DOI https://doi.org/10.15407/csc.2023.02.061
UDC 004.414.23

P.B. TIUTIUNNIK, Student, Computer Systems Software, Department of the Applied Mathematics Faculty, National Technical University of Ukraine "Igor Sikorsky Kyiv Politechnic Institute", Peremohy Ave 37, Kyiv, 03056, Ukraine,
artyoda05@gmail.com

N.A. RYBACHOK, PhD, Asoc. Professor, Computer Systems Software Department of the Applied Mathematics Faculty, National Technical University of Ukraine "Igor Sikorsky Kyiv Politechnic Institute", Peremohy Ave 37, Kyiv, 03056, Ukraine,
ResearcherID: I-5414-2017, ORCID: https://orcid.org/0000-0002-8133-1148,
rybachok@pzks.fpm.kpi.ua

ANALYSIS OF TASKS PARAMETERS OF SOLVE THE PROBLEM OF DETERMINING DELAYS AND RISKS IN AGILE-PROJECTS

Agile methodology is actively used for project management. This article presents the results of determining which task parameters are important in determining delays and risks in Agile-projects. The article provides information on the influence of parameters on the likelihood that a task is a risk or a delay. These parameters are typical for the Atlassian Jira bug tracker.

***Keywords**: Agile-project; Task; Risk, Delay, Machine Learning methods.*

Introduction

In recent years, the *Agile*-methodology has become incredibly popular. according to the 2021 State of *Agile* survey, up to 95% of technology companies use this methodology [1]. During the year of the pandemic, the proportion of software development teams working in an *Agile*-methodology increased from 37 percent to 86 percent. High rates of growth in the use of technology are also observed for teams that did not usually use *Agile*-methods (marketing, HR and finance). This methodology has both advantages and disadvantages.

The main reason for unsuccessful *Agile*-projects is the risks that are not detected at the planning stage and the resulting large delays [2]. The most popular existing methods for identifying risks and delays are not automated and therefore more prone to human error.

Thus, there is a need to develop an approach for identifying risks and delays in *Agile*-projects.

This article offers a solution to the problem of determining tasks parameters, which can be used to solve the problem of determining delays and risks in *Agile*-projects.

The Agile-methodology

The *Agile*-methodology has a large number of principles that have allowed it to reach a high level of use in software development [1]. One of these advantages of *Agile* is the flexibility to change requirements, which is achieved through the iterative and incremental nature of the methodology (adding new features and improvements to an existing product in cycles of several weeks).

The most popular *Agile*-paradigms are SCRUM and KANBAN. They use the division of requirements into small tasks. Such tasks are separate units of work. They can be performed by a single team member and added or removed during the life of the project. The execution of an *Agile*-project can be seen as the completion of many small tasks, each of which can be a delay or a risk. A large number of delays among tasks can cause an *Agile*-project to fail. Therefore, the task of identifying risks and delays in *Agile*-projects is to determine whether each individual task in the project is a possible risk and delay.

Each task is a separate object with different characteristics and can be divided into two classes:

– the task is a possible risk;

– the task is not a possible risk.

The task of identifying risks and delays in *Agile*-projects is to find which of the two classes the task object belongs to. That is, the task of identification is a formalized task of binary classification [3].

The main machine learning methods that learn in a supervised learning way and can be used to solve an automated binary classification problem are the following methods:

– decision tree;

– random forest method;

– artificial neural network;

– naive Bayesian classifier.

To define a task by delay or risk, it is necessary to determine the parameters of the task, on the basis of which this classification will occur.

Task Parameters for Agile-Projects

Teams that use the *Agile*-methodology break down their project into tasks that contain information about a small part of the work that team members need to do and information about the progress of that work. Relevant information can be used to identify whether a given task is a possible risk and delay in an *Agile*-project, and to train and test machine learning techniques.

The content of tasks was analyzed in more detail on the example of issues of the Atlassian JIRA bug tracking system, data from which was used to train machine learning models in this research.

Tasks in the JIRA system consist of fields that are typical for all users of this system and fields added by special plugins. For example, an *Agile*-project team may use different version control systems, such as GitHub or Bitbucket, where each system has its own plugin with unique fields. Only fields typical of Atlassian Jira were used to select the parameters for training the models.

The main characteristics of the task are the name, description, type, priority, the reporter who created the task, the developer who performs the task, comments under the task, history of changes to the task. Most of these features are textual and cannot be used to train machine learning models. Therefore, the following 16 parameters were determined, which can be risk factors of the task.

1. Discussion time. This is the period the team spends trying to find a solution to the problem. An *Agile*-project can be seen as a network of activities, where each activity is registered as an issue whose completion time affects the overall project schedule. For a problem that takes a significant amount of time to resolve, this can cause delays.

2. Waiting time. This time indicates the amount of time an issue is waiting to be resolved, such as waiting for a designated developer to take action. An abnormal wait time is a sign that the problem is being delayed due to a lack of team cooperation, or that no one wants to deal with the problem [4]. The waiting time for a problem starts from the moment the appropriate person is assigned to perform actions to solve the problem.

3. Type. Each issue in JIRA is assigned a type (eg, task, bug, new feature, enhancement, or documentation) that indicates the nature of the task involved in solving the issue (eg, fixing a bug or introducing a new feature).

4. The number of times the problem is reopened. Previous studies on risk identification [5] indicate that the re-opening of tasks (that is, repetitions in the life cycle of the problem) is considered a factor in the deterioration of the overall quality of the software. This leads to additional and unnecessary rework, which contributes to delays. An issue is reopened for a number of reasons, such as when it was not actually resolved properly and needs to be reworked.

5. Priority. This is the order in which an issue should be considered relative to other issues. For example, problems with blocker priority (a problem that blocks other problems) should be considered before other tasks.

6. Change of priority. This is the number of times the priority of the issue has been changed. A change in the priority of a problem may indicate a change in its complexity. For example, some studies [6] use this priority change as a function to predict the blocking error.

7. Number of comments. This is the number of comments from developers during a discussion that can indicate the degree of team collaboration [7]. In past research on the topic of risk identification and delays [8], it was found that the number of comments affects error resolution time: errors with two to six comments are generally resolved faster than errors with fewer than two comments and errors with more than six comments.

8. Number of fixed versions. This parameter indicates the versions for which the problem has been or will be fixed. Issues with a large number of patch versions require more attention from a development, testing, and integration perspective. An intensive validation process is also required to ensure that a fix does not introduce new problems with each patch version.

9. Number of versions with a problem. This parameter indicates the number of versions in which the problem was found. The number of affected versions is an indicator of potential risk, for example, more effort is needed to resolve a problem with a large number of affected versions.

10. Number of Related Issues. This parameter indicates the number of related issues. Linking issues allows teams to create associations between issues. For example, a problem may duplicate another, or its solution may depend on other problems. There are several types of problem references: related, duplicate, and blocking.

11. Number of issues blocked by this issue. Blocking is one type of relationship between issues. This parameter indicates the number of problems that are blocked by this problem.

12. The number of problems blocking this problem. This parameter indicates how many problems block this problem. Solving a large number of blocker problems is more difficult because all blocker problems must be fixed beforehand. Thus, the number of problems with the blocker indicates the time allocated to solving the problem [6].

13. The number of changes in the description. This parameter indicates how many times the description of the problem was changed. Problem description is important for all stakeholders of the problem. Changing the description of the problem indicates that the problem is unstable and can cause confusion and misunderstanding, and is therefore a possible risk factor.

14. Reputation of the reporter. This parameter indicates a relative assessment of the reputation of the reporter, a member of the team that created the task. The reporter's reputation factor was studied in existing works on the identification of possible risks and delays. For example, bugs reported by team members with higher reputations have been found to attract more attention than other issues [9] and are less likely to be reopened [5]. In the context of delayed problem identification, reporter reputation may be a risk factor, as reporters with a low reputation may write poor problem reports, which may lead to longer time to resolve the problem [10]. This work uses the definition of the reporter's reputation proposed in the work of Hooimeijer and Weimer [9]:

$$\text{reputation}(D) = \\ = \left|\text{opened}(D) \cap \text{fixed}(D)\right| \left|\text{opened}(D)\right| + 1. \quad (1)$$

Reputation of Reporter D is measured as the ratio of the number of issues that Reporter D has opened that have been fixed to the number of issues that Reporter D has opened plus one.

15. Developer workload. This parameter indicates the number of open issues assigned to a developer at one time. Developer workload is a reflection of the quality of resource planning, which is critical to project success. Lack of resource planning has implications for project failure [11], and developer workload can have a significant impact on project progress [12]. A developer's workload is (re)calculated immediately after a developer has been assigned an issue.

16. Percent of Delayed Issues Handled by Developer. This parameter indicates the percentage of delayed issues among all issues assigned to the developer. Team members do not have the special skills required for the project, and inexperienced team members are one of the main threats to overschedule [13]. Teams consisting of incompetent

Table. **The results of logistic regression**

Parametr	Logistic regression result
Discussion time	-6,974
Waiting time	-4,689
Type: Bug	-0,904
Type: Documentation	1,16
Type: Improvement	0,247
Type: New functionality	0,83
Type: History	-0,528
Type: Subtask	1,637
Type: Task	0,783
Priority: Blocker	-0,748
Priority: Critical	-0,947
Priority: Important	-0,901
Priority: minimal	-0,704
Priority: secondary	-0,764
The number of times the priority has changed	1,434
The number of times the problem was reopened	3,016
Number of comments	2,963
Number of fixed versions	2,207
Number of versions with a problem	-3,201
Number of related issues	1,415
The number of issues that are blocked by this issue	0,216
Number of issues blocking this issue	1,935
Number of changes to the description	1,678
Reputation of the reporter	-0,683
Developer workload	1,791
Percentage of latency issues dealt with by the developer	3,497

developers can be the cause of project delays [4]. On the other hand, recent studies have shown that the best developers often create the most errors because they often choose or receive the most difficult tasks [14]. This phenomenon can also apply to backlogged problems: the best developers may get the biggest/hardest problems and therefore take the longest time to solve them. A developer may have a large number of backlogs because he or she is an expert developer who is always tasked with solving complex issues.

Results and Discussion

The impact of task parameters on the likelihood that a problem is a risk or delay in *Agile*-projects was assessed. For this, l11 logistic regression was used.

Dataset

To create training data sets the AGILE issues data was collected from 4 open-source software projects that use Atlassian JIRA as their issue tracking system: Apache, Red Hat, Spring and Moodle. Overall, around 1,5 million issues were collected from the issue tracking systems, and only a small fraction of them could be used for method training, because of absence of necessary data to determine delay status of those issues.

Method

For the identified parameters, l1 logistic regression was applied using the log-likelihood function to estimate the effect of each of the proposed tasks parameters on the probability that the task is a risk. In l1 logistic regression, when the parameter is negatively correlated with the result, negative numbers are obtained, while when it is positive, positive numbers are obtained. If there is no influence of the value, the likelihood function goes to zero. To evaluate the influence of each of the proposed parameters of the problems, a dataset created from the problems of open source projects was used.

Results

The results of logistic regression are presented in Table.

As can be seen from the results presented in the table, Waiting-time, Discussion time parameters have the greatest influence on whether the task is a risk. Moreover, the higher the value of these parameters, the more risky the task is. The Percentage of latency issues dealt with by the developer parameter has a great influence on the riskiness of the task. Moreover, the higher the value of this parameter, the lower the probability that the task is a risk.

Conclusions

Identifying risks and delays in projects is an important task when working with the *AGILE*-methodology. To solve this problem, machine learning methods are widely used. The main contribution of this research is the tasks time parameters usage and the definition of parameters that are typical for *Agile*-projects that use Jira bug system. As the research results showed, these parameters (Waiting time, Discussion time) are a significant indicator of whether the problem is a delay or a risk in *Agile*-projects. Further research will be focused on evaluating existing methods and formulating a combined machine analysis method for determining whether a task is a risk or a delay in *Agile*-projects.

REFERENCES

1. State of Agile report. [online]. Available at: <https://stateofAgile.com/> [Accessed 03 Apr. 2023].
2. 9 Reasons for Agile Failures, [online]. Available at: <https://zegal.com/blog/post/Agile-methodology-failures/> [Accessed 03 Apr. 2023].
3. Har-Peled, S., Roth, D., Zimak, D. "Constraint Classification for Multiclass Clas-sification and Ranking." Advances in Neural Information Processing Systems 15 - Proceedings of the 2002 Conference, 2003. ISBN 0-262-02550-7.
4. Pika, A, van der Aalst W. M., Fidge C.J., ter Hofstede A.H., Wynn M.T. "Profiling event logs to configure risk indicators for process delays," Advanced Information Systems Engineering (CAISE 2013), 2013, pp. 465–481.
5. Zimmermann T., Nagappan N., Guo P.J., Murphy B., "Characterizing and predicting which bugs get reopened," in 34th International Conference on Software Engineering (ICSE), 2012. IEEE Press, Jun. 2012, pp. 1074–1083.
6. Garcia, H.V, Shihab E., "Characterizing and predicting blocking bugs in open source projects," in Proceedings of the 11th Working Conference on Mining Software Repositories - MSR 2014. ACM Press, May 2014, pp. 72–81.
7. Bettenburg, N., Just, S., Schroter, A., Weiss, C., Premraj R., Zimmermann, T. "What makes a good bug report?" in Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering - SIGSOFT '08/FSE-16. New York, New York, USA: ACM Press, Nov. 2008, p. 308, [online]. Available at: <http://dl.acm.org/citation.cfm?id=1453101.1453146> [Accessed 03 Apr. 2023].
8. Panjer, L.D., "Predicting Eclipse Bug Lifetimes," in Fourth International Work-shop on Mining Software Repositories (MSR'07:ICSE Workshops 2007). IEEE, May 2007, pp. 29–29.
9. Hooimeijer, P., Weimer, W. "Modeling bug report quality," in Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering - ASE '07. ACM Press, 2007, p. 34, [online]. Available at: <http://dl.acm.org/citation.cfm?id=1321631. 1321639>. [Accessed 03 Apr. 2023].
10. Guo, P.J., Zimmermann, T., Nagappan, N., Murphy, B. "Characterizing and predicting which bugs get fixed: an empirical study of Microsoft Windows," 2010 ACM/IEEE 32nd International Conference on Software Engineering, vol. 1, 2010, pp. 495–504.
11. Han, W.-M, Huang W.-M., "An empirical analysis of risk components and performance on software projects," Journal of Systems and Software, vol. 80, no.1, 2007, pp. 42–50.
12. Porter, A.A., Siy H.P., Votta L. G., "Understanding the effects of developer activities on inspection interval," in Proceedings of the 19th international conference on Software engineering - ICSE '97. ACM Press, 1997, pp. 128–138.
13. L. Wallace and M. Keil, "Software project risks and their effect on outcomes," Communications of the ACM, vol. 47, no. 4, 2004, pp. 68–73.
14. Kim, S, Zimmermann, T, Pan, K, and Whitehead E. Jr., "Automatic Identification of Bug-Introducing Changes," in 21st IEEE/ACM International Conference on Automated Software Engineering (ASE'06). IEEE, Sep. 2006, pp. 81–90, [online]. Available at: <http://dl.acm.org/ citation.cfm?id=1169218.1169308>. [Accessed 03 Apr. 2023].

Received 18.02.2023

П.Б. Тютюнник, студент, Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»,
03056, м. Київ, просп. Перемоги, 37, Україна,
artyoda05@gmail.com
Н.А. Рибачок, кандидат технічних наук, доцент, Національний технічний університет
України «Київський політехнічний інститут імені Ігоря Сікорського»,
03056, м. Київ, просп. Перемоги, 37, Україна, ResearcherID: I-5414-2017,
ORCID: https://orcid.org/0000-0002-8133-1148,
rybachok@pzks.fpm.kpi.ua

АНАЛІЗ ПАРАМЕТРІВ ЗАВДАНЬ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМИ ВИЗНАЧЕННЯ ЗАТРИМОК І РИЗИКІВ У *AGILE*-ПРОЄКТАХ

Вступ. Методологія *Agile* активно використовується для управління проєктами. Виявлення ризиків та затримок у проєктах – важлива задача під час роботи з методологією *AGILE*. Для вирішення цієї задачі широко використовуються методи машинного навчання. Команди, які використовують методологію *Agile*, розбивають свій проєкт на завдання, що містять інформацію про невелику частину роботи, яку мають виконати члени команди, та інформацію про перебіг цієї роботи. Відповідну інформацію можна використовувати для визначення того, чи є дане завдання можливим ризиком або затримкою в *Agile*-проєкті, а також для навчання та тестування методів машинного навчання.

Основним внеском цього дослідження є використання часових параметрів завдань та визначення параметрів, що мають найбільший вплив на рішення, чи є дане завдання можливим ризиком або затримкою в *Agile*-проєкті на прикладі використання системи помилок *Jira*.

Мета. Визначення параметрів, що мають найбільший вплив на рішення, чи є дане завдання можливим ризиком або затримкою в *Agile*-проєктах.

Методи. Для ідентифікованих параметрів була застосована логістична регресія з використанням функції логарифмічної правдоподібності для оцінки впливу кожного з параметрів завдання на ймовірність того, що завдання є ризиком.

Результати. У статті пропонується подолання проблеми визначення параметрів завдань, яке можна використовувати для розв'язання задачі визначення затримок та ризиків в *Agile*-проєктах. Надано інформацію про вплив параметрів завдань на ймовірність того, що завдання є ризиком або затримкою у проєктах *Agile*. Ці параметри є типовими для трекера помилок *Atlassian Jira*.

Висновки. Як показали результати дослідження, параметри, такі як час очікування та час обговорення є показниками того, що завдання може бути затримкою чи ризиком в *Agile*-проєктах. Подальші дослідження будуть зосереджені на оцінці наявних методів та розробці комбінованого методу машинного аналізу для визначення того, чи є завдання ризиком або затримкою в *Agile*-проєктах.

Ключові слова: *Agile*-проєкт; завдання; ризик, затримка, параметри завдання; машинне навчання.