www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Preliminary Insights for Automating the Software Development Process in Startups

Ms.B.Swapna[1]., k.keerthana[2]

1 Assistant Professor, Department of ECE, Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India
2, B.Tech ECE (19RG1A0486),
Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

ABSTRACT

Software Engineering (SE) activities, such as requirements categorization, software refactoring, defect prediction, and many more, have been progressively bolstered by AI and ML tools and methodologies. The urge to quickly offer a workable solution is especially difficult for software Startups, who build novel and scalable software solutions. Our goal is to learn more about the artificial intelligence (AI) and machine learning (ML) methods used by SE professionals and business owners to improve their Software Development Procedures (SDP) and make them more accessible to software start-ups. We plan to determine this data by using a web-based survey that will primarily be distributed in Brazil and Finland. This exploratory research offers suggestions for bettering Startups' SDP.

1 INTRODUCTION

As software becomes more and more ingrained in our daily lives, ensuring its high quality becomes an increasingly pressing concern for software engineers. To address these formidable problems, however, researchers are increasingly looking to the interface of SE and AI. To overcome these difficulties, AI provides tools for intelligent software engineering, such as ML, that can reason, problem solve, plan, learn, and more. Practitioners and business owners in software startups may gain a lot from learning about and adopting the best practices for SDP automation. Since unpredictability, lack of resources, fast change, and an immature team are hallmarks of the engineering setting of Startups [6]. The software industry as a whole gains greatly from the elimination of SE issues. For this reason, many different algorithms and methodologies have been devised and refined throughout time to enable the practical use of prediction systems [4]. We performed this Survey to learn more about the current state of AI and ML integration within software development prototyping. A survey's primary goal is to gather data from a statistically representative subset of a larger population [5]. Despite the early nature of the data, we were able to identify potential avenues for future study and research that might help advance the field.

2 THE EXPLORATORY STUDY

In order to better assist and automate the development process in software start-ups, this research aims to enhance SDP, which is made up of AI/ML components. We established the following research question based on our aims:
RQ. When creating new software, how can developers make advantage of AI and ML?
To investigate whether or not ML might be successfully introduced to the market, we designed a prototype survey. Our goal in doing this first research was to identify commonalities across the various types of participants, organizations, and development teams that have used ML-based SDPs for software startups. Seven sets totaling 26 questions make up the Survey, with 17 of them having direct relevance to the topics covered here. The next 8 questions all seek demographic data from respondents, while the ninth solicits thoughts on the poll itself. Our analyses were organized according to the first ten Knowledge Areas (KA) of SWEBOK [3], which cover topics like "Software Requirements," "Software Design," "Software Construction," "Software Testing," "Software Maintenance," "Software Configuration Management," "Software Engineering Management,"

"Software Engineering Process," "Models and Methods of Software Engineering," and "Software Quality."The whole instrument is included in our replication package1, and we based this version on the findings from the research by Borges et al.

3 PRELIMINARY RESULTS AND DISCUSSION

Here, we provide a synopsis of our analysis and discussion of the findings, organized according to the sections in which the survey questions appeared.

SWEBOK: Only a small number of research have made use of AI/ML to investigate KA1, KA2, and KA3.

Such KAs rely more heavily on human perceptions and are hence more challenging to completely automate. While studies sometimes provide semi-automatic methods for assisting throughout these stages, human involvement is ultimately required. However, KA4, KA5, KA7, KA8, K9, and KA10 are often referenced. Literature studies focus mostly on KA4 and KA10 because they represent two distinct but related paths to full automation. In addition, KA9 is intrinsically linked to automation since it involves the development and use of models, methodologies, tools, algorithms, and other strategies for SDP. Several research have proposed the development of such methods to aid in a certain developmental stage.

Tools and Methods: We started by inquiring about the respondents' prior experience with various research instruments [1]. The most widely used software packages are WEKA, MATLAB, and Scikit-learn, followed by libsvm, Mulan, RNNLM Toolkit, Tensor Flow, the R programming language, and Python-based software packages. One of the replies indicated that all KAs make regular use of AI/ML for SE-related duties. While some respondents stated they never used it for KA1, KA2, and KA3, others said they only used it a handful of times for KA4 and KA6, while yet others said they used it on a daily basis for KA5, KA7, KA8, KA9, and KA10.

Most of the algorithms (NB, RF, DT, J48, KNN, SVM, and SVR) that come up in discussions on AI/ML belong to Supervised Learning, either for classification or regression. In addition to Learning Sets (Ba, AB, Bo), Neural Networks (ANN, MLP, CNN, etc.) may be used. In most discussions on Unsupervised Learning, techniques like K-means clustering are brought up. Merchandise, Function, or Offering: Based on the responses analyzed, it appears that most respondents' efforts were focused on Knowledge Area 5 (KA5), as the details provided about the development of their products and services are useful for developers during software upkeep procedures, such as in the case of building and maintaining research tools or collecting and analyzing documents based on scores.

Priority: Getting a sense of which KAs are most interesting from a potential automation standpoint requires first learning which KAs respondents would want to automate, and in what sequence. The analyzed findings provide light on why certain KAs are more popular than others. Focusing on a KA may not always be desirable due to circumstances like as the project's goal, the timeline for its production, and other aspects of project management. As a result, the environment into which the effort at automation is introduced may render it impossible to succeed. Alternatively, reluctance to learning new automation techniques may result from a lack of background information on a particular KA. Difficulties Several difficulties in using AI/ML to aid SE may be seen in the literature [2]. This is why the requirement to clean and analyze this data was highlighted as the primary barrier for automation in KA5. We found that this view and the literature on the topic of this difficulty in automation coincide, which presents a new stage of opportunity for researchers keen to learn more about automating this process.

Even while KA5 is more amenable to automation in comparison to earlier assessments, there are still obstacles to be overcome. The stated KA's (KA3, KA4, and KA5) are connected to the difficulties in solving SE assignments since they are often worked together. One person specifically noted the difficulties of integrating them, highlighting the fact that they share common ground. KA1 and KA8 topped the list of most-mentioned KAs, indicating that solutions to problems in these areas are more often discovered by respondents. As a last question, we inquired about their ideal AI/ML solution for automating the SDP. Four respondents mentioned a KA3-specific tool in the study. According to our examination of the responses, the majority of the proposed resources are geared toward the software development process, opening up yet another fascinating avenue for investigating automation. There are more attempts to address issues with SE tasks in the literature for KA4 than for any other phase.

It's encouraging to see that there is still a lot of room for exploration in this area, despite the fact that responders still need tools to enable it. This KA has as its primary emphasis the organization and management of the ecosystem in which software is created. The architecture of the software is crucial since it establishes the foundation of the program. We emphasize that this KA is an important contribution to our study.

4 CONCLUSION REMARKS

We suggested a first study of how business owners and professionals are using AI and ML in their product and service creation efforts. As a result of our investigation, we have a better idea of how to go with automating some of the KAs.We discovered that KA7 and KA8 are keen on automation since the completion of their jobs improves software managers' management procedures. We established that KA5 is very interested in finding answers since doing so can hasten software upkeep, but that poor data quality remains a barrier to progress.
Since the KA2 is the software's foundational framework and any flaws in its creation might have far-reaching effects on the project, it was deemed the top automation priority. There is still need for research into solutions that assist processes in this area, despite the fact that many studies center on activities related to KA4. This research is one of many studies conducted at the interface of SE and AI with the end goal of helping new software companies get off the ground. As a minimum, we expect that the early findings of this research will help software start-ups incorporate AI/ML into all stages of their development processes.

REFERENCES

[1] Olimar Borges, Julia E Couto, et al. 2020. How machine learning has been applied in software engineering?. In ICEIS. 306–313.
[2] Olimar Borges, Julia E Couto, et al. 2021. Challenges in using Machine Learning to Support Software Engineering. In ICEIS. 224–231.
[3] Pierre Bourque, Richard E Fairley, et al. 2014. (SWEBOK (R)): Version 3.0. IEEE, Washington, US.
[4] Anh Viet Phan, Phuong Ngoc E Chau, et al. 2018. Automatically classifying source code using tree-based approaches. DKE 114 (2018), 12–25.
[5] Singer, Janice and Sim, Susan E and Lethbridge, Timothy C. 2008. Software Engineering Data Collection for Field Studies. Chapter 1, 9–34.
[6] Stanley M Sutton. 2000. The Role of Process in Software Start-Up. IEEE Software 17 (2000), 33–39.
[7] Yuwei Zhang, Dahai E Jin, et al. 2020. Automated defect identification via path analysis-based features with transfer learning. JSS 166 (2020), 110585.