

A NOVEL METHOD FOR SOFTWARE BUG REPORT ASSIGNMENT

LUKASZ CHMIELOWSKI^{1*,2}, PAVLO KONSTANTYNOV¹,
RYSZARD LUCZAK¹, MICHAL KUCHARZAK^{1,2}, ROBERT BURDUK²

•

^{1*}Nokia Solutions and Networks sp. z o.o., Poland

lukasz.chmielowski@nokia.com

pavlo.konstantynov.ext@nokia.com

ryszard.luczak@nokia.com

michal.kucharzak@nokia.com

² Wroclaw University of Science and Technology, Poland

robert.burduk@pwr.edu.pl

Abstract

During the development of software and electronic devices, it is inevitable to make mistakes. In large, developed companies, assigning a request to the right development team or even a department is not an easy task. Often, the creation of software bug reports and assignment to groups is also formalized by appropriate processes. The paper presents a novel method of software bug report assignment to a group of developers or analysts. A specific usage of organizational structure at the company is a key component of the proposed approach. There are presented results from real use application including both machine learning predictions and human decisions. Human predictions are not independent, the issues are raised as to why comparing the results of machine learning models with those of humans may be inappropriate and what factors influence human decisions. The work also covers conclusive research about potential benefits of the application of automated assignment of bug reports.

Keywords: Software bug assignment, Software bug triaging, Software bug report, Software bug, Text analysis

1. INTRODUCTION

Discussed problem concerns about assigning a software bug report to correct group automatically based on given data like description, system information in raw format or already processed by analyzing tools. Approaches similar to these presented in the paper may be applied to situations which work with other software development cases like related to feature requests, supporting questions or similar issues which should be handled during software development or maintenance. The approach might also be applied to any other different task related to machine learning tasks like classification or labeling in a similar context. It is expected that if a software bug occurs at unit testing level it should be handled by one of developers responsible for development of this unit. More challenging part is when a software bug occurs at the later stage of development or even in real customer use. For large and complex systems even pointing out department can be a complicated task [6]. Additionally, there is an assumption that the company is divided into at least two organization levels, like departments and divisions, as shown in Figure 1. Please note that the names "department" and "division" and relations between them are shown in Figure 1

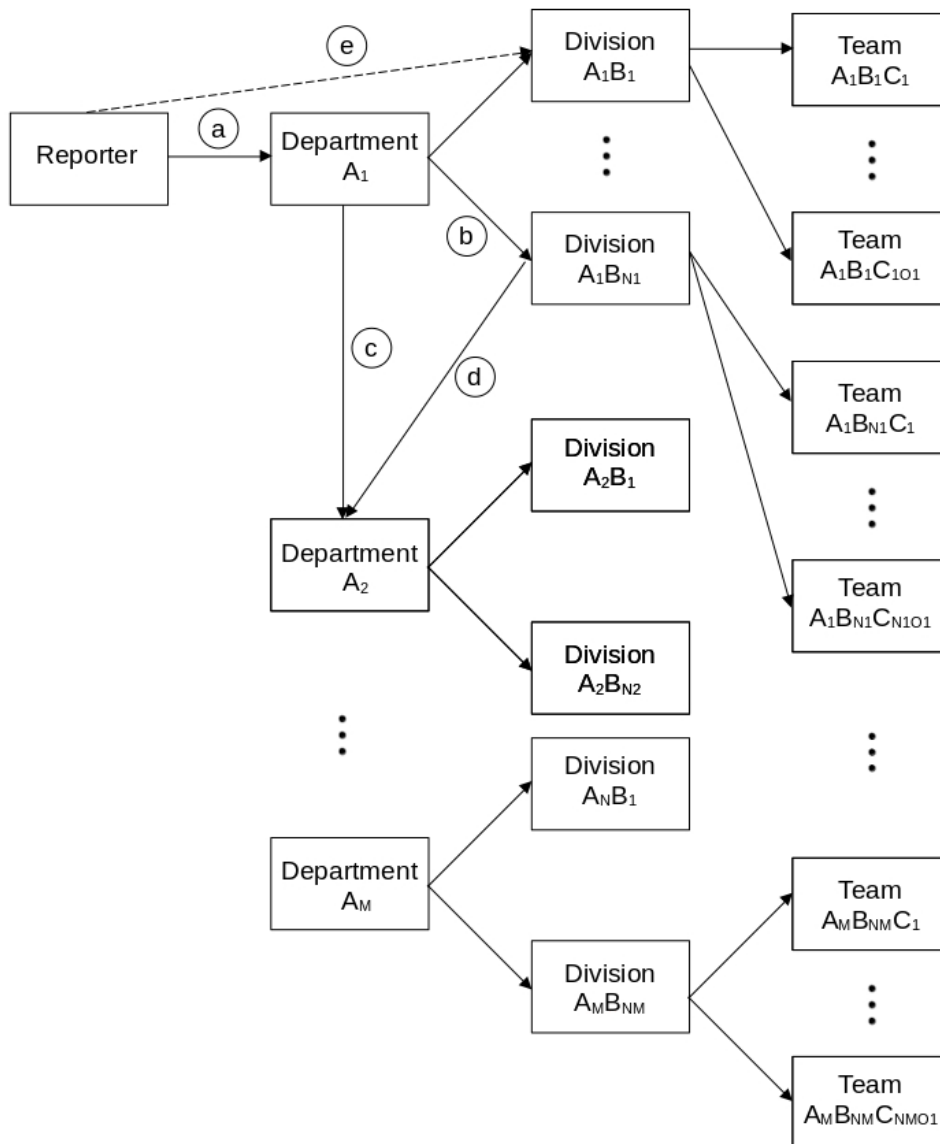


Figure 1: Flow chart of process of transferring reports of bugs inside company, 3 layers shown.

serve only to better illustrate the example. In real use cases testers or customer support engineers decide which department is most suitable for resolving issue (a), while reporting a software defect or anomaly. Next, a report is being assigned to one of the divisions inside the current department (b) or transferred to another one (c). The problem of assigning a software bug report to correct group in that context may be interpreted as:

- assigning to department (a) or (c),
- assigning to division in context of department (b),
- assigning to division directly (e).

2. RELATED WORKS

There is a plethora of ways to classify issues, i.e., classifying severity [7] or assigning the issue to a group which should handle particular case. As there may be numerous bug reports, not all of

them are handled simultaneously. Among others based on classification of severity decisions are made as to whether the bug will be fixed now, later or never. In [8] an approach to assign issue to specified components is presented. In above-mentioned work authors predict if reassignment of created bug report will occur. For that purpose they are using data which come from major projects Eclipse and Mozilla. In [2], bug report assignment is done directly to developers. In the scope was to build time oriented expert model which assigns more priority to developer who had worked on the similar bugs in the past. There are created activity profiles of people who deliver corrections with usage of factor for normalizing which uses the time of last usage of term by developer. [4] addresses bug report assignment to departments. It uses a specific time dependencies for creating train and test sets. Also one of the scope of that work was to investigate the impact of different way of preprocessing and vectorization on bug assignment accuracy. [9] considers bug report assignments to development teams. Moreover, the approach presented in [9], uses only selected cases to automate bug triaging. Selection is made based on confidence of the prediction. A threshold (cutoff) for the confidence is used as there exists a trade-off between accuracy and the number of predictions. [5] presents dual-output deep neural network which simultaneously predicts developer and team. Authors of that work also indicate the fact that this approach is robust against organizational changes as relations between teams and developers may change. In different kind of applications like disease detection and classification, a hierarchical concept of combination of machine learning models is used in [1]. There are two layers used. The purpose of the first of them is disease detection and the second one is its classification. The results also suggest that the hierarchical approach can outperform the flat one, especially in case of small amount of data. A general concept of hierarchical classification algorithms is described in [3]. It presents among others different type of structures for hierarchical problems.

Although in [9] was introduced the possibility of transferring bug reports to selected organization parts based on the thresholds, there is lack of publication which applies specific context and additional possibilities which can be gained due to known hierarchical structure, like for instance combination of models related to different levels especially with usage of tuning of thresholds.

3. RESEARCH QUESTIONS

- What are results of human predictions on department and inside department levels?
- What is the relation between results of machine learning predictions versus human predictions?
- What are the factors impacting human predictions?
- What are minimal requirements for solution applicability in software development company?
- What are the scenarios of deployment of application?
- What are main advantages and disadvantages of human versus computerized approaches?

4. PROPOSED SOLUTION

The novelty is that the cases incoming into department are being transferred into divisions (operation (b) in Figure 1) only in specific conditions. In that case a novel combination of machine learning models may be used, where one of the models predicts which division the issue should be addressed to, whereas the second one predicts whether the current department is the proper one. As a result, it is a transfer to proper division only in case if both of used models exceeded respective threshold. By the threshold we understand the cutoff confidence score based on the output of machine learning model. The novel features can be expressed as follows:

- assigning only selected cases of all cases incoming into department (or created inside) to specific divisions based on confidence level of prediction or state of issue,
- creating a model or other decision system based on at least two classification models, where at least one of them predicts division and at least one of them predicts department.

That approach is general and it is not limited to one way of creating a model. In a specific implementation in the company, predictions come from models prepared in a way similar to the approach presented in [4]. Fields used from bug report are title, description, product and release. The preprocessing phase uses methods for cleaning the text like removing chosen special characters, changing to lower letters, removing stopwords and part of content related to company template. The training set is built from data related to relevant cases from last 365 days up to date of creation of model. For each case the result of prediction was collected at the time when formally correct bug report appeared for the first time at A_1 . Different models are used for predictions at different levels of organization. Department is being predicted with the use of logistic regression classifier; division with use of support vector classifier with linear kernel. This production setup is being updated daily to get the newest available data for training as fast as possible.

5. RESULTS

The doctoral student conducted research on the possibility of automatic assignment of bug reports in selected cases. The results of the studies show what the effect on historical data would be if bug reports were sent to inside department from interface group A_1 . The predictions come from the cases which passed formal check of correctness of report filling at the time of use of machine learning model to make prediction and contain valid log content. Table 1 shows the number of cases above a certain threshold of a model which would be sent predicting department and respective precision. Similar research was conducted for model which predicts divisions inside the department. Table 2 indicates the number of cases which would be transferred from group A_1 to A_1B_1 and its precision. The results with combination of those two models are placed in Table 3. Based on above data, the decision about implementation pilot solution with thresholds 0.6 for department, 0.3 for division was made. Within that solution problem reports which meet the above-mentioned requirements were transferred. For those cases which did not meet these conditions were only placed information about suggested transfer possibilities. Selected results are presented in Tables 4 to 6 and Figures 2 to 4. Presented data do not show cases which were created at early stages of software development and discovered in later phases or even in customer use. The following notations are used in Table 6:

- Human only - percentage of cases where human prediction was correct, but ML model prediction was incorrect;
- ML only - percentage of cases where ML model prediction was correct, but human model prediction was incorrect;
- ML & Human - percentage of cases where both ML and human model predictions were correct;
- Both incorrect - percentage of cases where neither ML nor human model predictions were correct.

Additionally, we can see the benefits like that for cases in date ranges from November 2021 to January 2022 where the decision about transfers was made 79% of them were resolved¹ (or fix was not required²) inside department A_1 , but for cases where only decision about suggestion was made only 66% were resolved (or fix was not required) inside.

¹Resolved - resolved; not including internal department cases; ended inside department

²Fix not required - fix not required; not including internal department cases; ended inside department including inflow group A_1

Table 1: *General flow of issues in organization.*

Threshold set	Cases predicted as A_1	Precision of A_1 in the context of cases above threshold
0.2	266	58
0.3	244	61
0.4	183	64
0.5	130	68
0.6	71	67
0.7	48	71

Table 2: *General flow of issues in organization.*

Threshold set	Cases predicted as $A_1 B_1$	Percent of cases $A_1 B_1$ among accepted ended on A_1	Percent of cases $A_1 B_1$ among accepted
0.2	206	39	20
0.3	148	40	23
0.4	95	42	23
0.5	53	38	23
0.6	28	39	25

6. DISCUSSION ON REQUIREMENTS FOR APPLICATION OF SOLUTION INSIDE COMPANY

6.1. Minimal requirements

Although many people at glance think that such solutions have opportunities to be introduced only in case the predictions are better than human ones, this is not so simple as it is thought. In the case when machine learning predictions are better than it is rather obvious that is worth to make it application. Otherwise when it is worse, in some cases it may be also worth developing and applying such solutions. One of the reasons is that it may work as decision supporting system which does not make a binding decision, but only delivers suggestions which may be helpful for cases when a reporter has no idea how to address the problem, and sometimes may ignore suggestions when is sure where to address that or knows that the suggested target is wrong. Sometimes an issue with group overloading may occur, like for instance they currently handle too many bug fixes simultaneously, or have to deliver already committed new important features to product. Then, from the businesses perspective it may be reasonable to redirect cases to groups where it is less likely that the corrections will be delivered, but they may deliver detailed analysis or reject bug report as not valid. That effect may be gained due to tuning of mentioned in this work thresholds. Even if this change could lead to accuracy reduction, it may help to achieve business goals.

Table 3: General flow of issues in organization.

Threshold set for department A_1	Threshold set for A_1B_1	Cases predicted as A_1B_1	Precision of A_1B_1 in the context of cases above threshold and ended in A_1	Precision of A_1B_1 in the context of cases above threshold
0.3	0.3	88	36	25
0.3	0.4	60	38	26
0.3	0.5	40	38	25
0.3	0.6	26	41	27
0.4	0.3	65	34	25
0.4	0.4	44	34	25
0.4	0.5	31	34	26
0.4	0.6	21	33	24
0.5	0.3	49	40	29
0.5	0.4	34	40	29
0.5	0.5	21	43	33
0.5	0.6	13	44	30
0.6	0.3	29	57	41
0.6	0.4	20	64	45
0.6	0.5	11	75	55
0.6	0.6	7	80	57

Table 4: Chosen results transferred cases based on ML model decision.

Date range	type of resolution	Accuracy
November and December	Resolved	38 %
November and December	Fix not required	55 %
December	Resolved	50 %
December	Fix not required	80 %

6.2. Human factors

There are usually many validation aspects when comparing machine learning and human predictions in software bug report assignment process. Some of them are presented in the following paragraph. One of the most important ones is that the reporters may use an already introduced decision supporting system. The different aspect is that in cases when multiple groups delivered correction people can choose which one will be the final main one and may want to boost human or machine learning result if they wish so. What is more, reporters sometimes ask before creating reports where reports should be sent before creating formally one. At that step, many developers might be involved or even the solution might be known before the actual report is officially processed. Sometimes developer teams ask for verification of some functionality and create a report directly against them. In those last two cases the final group is known even before creating bug report. What is more, not always the best accuracy is the aim of introduction of such solutions. Last, but not least, recently detailed instructions on how to address the most common types of bug reports and responsibilities of divisions inside department were made to improve human decision making.

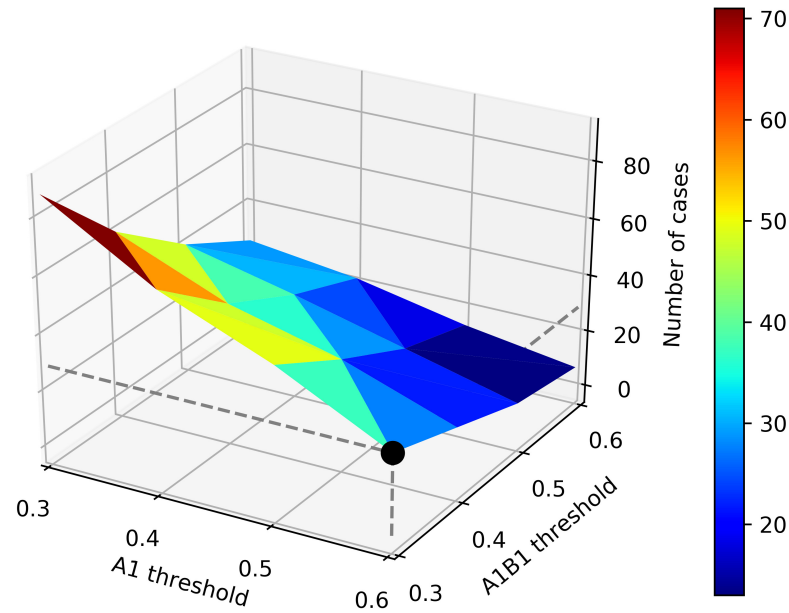


Figure 2: Number of bug reports meet threshold conditions in function of given thresholds.

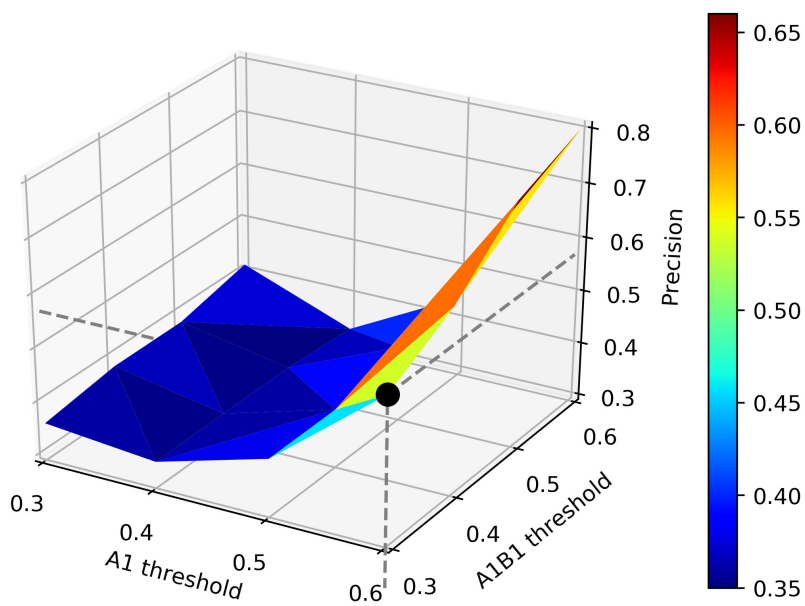


Figure 3: Precision of predictions of bug reports meet threshold conditions in function of given thresholds excluding cases which ended outside of department A_1 .

Table 5: Chosen results of suggested cases based.

Date range	type of resolution	Human accuracy	ML model accuracy
November and December	Resolved	54 %	38 %
November and December	Fix not required	65 %	35 %
December	Resolved	56 %	38 %
December	Fix not required	64 %	32 %

Table 6: Distribution of specific types of results of suggested cases.

Date range	type of resolution	Human only	ML only	ML & Human	Both incorrect
Nov and Dec	Resolved	31 %	15 %	23 %	32 %
Nov and Dec	Fix not required	40 %	10 %	25 %	25 %
Dec	Resolved	34 %	16 %	22 %	28 %
Dec	Fix not required	40 %	8 %	24 %	28 %

6.3. Advantages and disadvantages of such solutions

The main disadvantage of such solutions, which are often pointed out during automatic transfers, is the lack of analysis that would provide information on why such a decision was made. The second issue that will help minimize these defects is the implemented solution, which conveniently displays to developers' key information about the content of the base station configuration state logs at the time of collecting logs, provided of course the logs have collected in the correct way.

7. NEXT STEPS THAT HAVE BEEN TAKEN

Referring to the progress of implementation in industry, prepared an earlier pilot solution supporting the group A_1 , dealing with the handling of applications within the department, was gradually extended to handle more bug reports. The solution was to transfer selected bug reports from the group symbolically marked as A_1 to selected groups in the conditions specified by the machine learning model and if met the formal conditions for notification. This decision shall be taken automatically without the need for human verification. The model prepared was also used for transfers from department A_2 to the teams (A_1B_x) responsible for analysis in department A_1 as well as transfers to department A_2 . One of the models prepared is also used as one of the component models to solve suggesting to the reporter whether the report should be opened against department A_2 . In addition, it is also used as a component model for the automatic transfer solution applications between departments A_2 and A_3 . It was decided to remove a group responsible for initial investigation inside that department A_1 in June this year, thus fully abandoning one layer of analysis. In connection with these changes, the solution was adapted so that the submitted applications from department B were sent directly to groups A_1B_x . In addition, the system had to be adapted to indicate new groups after organizational changes, because on this layer the structure has also changed.

8. SUMMARY

The paper discusses problems related to methods of assignment of reports, feature requests, supporting questions or similar issues to group of employees, developers, organization unit, etc. The novelty introduced in this paper is related to the specific usage of the organizational structure in processes of handling (assigning) an issue. The paper shows possible scenarios of deployment of application supporting fault management with the use of solution based on

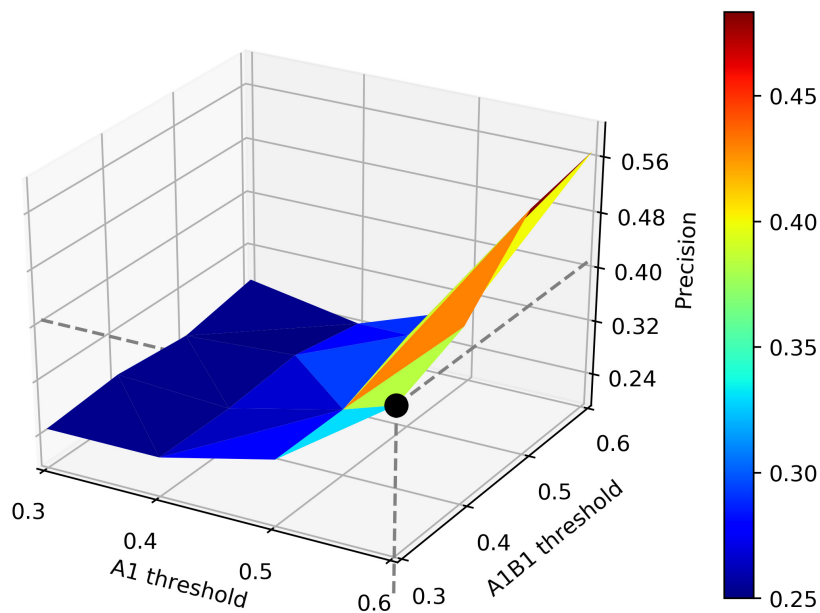


Figure 4: Precision of predictions of bug reports meet threshold conditions in function of given thresholds including cases which ended outside of department A_1 .

machine learning. The study demonstrates factors impacting human predictions, main advantages, and disadvantages of automated solution against human. Comparison of results between human and model predictions at both department and inside department levels are presented. Minimal requirements for the company in case of application of machine learning supporting system in the company are also defined.

ACKNOWLEDGEMENTS

This work has been carried out in cooperation between NOKIA and Wroclaw University of Science and Technology in context of a Ph.D. grant under the fourth edition of the "Implementation Doctorate Programme".

DECLARATION OF CONFLICTING INTERESTS

The Author(s) declare(s) that there is no conflict of interest.

REFERENCES

- [1] Guangzhou An, Masahiro Akiba, Kazuko Omodaka, Toru Nakazawa, and Hideo Yokota. "Hierarchical deep learning models using transfer learning for disease detection and classification based on small number of medical images". In: *Scientific Reports* 11.1 (Nov. 2021), p. 4250. ISSN: 2045-2322. DOI: 10.1038/s41598-021-83503-7. URL: <https://doi.org/10.1038/s41598-021-83503-7>.
- [2] Anjali, Devina Mohan, and Neetu Sardana. "Visheshagya: Time based expertise model for bug report assignment". In: *2016 Ninth International Conference on Contemporary Computing (IC3)*. 2016, pp. 1–6. DOI: 10.1109/IC3.2016.7880218.

- [3] Helyane Bronoski Borges, Carlos N. Silla, and Jlio Cesar Nievola. "An evaluation of global-model hierarchical classification algorithms for hierarchical classification problems with single path of labels". In: *Computers & Mathematics with Applications* 66.10 (2013). ICNC-FSKD 2012, pp. 1991–2002. ISSN: 0898-1221. DOI: <https://doi.org/10.1016/j.camwa.2013.06.027>. URL: <https://www.sciencedirect.com/science/article/pii/S089812211300432X>.
- [4] Lukasz Chmielowski and Michal Kucharzak. "Impact of Software Bug Report Preprocessing and Vectorization on Bug Assignment Accuracy". In: *Progress in Image Processing, Pattern Recognition and Communication Systems*. Ed. by Michal Choras, Ryszard S. Choras, Marek Kurzynski, Pawel Trajdos, Jerzy Pejas, and Tomasz Hyla. Cham: Springer International Publishing, 2022, pp. 153–162. ISBN: 978-3-030-81523-3. DOI: 10.1007/978-3-030-81523-3_15.
- [5] Christopher A. Choquette-Choo, David Sheldon, Jonny Proppe, John Alphonso-Gibbs, and Harsha Gupta. "A Multi-label, Dual-Output Deep Neural Network for Automated Bug Triaging". In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, pp. 937–944. DOI: 10.1109/ICMLA.2019.00161.
- [6] Mainak Dutta. *BEFORE AND AFTER OF DevOps: A PEEK INTO AGILE DevOps*. Nov. 2019. URL: <https://medium.com/%7B%5C%7Dmainakdutta76/before-and-after-of-devops-a-peek-into-agile-devops-3600c26129ac>.
- [7] S. Gujral, G. Sharma, S. Sharma, and Diksha. "Classifying bug severity using dictionary based approach". In: *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*. 2015, pp. 599–602. DOI: 10.1109/ABLAZE.2015.7154933.
- [8] A. Lamkanfi and S. Demeyer. "Predicting Reassignments of Bug Reports - An Exploratory Investigation". In: *2013 17th European Conference on Software Maintenance and Reengineering*. 2013, pp. 327–330. DOI: 10.1109/CSMR.2013.42.
- [9] Aindrila Sarkar, Peter C. Rigby, and Bela Bartalos. "Improving Bug Triaging with High Confidence Predictions at Ericsson". In: *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 2019, pp. 81–91. DOI: 10.1109/ICSME.2019.00018.