#### A. Questionnaire

As reported in Fig 4, *StackDoc* had a good user-friendly interface (84% in Q1). In Q2, participants have agreed that *StackDoc* retrieved moderately sufficient number of examples to understand the API usage (83%, about 10 participants voted for *Agree* and 5 participants voted for *Strongly Agree* out of 20). The ratings in Q3 and Q4 indicate that *StackDoc* has helped participants learn about partly unaware API calls, reducing the search time (73% in Q3 and 79% in Q4). Participants have also suggested to improvise *StackDoc* to support other languages and to provide descriptions to a wider range of API calls. In Q5, most of the participants have agreed that they would recommend *StackDoc* to their peers (83%).

### VI. LIMITATIONS

We presented a prototype of *StackDoc*, that augments Stack Overflow by helping programmers to learn about Java API calls. However, our plug-in could be improved in multiple ways in future versions. Currently, *StackDoc* shows API definitions only for Java, restricting its application to one programming language. Also, we were not able to find API definitions for few APIs as they do not exist in OpenJDK 10 or might have been deprecated. Similarly, the limited availability of examples on JExamples website limits *StackDoc* to retrieve examples for few of the desired APIs.

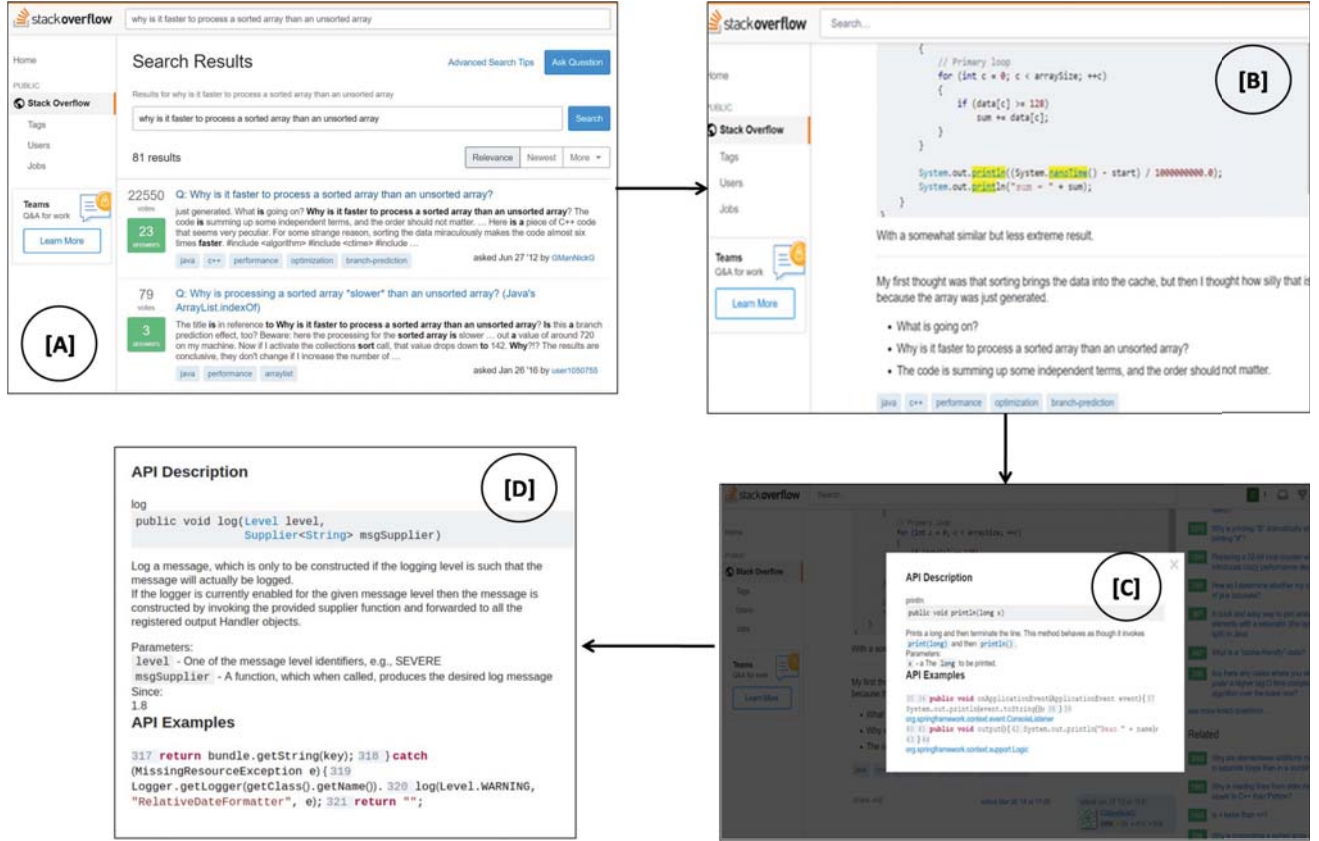


Fig. 3. Example user scenario of *StackDoc* by user Veda; A: Search question on Stack Overflow; B: API calls highlighted by *StackDoc*; C: Description of API call with Example as given by *StackDoc*; D: API Description for another example *StackDoc*

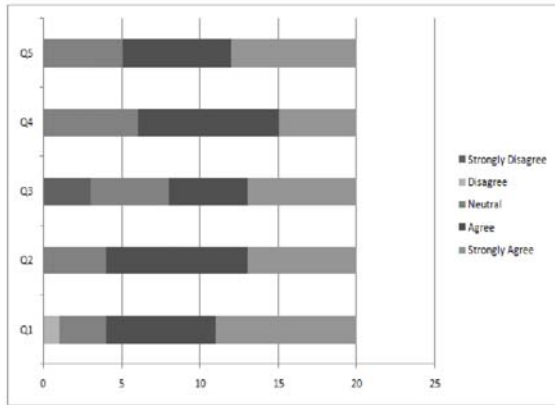


Fig. 4. Results of Questionnaire

In its current version, there is a delay between the user asking for API definitions and *StackDoc* retrieving it from the online sources, OpenJDK 10 and JExamples. The reason of

this delay being that *StackDoc* fetches required details only after the user clicks on highlighted API call.

## VII. CONCLUSIONS

We emphasized the need to support novice programmers when they browse Q&A websites like Stack Overflow. Hence, we have introduced *StackDoc*, a browser plug-in that augments Stack Overflow. Our tool is an initial step to support novice programmers with better mechanisms. We have extracted documentation of java API calls from OpenJDK 10 and examples of these API calls from JExamples website. These API calls are highlighted on Stack Overflow page and the extracted information is displayed to the user as a pop-up when clicked on the highlighted API calls. Our initial user study indicated that *StackDoc* has helped users in finding definitions, but with the need to have more examples. We plan to extend *StackDoc* to support a larger number of programming languages and multiple data sources. We plan to do an extended study with 50 users and incorporate changes to improve our plug-in.

## VIII. FUTURE RESEARCH DIRECTIONS

Beyond *StackDoc*, our core idea is to support software engineers by integrating documentation with software development platforms. Here are a few potential future directions:



TABLE II  
QUESTIONS IN SURVEY USING A 5-POINT LIKERT SCALE.

<b>Q1:</b> How easy was it to use the plug-in interface?(1=very easy, 5=very difficult)
<b>Q2:</b> <i>StackDoc</i> has retrieved enough number of examples to understand a particular API usage. (1=strongly agree, 5=strongly disagree)
<b>Q3:</b> <i>StackDoc</i> has helped me in learning about API calls that I wasn't aware prior to this exercise. (1=strongly agree, 5=strongly disagree)
<b>Q4:</b> <i>StackDoc</i> has made my learning quicker by reducing my search time. (1=strongly agree, 5=strongly disagree)
<b>Q5:</b> I will recommend <i>StackDoc</i> to my peers. (1=strongly agree, 5=strongly disagree)

#### A. Code hosting platforms

For example, *GitHub* could be augmented with appropriate documentation such as API information to help developers, especially novice developers to understand source code repositories.

#### B. Documentation for System Administrators

Documentation is critical in system administration tasks, but is often missed as the focus is only on executing the tasks, making it difficult for novice system administrators. We see that developing a plugin to support novice system administrators based on *StackDoc* is a valuable future direction.

#### C. Algorithm Documentation

Understanding code snippet might be difficult than understanding an algorithm for a novice programmer. If an algorithm is available or if an abstraction can be created, code snippets could be integrated with the algorithm.

#### D. Deployment Scenarios

While deploying an application, in case of failures, it is difficult to identify code location of failed modules and make necessary changes. Each module could be integrated with deployment document containing code repository information.

### IX. ACKNOWLEDGEMENT

We thank all the participants for their valuable time and feedback that helped us in evaluating *StackDoc*.

### REFERENCES

- [1] Yang, Di and Martins, Pedro and Saini, Vaibhav and Lopes, Cristina, "Stack overflow in github: any snippets there?" in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. IEEE, 2017, pp. 280–290.
- [2] R. Garcia, K. Falkner, and R. Vivian, "Systematic literature review: Self-Regulated Learning strategies using e-learning tools for Computer Science," *Computers & Education*, vol. 123, pp. 150 – 163, 2018.
- [3] Ko, Andrew J and Myers, Brad A and Aung, Htet Htet, "Six learning barriers in end-user programming systems," in *2004 IEEE Symposium on Visual Languages-Human Centric Computing*. IEEE, 2004, pp. 199–206.
- [4] J. E. Montandon and H. Borges and D. Felix and M. T. Valente, "Documenting APIs with examples: Lessons learned with the APIMiner platform," in *2013 20th Working Conference on Reverse Engineering (WCRE)*, Oct 2013, pp. 401–408.
- [5] Lahtinen, Essi and Ala-Mutka, Kirsti and Järvinen, Hannu-Matti, "A Study of the Difficulties of Novice Programmers," in *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ser. ITICSE '05. New York, NY, USA: ACM, 2005, pp. 14–18.
- [6] Bosse, Yorah and Gerosa, Marco Aurelio, "Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage," *ACM SIGSOFT Software Engineering Notes*, vol. 41, pp. 1–6, 01 2017.
- [7] Joorabchi, Arash and English, Michael and Mahdi, A.E., "Text mining stackoverflow: Towards an Insight into Challenges and Subject-Related Difficulties Faced by Computer Science Learners," *Journal of Enterprise Information Management*, vol. 29, pp. 255–275, 03 2016.
- [8] Zhang, Tianyi and Upadhyaya, Ganesha and Reinhardt, Anastasia and Rajan, Hridesh and Kim, Miryung, "Are code examples on an online Q&A forum reliable?: a study of API misuse on stack overflow," in *Proceedings of the 40th International Conference on Software Engineering*. ACM, 2018, pp. 886–896.
- [9] Subramanian, Siddharth and Holmes, Reid, "Making sense of online code snippets," in *Proceedings of the 10th Working Conference on Mining Software Repositories*. IEEE Press, 2013, pp. 85–88.
- [10] Resnick, Mitchel and Maloney, John and Monroy-Hernández, Andrés and Rusk, Natalie and Eastmond, Evelyn and Brennan, Karen and Millner, Amon and Rosenbaum, Eric and Silver, Jay and Silverman, Brian and others, "Scratch: programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60–67, 2009.
- [11] Cooper, Stephen and Dann, Wanda and Pausch, Randy, "Alice: a 3-D tool for introductory programming concepts," *Journal of Computing Sciences in Colleges*, vol. 15, no. 5, pp. 107–116, 2000.
- [12] Treude, Christoph and Robillard, Martin P., "Augmenting API Documentation with Insights from Stack Overflow," in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE '16. New York, NY, USA: ACM, 2016, pp. 392–403.
- [13] Ponzanelli, Luca and Bavota, Gabriele and Di Penta, Massimiliano and Oliveto, Rocco and Lanza, Michele, "Mining StackOverflow to turn the IDE into a self-confident programming prompter," in *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 102–111.
- [14] Grechanik, Mark and Fu, Chen and Xie, Qing and McMillan, Collin and Poshyvanyk, Denys and Cumby, Chad, "A search engine for finding highly relevant applications," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*. ACM, 2010, pp. 475–484.
- [15] E. L. Glassman, T. Zhang, B. Hartmann, and M. Kim, "Visualizing API usage examples at scale," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*, 2018, p. 580.
- [16] Brandt, Joel and Dontcheva, Mira and Weskamp, Marcos and Klemmer, Scott R, "Example-centric programming: integrating web search into the development environment," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 513–522.
- [17] Spinellis, Diomidis, "Modern debugging: the art of finding a needle in a haystack," *Communications of the ACM*, vol. 61, no. 11, pp. 124–134, 2018.
- [18] Torroja, Yago and López, Alejandro and Portilla, Jorge and Riesgo, Teresa, "A serial port based debugging tool to improve learning with arduino," in *Design of Circuits and Integrated Systems (DCIS), 2015 Conference on*. IEEE, 2015, pp. 1–4.
- [19] Likert, Rensis, "A technique for the measurement of attitudes," *Archives of psychology*, 1932.