A STUDY ON THE EFFECTIVENESS OF AGILE METHODOLOGY USING A DATASET

Emira M. M. ALZEYANI, Csaba SZABÓ
Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic,
E-mail: emira.mustafa.moamer.alzeyani@tuke.sk, csaba.szabo@tuke.sk

ABSTRACT

There are many methodologies that we can follow in software projects, including the traditional methods, at the forefront of which it was effective and with a good result, but with the constant increase in requirements resulting from the continuous and rapid development in science and technology and digital transformation; Traditional, highly structured project management methods lack flexibility, awareness of change, adaptability to it, and independence; It led to developing this methodology into an Agile methodology in the form it is currently known, which contributed to developing the team's skills and knowledge. Our focus in this paper is a study on a data set affiliated with the Agile methodology by measuring effort using one of the well-known methods, monitoring the actual time and comparing it with the expected time to deliver a specific work, and also identifying the difficulties they face while working in different environments to build a comprehensive picture of the benefits and advantages of the Agile methodology.

Keywords: *Agile methodologies, COCOMO model, Effort measurement.*

1. INTRODUCTION

Technology has changed our lives in the past two decades. It has improved our quality of life and enabled us to connect with people worldwide. It has also radicalised the business sector on a massive scale, which has helped entrepreneurs build their apps seamlessly.

The software development process is a procedure that has several steps. First, we can put it into practice in a few days. A unique and innovative idea may only take a few minutes, but turning that fledgling idea into a fully functional product ready for marketing takes time.

Software development is generally done with some structure in mind, and developers may use one of the models, for example, the waterfall model or one of the methodologies associated with the Agile methodology, to help them build the required software. These methodologies help developers adhere to the methodological standards they work with to ensure project success. Because the primary measure of the success of any software project is the degree to which the project meets the language for which it was designed and the project is delivered to the required standards and on time.

And through V. Tawosi et al. [1] and P. Baláž [2] both of them work under the Agile methodology; P. Baláž [2] works under the Scrum model, which is one of the frameworks according to the Agile methodology and is characterised by flexibility as it provides the opportunity for the customer to change his requirements during implementation, and depends on the union where members work for the team as one and the ability to self-motivate.

The Scrum methodology developed from software projects at the beginning, but it became applied in many different fields, and this transformation led to speed and improvement in several areas.

Scrum depends on dividing the work stages into small tasks and short periods called (sprints), and its duration ranges from one to three weeks. The team begins to work on each sprint with single phases that are recognised by the highest levels of planning, building, testing and revision, and after completing these stages, we have obtained a first product that is suitable for building the rest of the project on and then starting a new (sprint). The stages are repeated each time to obtain the final product at the end of the last sprint.

As for V. Tawosi et al. [1], it depends on the Scaled Agile Framework(SAFe). This is relied upon in the large team that is geographically distributed on a larger scale and depends mainly on the feasibility of all work agreements, including design quality, code feasibility, system quality, and others that help us ensure project delivery despite the large size of the team and the complexity of the project.

The goal of this contribution could be summarised as answering the following two research questions:

1. Is it true that if software development is carried out using agile methodology, project delivery will not be delayed?
2. How can we benefit from knowing the effort put into software development by students studying software development?

So, we need to gather more information about the behaviours and methodology of Agile methodology in more situations by analysing some of the open source datasets to see how effective it is in overcoming the difficulties that any project manager or work team can face. And how we can benefit from it in a way that makes us reach the goal of delivering a project with all the required standards successfully.

2. EFFORT MEASUREMENT

Here we will study the analysis of some datasets of software engineering. We will use one of the models to find the effort to develop software with one of the models recognised on this data set. We will also study some data to monitor the effectiveness of the Agile methodology for software development in delivering projects with the required quality.

To measure the project effort, we need several factors and different characteristics of the measurement process, the most important of which are the number of project codes, the number of working hours during the implementation of the project, the number of people working on the development of the project, etc.

Even measurement methods There are many methods and models for measuring or predicting the effort involved during the project implementation. The most famous models for estimating the effort are the software specialised in software engineering Constructive Cost Model (COCOMO) [4], Software Lifecycle Management (SLIM) and other models. Each of these models has independent and common factors on which it depends in the analysis process. For example, as recognised in the COCOMO model, it depends on cost, scope, and time. The Software Life Cycle Model depends on three essential variables: the requirements (Scope), the employees (the team size) and the tools you need for the project.

Also, the team's experience factor is considered one of the important factors because it depends on determining the time needed for the development process (the relatively high the expert factor, the less time it takes to implement the work) [3].

Other factors affect the legitimate effort, including the language used in the development process, the tools by which the design is carried out, and other tools that can be difficult to understand and require high-level expertise

3. RELATED WORK

Some of the previous studies were used to lay the basis for our analysis processes by knowing the most used tools to manage the software used in our time, and through which it helped us to understand the most used methodology. So through D. Özkan et al. [5], we learned the most important tools used, including Jira and Trello, which are among the most reliable tools since 2015 and are distinguished by helping managers of large projects to obtain the desired result through the same study. It has been shown that distributed teams rely on these intelligent tools more than local teams, which are used to complete some of the most complex communication and collaboration tasks.

The study of P. Serrador et al. [6] aimed to clarify the effectiveness of goal setting and the team's experience in its relationship to the project's success and satisfy the stakeholders through a survey through social platforms, which obtained approximately 1386 projects and were distributed to more than 60 countries. The most significant proportion of the participants were project managers, with 85% and their experience is more than five years. The project's overall success, high efficiency and good satisfaction of the stakeholders characterise most of the participating projects. The success of the results was closely related to the methodology used, as most of them depended on the Agile methodology. With the percentage of failure in the project's success, the majority relied on traditional methodologies.

L. Cheikhi et al. [7] introduced a complete look at data sets that can be worked on by analysts and programmers and used by the project owners by comparing the versions they worked on it. Which we, in turn, used by getting acquainted with these sources and benefiting from them through our study.

S. Goyal et al. [4] show the importance of measuring effort, and its complete relationship to the project's success has been clarified. It is a key factor to ensure the project's success and not wasting effort, time and money. It helps ensure that the project is delivered as required. The well-known stress measurement model COCOMO was developed and supported by neural networks. The extent of the effectiveness of the developed model was clarified by comparing the obtained results and comparing them with previous results on the same selected data set to obtain the best result.

4. THE STATE OF THE STUDY

At this stage, the focus was on the datasets collected from different sources to analyse according to certain conditions mentioned in the following stages of the analysis process. The results and conclusions are monitored through the analysis stage, where the desired goal can be reached. To build a model that follows the Agile methodology, students can benefit during their work in developing projects to overcome the most significant difficulties mentioned in [3].

4.1. Benefits of measuring the effort

Software engineers are primarily responsible for cost estimations and need to appreciate the reasoning behind the actual price of the software. Miscalculations can quickly cause major problems between startups and their customers.

The cost consists of actual time and the effort spent on the project. It has several components, including direct and indirect expenses. The direct cost of software projects mainly includes the income of everyone on the team, payments for the basic hardware used and various expenses on other things required for the business. Indirect costs include quality control, security issues, audits, and other things that may constitute the actual cost of project software development [7].

There are many reasons and benefits for measuring effort through which we can obtain different results and benefit from it in many studies.

1. The project's approval may depend on the effort exerted, which is considered an indication of the success or completion of the project.
2. Project managers are considered one of their tasks to plan and study the project's economic feasibility, which is part of measuring effort.
3. Good coordination and distribution of tasks depend on the effort that will be well distributed among the team members.

4.2. Collection datasets

The first stage was done by searching the datasets. The search process was carried out using specific keywords, including:

- Dataset for effort measurement.
- Dataset for the Agile model.
- Dataset of the Software Engineering Department.

After the collection process, we saved and sorted it until we got the required data set. The sorting process was done by studying the datasets:

- The quality of the existing data.
- The quality of the methodology through which the data was collected.

One of the challenges was finding the required datasets from which we could trace the software development process.

Through the datasets, we can identify the difficulties faced by developers who worked under different conditions, such as geographical distribution, large teams, and others.

We can benefit from this knowledge early by teaching and training the software development stage.

In the second stage, we started the sorting process, i.e. opening and decoding the files according to the downloaded file type and working on the appropriate environment for it. It was found whether these datasets provide the data we need.

The datasets that provided us with the required were kept under the Agile software development environment. The required data is the data that shows the workflow process during the development of the project and not data specific to the program that it developed. After the sorting process, Four datasets were kept for analysis.

4.3. The Datasets

In this section, we will display the datasets which we will use.

4.3.1. Promise Software Engineering Repository Dataset

It provides us with 22 open-source collections that can be downloaded and worked on directly.

After downloading all the datasets and the sorting process was completed. It became clear that there were only two datasets that we could work on, which were the Reuse and Kemerer datasets because they provided us with the required data related to the subject of software project management [8].

- **Reuse dataset**

It consists of 24 projects, each with 28 attributes. These attributes show us the information we need, such as whether the project was successful, the number of team members, lines of code, and other important information.

- **Kemerer dataset**

It consists of 15 projects, and each project has the same attributes, which are eight attributes: Language, KSLOC, EffortMM, etc.

4.3.2. Tscrum_peter_balaz dataset

The dataset was worked by designing a web application to simulate Agile software projects using appropriate artificial intelligence technology to improve, accelerate and facilitate the rapid development of product owners. The application is based on the Scrum model, which consists of the following tables: Project, Sprints, Stories, and Tasks. And it contains 80 projects to show the effectiveness of the designed model [2].

4.3.3. TAWOS dataset

The last dataset is TAWOS, which is open-source and consists of 39 projects. The Jira application compiles it; its latest update was on October 2022. It consists of 13 tables, and the important ones are project, sprint, and Issue. All the projects were compiled by a Jira application consisting of 458,232 problems, and each project has different data in terms of project quality, difficulties, number of developers and other data. But at the same time, they all work under the Agile methodology [1].

5. ANALYSIS OF DATASETS

Here we will start to analyse the datasets that we described in the previous section

5.1. Reuse Dataset

As we mentioned previously, there are few projects in this dataset, but through the small number we have, we were able to work on some analysis of it.

- Find out the number of successful projects.
- Find the relationship between project success, team number, and team skill.
- Find the relationship between projects that did not fail in the delivery process, the relationship between them, and the reasons for failure.
- To find the effort through the number of lines of code.

We can notice (see Table 1) that the successful projects had experience ranging from five years or more to the team and that some of them were based on the Agile methodology, which we can know through the self-organise attribute in the management process. The team was generally between 50 and 200 people, depending on the type of the project.

Table 1 Shows the success project

	Type of Software	Staff Experience	Work Products	Independent Team	assests	Success or Failure
6	Embedded-RT	middle	R+D+C	no	51_to_100	success
9	Non-Embedded-RT	high	R+D+C	no	100+	success
10	Technical	high	R+D+C	no	51_to_100	success
11	Technical	middle	R+D+C	no	100+	success
12	Non-Embedded-RT	middle	R+D+C	no	51_to_100	success
14	Embedded-RT	middle	R+D+C	no	100+	success
15	Technical	middle	R+D+C	no	100+	success
19	Business	low	R+D+C	no	100+	success

But if we note, the projects that were not completely delivered or failed have the same factors as the successful projects (see Table 2), which is why finding a reason to fail was challenging.

Table 2 Shows the failed project

	Type of Software	Staff Experience	Work Products	Independent Team	assests	Success or Failure
2	Technical	middle	C	no	21_to_50	failure
3	Technical	middle	C	no	21_to_50	failure
4	Technical	middle	C	no	21_to_50	failure
7	Technical	high	D+C	no	51_to_100	failure
8	Technical	middle	D+C	no	51_to_100	failure
13	Embedded-RT	middle	R+D+C	no	1_to_20	failure
18	Embedded-RT	high	C	no	1_to_20	failure
22	Business	middle	NaN	NaN	NaN	failure
23	Embedded-RT	not_available	C	no	100+	failure

The second stage is through the number of lines of the code (KLOC) that we can benefit from and extract the effort exerted through the COCOMO model. This model was chosen to measure effort after analysing the data obtained through the dataset. Here we show the results of the effort measurement process, which was carried out through the following COCOMO equation.

$$Effort = a(KLOC)^b \quad (1)$$

To know the time needed to implement or develop the project, we can calculate it through the following equation:

$$Time = c(Effort)^d \quad (2)$$

To measure or calculate the number of people we need to develop this project, we can know it through the:

$$personal\ required = \frac{effort}{Time} \quad (3)$$

(see Table 3) presents the different system categories on which the analysis can be conducted. Here we have the main categories for the clarity of the project complexity, the team's experts.

Table 3 Shows the different categories of the system

Software project	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Here Orient classification was chosen only to measure the effort because this research is based on the methodology that can be followed by students with less experience or who will not work in a large team. The results obtained according to the Reuse dataset, we only had three values for the lines of code (KLOC), from which we obtained the effort measurement results as shown in the following (see Table 4):

Table 4 Effort of Reuse dataset

KLOC	Effort	Time	Person required
500	1637	51	39
100	302	21	14
10	26	8	3

We can note that the results obtained are that all projects here do not need a great effort or a long time to develop, and the students have to rely on a suitable methodology to help them not exceed these numbers, which will help them in the process of dividing tasks and successful the development.

5.2. Kemerer dataset

Through this dataset, we can obtain the effort by taking a value of data on the number of lines of code (KLOC), and we used the COCOMO model to find the effort, time and person required. Finally, we used organic categories of the system of the values shown in (see Table 3) and the obtained results (see Table 5).

Table 5 Effort of Kemerer dataset

Project	KLOC	Effort	Time	Person required
1	253.6	802	31	25
2	40.5	115	15	7
3	450	1465	39	36
4	214.4	617	29	22
5	449.9	1462	39	36
6	50	145	16	8
7	43	124	15	7
8	200	625	28	21
9	289	920	33	27
10	39	112	15	7
11	254.2	804	31	25
12	128.6	391	24	16
13	161.4	498	26	18
14	164.8	507	26	19
15	60.2	176	17	9

5.3. Exploratory Data Analysis

In this stage, we will use the Exploratory Data Analysis (EDA) model for the data analysis. It helps us compare the obtained data and the effectiveness of the selected models, which depend on the Agile methodology. This model relies mainly on detailed and multi-format graphs that can choose according to the need in data analysis, through which we can build complete knowledge about a specific topic by discovering errors or strengths in the data or gaining new knowledge from the process of analysis. Many tools can help us work with (EDA), programming, and non-programming. For example, the most famous programming language it can work on it is Python, which was relied upon in our paper, and also the R language, which is recognised as one of the significant statistical languages [9].

As mentioned previously, we used Python for the analysis process because an open-source language with many libraries helped us in the analysis process.

One of the most well-known libraries is Pandas, which helps us clean, convert and analyse data as required smoothly, and we worked with it under the Jupiter Notebook environment. It was used because of the ease of dealing with it and the flexibility in its use [10].

5.4. Tscrum_peter_Balaz dataset

The purpose of the data that has been worked on is to test the actual extent of the model designed to facilitate the development process using the prediction model, which works based on collected expectations. The team member who will be assigned the task is selected by selection according to the occupancy of the members [11]. The main tasks are (see Fig. 1)

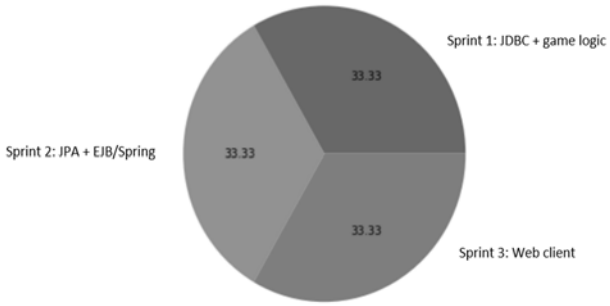


Fig. 1 The main tasks

We will compare the final approximate time(ended_at) with the last adequate time (updated_at) of the Sprint table for a specific project (see Fig. 2):

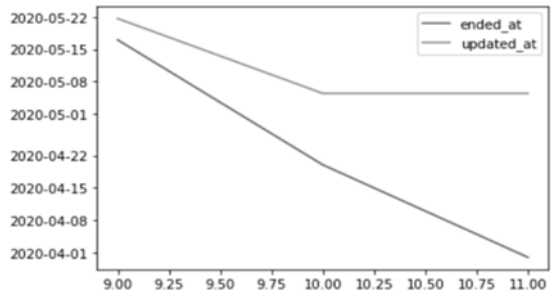


Fig. 2 Project number 7

Here we chose one of the projects to compare, and as we mentioned earlier, each project has three main tasks (Sprint). The data are based on group projects for instructional material that they must complete on time; the supervisor sets a specific time for the beginning (started_at) and end (ended_at) of the task and the results obtained here. There are minor differences compared to the supervisor's (ended_at) and the student's (updated_at).

We randomly chose some projects to get different results (see Fig. 2 and Fig. 3).

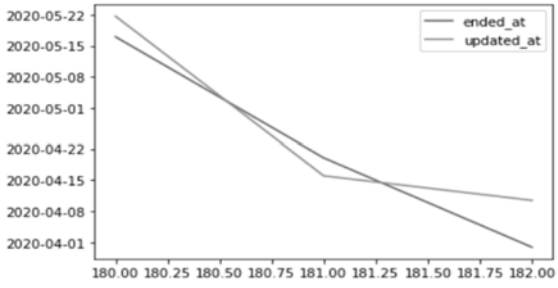


Fig. 3 Project number 66

The Stories Table shows the detailed eight tasks for each project (see Fig. 4):

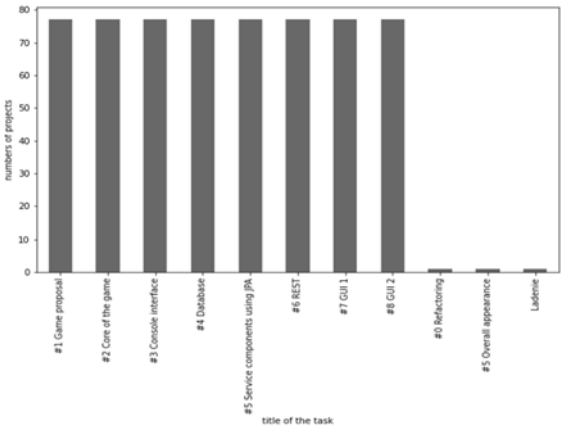


Fig. 4 Tasks of Stories Table

The comparison of (created_at) (updated_at) the same projects we worked on previously (see Fig .5 and Fig. 6):

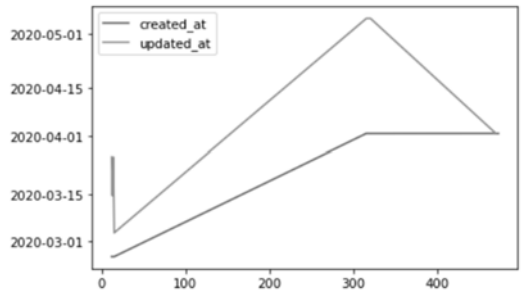


Fig. 5 Project number 7

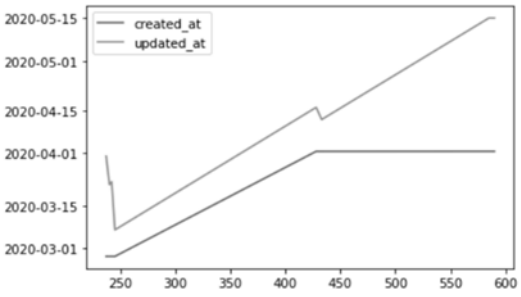


Fig. 6 Projects number 66

We can notice the discipline in the project delivery times (see Fig. 7), except for a few who consider that they missed the delivery date for unknown reasons.

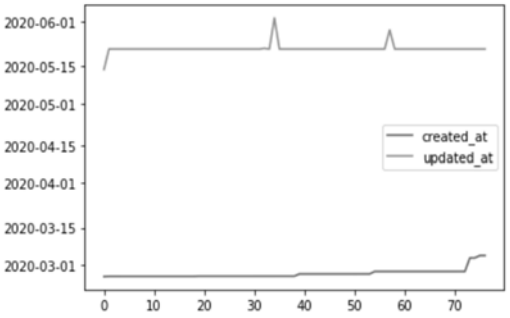


Fig. 7 Shows all projects

We got the desired result through the previous model by delivering all the required tasks on time. This shows the

effectiveness of the model designed and applied through easy-to-handle user interfaces that depend on the prediction model to facilitate the development process and ensure the project's success.

5.5. TAWOS dataset

Here we begin the process of analysing the dataset of TAWOS, which, as we mentioned above, is open source and was designed and configured to help specialists in the field of software engineering develop their work by working on studies and analysis.

In the beginning, we will display the projects stored in the data set (see Fig. 8):

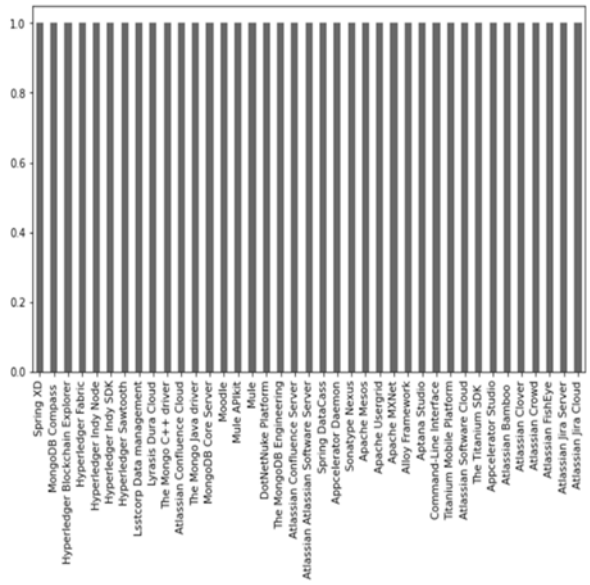


Fig. 8 All projects in the TAWOS dataset

Each of these projects has many essential attributes. The most important of these attributes is the project delivery time (see Fig. 9) . We will notice that all these projects are successful through the date of their last update.

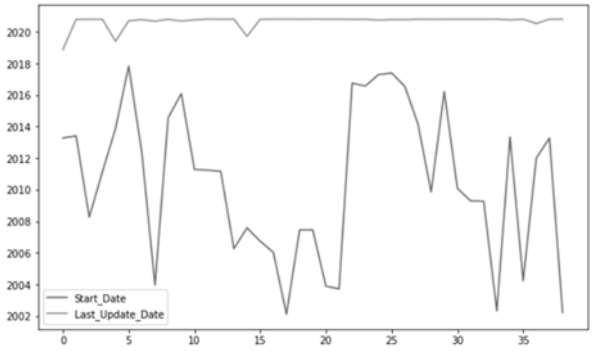


Fig. 9 Start_date and last_update of projects

Here is the Sprint to study it through the time sequence for each task of a specific project. The projects will be chosen randomly (see Fig. 10. And Fig .11). Each project has a different number of sprints than the others, so we have presented only a particular group of Sprints.

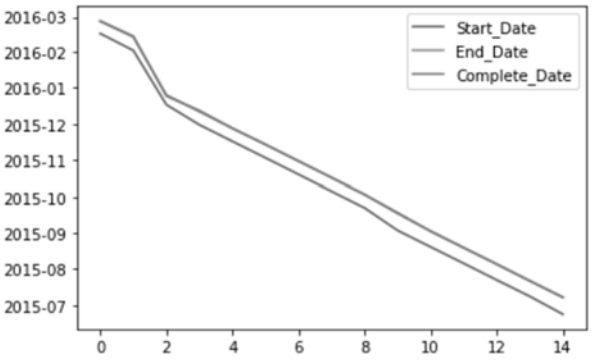


Fig. 10 Project number 1



Fig. 11 Project number 20

Each project had several problems, such as a problem during a specific task or a problem with a particular code during the development process. There were also post-delivery problems and other difficulties that faced developers, project managers, and users (see Fig. 12):

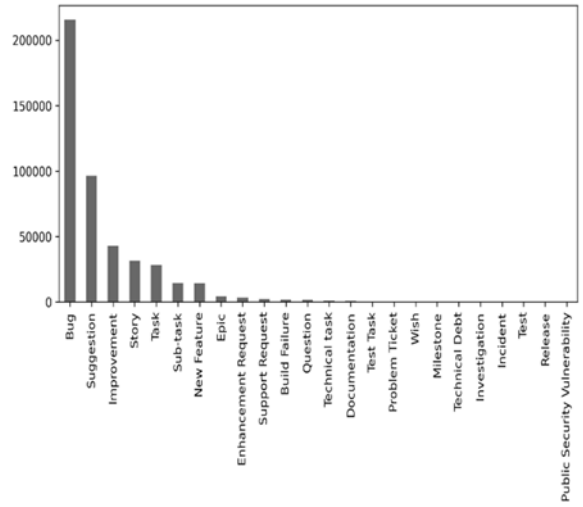


Fig. 12 The issues of the TAWOS dataset

As we notice in the previous figure, all projects have, all projects have different problems. The most common issues are Bugs, which can be technical problems or problems in the resources used or software because they are considered more comprehensive in the definition.

These problems were explained by their causes through observations. The most significant majority was a problem with a specific code or failing to work a particular function correctly. The extent of the issues was on the workflow for not adhering to the specified time for delivery and skipping it.

Here we will display the timing in which some of these issues were resolved for projects and problems randomly chosen.

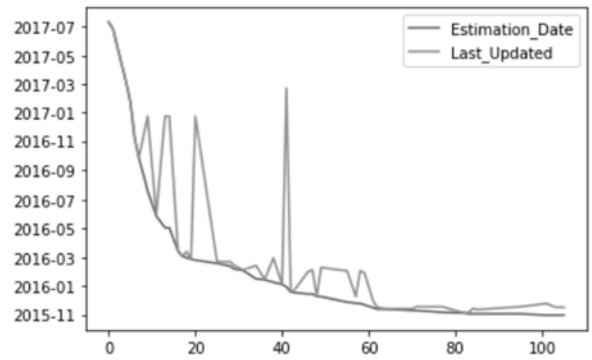


Fig. 13 Project 1, type of Issue (Bug)

Here we presented about 50 difficulties, and we can notice (see Fig. 13) that sometimes there was a big difference between the expected time (Estmation_Date) to solve the problem and the actual time (Last_Updated) to solve it. However, it is due to some factors, including the problem's difficulty, the difficulty of sequencing it may depend on another issue that still needs to be solved and other factors.

(See Fig. 14), another type of problem was presented for a different project, showing the result. There is no significant difference between the expected time and the actual time to solve the problem, not in the dimension of the situation that could be due to another problem.

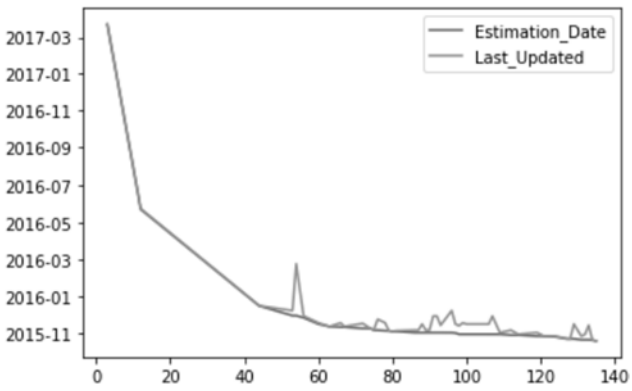


Fig. 14 Project 33, type of Issue is Story

6. CONCLUSIONS

Here we conclude that whenever the Agile methodology is dealt with consistently and strictly between the project manager, team members, and users, it results in the desired result, which is the success of the project and delivery on time and within the required requirement. This study examines the effectiveness of the Agile methodology and the Scrum model in software engineering management. And through the different set of data that has been worked on, we have obtained that every difficulty or problem we can face can be overcome through good management, work monitoring and interdependence between team members. Thus, we guarantee the effectiveness and success of the Agile methodology.

We got our first goal through the previous study by analysing some of the open source data sets, which had a very high success rate for the project without delay. In some cases, there were some delays whose causes were unclear, and the breakdowns were comprehensive (see Fig. 12). Despite these breakdowns or difficulties, agile projects always deal with difficulties better. Therefore, the projects are delivered on time and in the required form.

And for the second goal and through our study on [3], it was clarified that the student does not go into details about the project at the beginning of the project development stages and shows a lack of commitment to it, which causes it to fail at some stages. And delay a delivery—inability to fully adhere to the plan, incompatibility in thoroughly studying the requirements, and other reasons. Therefore, the importance of knowing the effort that will be worked on in the project was clarified from the first stage of the development process so that it helps us avoid the difficulties they faced and check all the initial details on which the project will be based.

REFERENCES

[1] TAWOSI, V. – AL-SUBAIHIN, A. – MOUSSA, R. – SARRO, F.: "A Versatile Dataset of Agile Open Source Software Projects," 2022.

[2] BALÁŽ, P.: "Web Application for Agile Software Project Simulation," TU of Košice, Košice, 2022.

[3] ALZEYANI, E. M. M. – SZABÓ, C.: "A study on the methodology of Software Project Management used by students whether they are using an Agile or Waterfall methodology," in ICETA 2022, pp.22-27, 2022.

[4] GOYAL, S. – PARASHAR, A. "Machine Learning Application to Improve COCOMO Model using Neural Networks," International Journal of Information Technology and Computer Science (IJITCS), pp. 35-51, 2018.

[5] ÖZKAN, D. – MISHRA, A.: "Agile Project Management Tools: A Brief Comprative View," pp. 17-15, 2019.

[6] SERRADOR, P. – PINTO, J. K.: "Does Agile work? — A quantitative analysis of Agile project success," International journal of project management, vol. 33, no. 5, pp. 1040-1051, 2015.

[7] CHEIKHI, L. – ABRAN, A.: "PROMISE and ISBSG Software Engineering data repositories: A survey," 2013.

[8] USMAN, M. – MENDES, E. – WEIDT, F. – BRITTO, R.: "Effort Estimation in Agile Software Development: A systematic literature review," in PROMISE '14: Proceedings of the 10th International Conference on Predictive Models in Software Engineering, 2014.

[9] SAHOO, K. – KUMAR SAMAL, A. – PRAMANIK, J. – KUMAR PANI, S.: "Exploratory Data Analysis using Python," International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, no. 12, pp. 4727- 4735, 2019.

[10] RAHMANY, M. – MOHD ZIN, A. – SUNDARARAJAN, E. A.: "Comparing Tools Provided By Python And R For Exploratory Data Analysis," pp. 131-142, 2020.

[11] BALÁŽ, P. – MUSTAFA MOAMER ALZEYANI, E. – SZABÓ, C.: "Web Application for Simulation of Agile Software Projects – Stage One: Team Builder," *Journal of Applied Technical and Educational Sciences,* vol. 12, pp. 1-11, 2022

[12] TARIQ, S. – USMAN, M. – FONG, A. C. M.: "Selecting best predictors from large software repositories for highly accurate software effort estimation," 2020

[13] DE SILVA, I. J. – RAYADURGAM, S. – HEIMDAHL, M. P. E.: "A reference model for simulating Agile processes," 2015.

[14] ALZEYANI, E. M. M. – SZABÓ, C.: "Comparison of Agile Software Project Management Methods," in 22nd Scientific Conference of Young Researchers, 2022.

[15] ALZEYANI, E. M. M. – SZABÓ, C.: "Simulation Model for Agile Software Project Management," in 2021 19th International Conference on Emerging eLearning Technologies and Applications (ICETA), 2021.

[16] NAJAH MAHDI, M. – HAZLI MOHAMED ZABIL, M. – AHMAD, A. R. – ISMAIL, R. – YUSOFF, Y. – KOK CHENG, L. – SUFYIAN BIN MOHD AZMI, M. – NATIQ, H. – HAPPALA NAIDU, H.: "Software Project Management Using Machine Learning Technique—A Review," 2021.

[17] ALZEYANI, E. M. M.: "DIGITAL SKILLS IN AGILE TEAMWORK ENVIRONMENT," in Skills for (post) digital society, Košice (Slovakia), 2022.

Received December 14, 2022, accepted February 23, 2023

BIOGRAPHIES

Emira Mustafa Moamer Alzeyani received her M. Sc. from the Faculty of Electrical Engineering and Informatics, the Technical University of Košice, in 2018; she was working as an assistant lecturer at the College of Electronic Technology - Tripoli, Libya. She is a second-year PhD student at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice. Her research interests include Agile software project management improvements and artificial intelligence in Agile projects**.**

Csaba Szabó received his M. Sc. From the Faculty of Electrical Engineering and Informatics, Technical University of Košice (TUKE) in 1998 and PhD in Informatics from TUKE in 2008. He is an associate professor at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice. His research interests include classic and Agile software project management, green software and intelligent systems.