Mining Github for Factors That Affect Open Source Software Sustainability

Mohammad Azees Abdulhassan Alshomali[1*], Wurood Albayati[2]

[1,2]*Mustansiriyah University, College of science, Computer science dept., Baghdad, Iraq;*

E-mail: mohammada.abdulhassan@uomustansiriyah.edu.iq[1] ,w.albayati@uomustansiriyah.edu.iq[2]

Abstracts: Open-Source Software (OSS) is everywhere. The availability of such software enables researchers in software engineering to have a deep insight into the factors that affect the success of the software. Some OSS repos get more popular and evolve over time, while others may only survive for a couple of months. This study aims to help developers identify internal factors that affect the sustainability of their software. Firstly, identify the most demanding application domain; secondly, observe the most popular repositories (for the demanded application domain) for about three years to identify the factors that help such repos to survive (sustainability factors). Dara mining algorithms (classification and regression) are used as a tool to find the GitHub factors that affect the sustainability of OSS.

Keywords: Software, Repositories, GitHub, Regression, Data mining , Open Source Software, Sustainability.

1. INTRODUCTION

Open-Source Software (OSS) become more and more popular because of the cost and functionality of this type of software. OSS is built by a team of volunteer developers (S. S. Bahamdain, Open Source Software(OSS) Quality Assurance: A Survey Paper. International Workshop on Enterprise Web ApplicationDependability , 56 p. 460, (2015).). For each software project, a small team usually starts coding and uploads their work to one of the OSS websites, so other developers can participate or contribute if they like. Repositories (repos for short) are used to hold the code of the developer, contributors, and all other documentation and collaboration efforts (Bird, Gourley, Devanbu, Swaminathan, & Hsu, 2007; Coelho & Valente, 2017; Mens, Claes, & Grosjean, 2014). The repo is usually shared publicly. Contributors are a larger group of developers who have the right to commit changes to the projects (S. S. Bahamdain, 2015; Baharuddin, Izhar, & Shoid, 2018).

OSS requests a website host that provides all the required facilities to communicate between developers(Wang, Poo-Caamano, Wilde, & German, 2015). Source Forge, GitHub, and BitBucket are examples of such hosts. The OSS ecosystem is a collection of elements related to each other. Accordingto (Bird et al., 2007; Coelho & Valente, 2017; Mens et al., 2014) OSS is not mature yet; because of a lackof understanding of the OSS ecosystem and the elements that affect it in addition to the security issues relates to risk assessment that developers may not aware of (Khalil, Haider, Al-Shamayleh, Akhunzada, &Gani, 2023) . The availability of OSS enables researchers in software engineering to have a deep insight into the factors that affect the success of software.

The widespread use of free software enables many OSS developers to maintain their repositories, new versions, and updates to continuously improve their software. The existing OSS covers many application domains, such as: business, mobile applications, learning, gaming, and utilities. The lifetime of OSS depends not only on the user's needs but on many other factors. Some OSS repos get more popular and evolve over time, while others may only survive for a couple of months. This study aims to help developers identify internal factors that affect the sustainability of their software. It answers the followingquestions:

- What is the most popular (demanded) application domain?
- What factors affect software sustainability?

To answer the questions, firstly, we identify the most demanding application domain, and secondly, we conduct a long observation for the resulted application (the demanded application domain). In this study, the observation period was three years. Any software that survives this observation period is considered sustainable (if it maintains its position among the most popular repos). The observation period will beused to identify the factors that help such repos to survive (sustainability factors). Data mining algorithms (classification and regression) had been used as a tool to achieve our goal.

In this paper, the top 568 GitHub repositories (dataset) were collected, filtered, classified (according to repositories' application domain), and then observed for more than three years, to predict factors that enable such repositories to survive. The results showed that learning is the most popular application domain, followed by mobile applications, business applications, gaming, and utility. There wasn't too much of a gap between mobile and business applications. We found that factors affecting OSS depend highly on application domain; thus, for the most popular application domain, the effects of OSS ecosystem were calculated.

Factors that affect business and mobile application sustainability are *watchers, pull requests*, *stars, issues, and releases.* The most important factors developers have to worry about to get sustainable and healthy business apps are watchers and pull requests. The number of stars the software gets is not important in business applications. On the other hand, stars play a central role in the mobile application domain, which is a dynamic domain. In mobile application software issues resolved are an important measure that developers have to keep an eye on. The resolved issues usually lead to new releases (that's true for fast growing mobile markets), and repos with more releases stay active and sustainable.

The paper is articulated as follows: Section 2 reviews the literature and related works. The methodologyis presented in Section 3. Section 4 displays results and discussion. The conclusion presented in section 5.

2. literature Review

Research in the OSS ecosystem increased but was only slightly done from the viewpoint of software engineering(Andrea Capiluppi, 2019). Open source ecosystems have been studied by many researchers (Alshomali, 2018; Andrea Capiluppi, 2019; S. S. Bahamdain, 2015; Baharuddin et al., 2018; Bird et al., 2007; Coelho & Valente, 2017; Farias, Wiese, & Santos, 2019; Franco-Bedoya, 2015; Jansen, 2014; Jiang

et al., 2016; Johannes Wachs, 2022; Konstantions Manikas, 2013; Mens et al., 2014; Palvia, 2012; Peterson, 2014a; Rouder, 2016; Valiev, 2021; Wang et al., 2015; Xanthopoulos & Xinogalos, 2013), Alshomali proposed a general model for the OSS ecosystem (using GitHub sites) depending on programming languages. Andera Capiluppi, on the other hand, derived OSS ecosystem that depends on the application domain.

Franco Bedoya (Franco-Bedoya, 2015) mentioned the need for modeling the OSS ecosystem; thus, he developed a framework for the OSS ecosystem, claimed that his model supports the synthesis, analysis, and evolution of the ecosystems . While Mens et al.(Mens et al., 2014) trace the OSS ecosystem using the biological ecosystem framework. Other researchers found that OSS developments did not follow the software engineering process (Konstantions Manikas, 2013). Munaiah (Munaiah, Kroh, Cabrey, & Nagappan, 2017)used data mining (classification) to identify projects that have engineered software. The author extracted his dataset from GitHub repositories, and he claimed his classifier is better than others that used Stargazer as the main classification criteria, Munaih used eight dimensions (documentation, issues, unit tests, architecture , community, history, license, and continuous integration) as the bases for his classifier in searching for software that had an engineering process.

Fork and star are used as basic project success factors (Chen, Li, Jiang, & Zhang, 2014; Coelho & Valente, 2017; Pipinellis, 2018). The number of core developers differs from project to project; generally speaking,core developers in OSS may change as projects evolve. When users fix a bug or suggest a change to a project, they

participate in project development, and they may then become contributors (when pull requests are closed on the GitHub platform, for example), but the increasing number of contributors may have a negative impact on the project (Song & Kim, 2018).

Farias et al. (Farias et al., 2019) studied the role of the influencer in the OSS ecosystem. In this paper, contributors represent all project developers (influencers, according to Farias definition, are all developers who influence and contribute to the project). They indicated the importance of commits and code in the ecosystem and OSS success. In this paper, commit is one of the factors that, when it is high, means the project is still active and demanded by the user. In OSS, popularity indicates success, so developers thrive on making their projects popular. The increasing popularity of software will encourage newcomers to follow OSS projects and maybe become contributors in the future. Popularity means a number of stars, the time required to review pull requests, programming languages, and project age (Fronchetti, Wiese, Pinto, & Steinmacher, 2019).

Janssen (Jansen, 2014) studied OSS ecosystem health in 2014 and produced a framework that measures OSS health to help stakeholders, developers, the government, and users. By ecosystem health, he meant success and evolution. Janssen believed in the role of the application domain in OSS ecosystem health, and he studied software health, focusing on software success and sustainability.

The number of downloads and the existence of supported platforms are considered important metrics from Janssen's point of view (in addition to others), Janssen stated the importance of the three pillars in any OSS ecosystem. Alshomali's 2018 findings in 2014, applied precisely with a clear definition and specific model by Alshomali in 2018 (Alshomali, 2018), suggested an OSS ecosystem that ticks all the boxes (the three pillars) suggested by Janssen, But, Alshomali considered factors at the network and project levels.

Alshoamli developed a general OSS ecosystem model based on eight factors that affect the OSS ecosystem. His research aims to determine specific factors that represent keystones that developers should focus on when designing software . Table 1 below shows the difference between our work and the related article. The literature in (Andrea Capiluppi, 2019; Fronchetti et al., 2019; Jansen, 2014) agreed that the application domain plays an important role in the OSS ecosystem, this is the point at which our idea meets with them. Capiluppi et al, parsed source code to extract and classify application domains for OSS and found that mobile applications were among the most famous application domains in OSS.

In 2020, Marat Valiev (Valiev, 2021) investigated the role of external factors in software sustainability, Reused libraries are considered an example of external factors the thesis studied. The authors give simple solutions that help maximize the sustainability of OSS.

Table 1 shows the relationship between our work and related literature.

Resource No.	Main idea	Related to the study
(Andrea Capiluppi, 2019; Fronchetti et al., 2019)	The importance of the application domain	The most demanding application domain considered
(Song & Kim, 2018)	An increased number of contributors will have a negative impact on sustainability.	The number of contributors has no effect on sustainability.
(Jansen, 2014)	Software health help to maintain sustainability (No. of download, platform)	Downloads and platforms were not considered here; the study focused on the internal factors that support sustainability.
(Alshomali, 2018; Mohammad Azeez Alshomali, 2017)	determined the internal factors that affectthe ecosystem of OSS	urther observation of the factors is needed to determine the important internal factors that have proven to be important for sustainability.
(Valiev, 2021)	Study the external factors that affect OSS sustainability.	This study Only considered internal factors.

3. Methodology

To study OSS, a platform must be carefully selected, taking into account different OSS platforms (GitHub, Bitbucket, Azure DevOps Server, GitLab, Buddy, and others).

- The first step is to create an account on the OSS platform.
- Secondly, collect the most popular repos on the selected platform.

Thirdly, eliminate the outliers, and then, use the keywords to classify the repos according to their domain. Next, the resulting repos have to be observed further for three more years to ensure their sustainability. Finally, obtain the factors that have more influence on sustainability by using.

It is important to mention that words like app (or apps) and repos (or repos) were used interchangeably in this study, but they have the same meaning, both mean software. The methodology consists of six main processes depicted in Figure 1.

3.1. Selecting OSS platforms

As mentioned earlier in the methodology, an OSS site (host, or platforms) had been chosen, GitHub provides a platform for sharing, updating, and controlling the project in an easy way, it also provides all the developer needs to start and manage a project (Mohammad Azeez Alshomali, 2017; Valiev, 2021). GitHub GitHub is the world's largest collection of open-source materials (Bai, Liu, Meng, & Liu, 2023; Dabbish, Stuart, Tsay, & Herbsleb, 2012; Johannes Wachs, 2022; Peterson, 2014b; Wang et al., 2015). GitHub facilitates: communication among developers, creates repositories for OSS projects,and offers attributes used to measure project popularity and success (Farias et al., 2019; Fronchetti et al., 2019; Mohammad Azeez Alshomali, 2017).

To report bugs in software or manage tasks developers planned to accomplish, the "issues attribute is used. The user could use the software (repo), but he/she had to fork it. The user could suggest any changes to the software through a pull request. All documentation and historical information could be easily accessed using discussion attributes.

Since the study seeks to investigate software sustainability, 568 of the most popular repos have been chosen. According to literature (Alshomali, 2018; Chen et al., 2014; Mohammad Azeez Alshomali, 2017; Palvia, 2012), the most popular repos, are those that have a higher number of stars and forks. All the selected repos were programmed using JavaScript. Because the JavaScript language is very popular and itis very easy to program any app with it.

The 568 repos have been chosen carefully, considering not only the popularity of the repos, but, the time they were created (all selected repos were created in 2017 or earlier). Next, apply the data mining concept of data cleaning and classification. The aim of data cleaning was to remove the outlier, which means repos with a minimum number of developers and too many stars, according to (Alshomali, 2018), such repos are questionable (illegal). The resulted dataset consists of 554 repos.

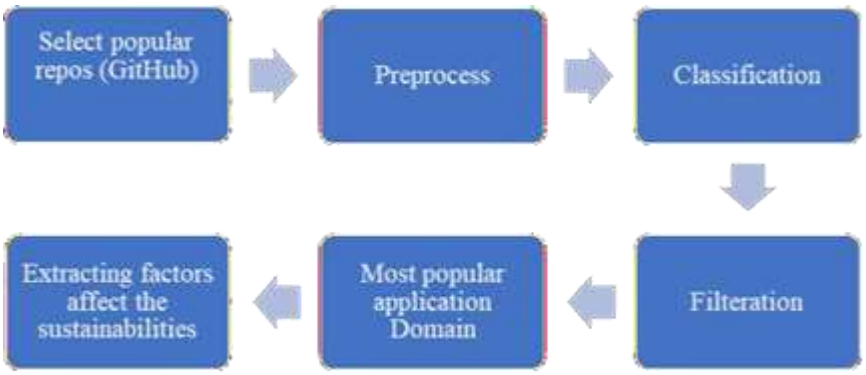

Figure 1 Research methodology block diagram

3.2. Classification

The study suggests six classes cover the main application domains, The application domains considered were: mobile, business, learning, library and utility, and others. To extract the application domain from the 554 collected repos (dataset), the classification criteria used keywords extracted from repo documentation. Not all the documentation files used, including the readme file (in addition to therepository name if required), had been considered.

The repo creation process on the GitHub platform follows a specific set of steps, one of which is the creation of a readme file, so every repos in GitHub has to have one (Prana, Treude, Thung, Atapattu, & Lo, 2018). It's automatically created when creating a new repo. For each application domain, we build a dictionary that has common words associated with that specific application domain.

Table 2 shows the keywords associated with each application domain. These keywords were carefully selected after observing the readme file manually for 50 randomly chosen repos. MATLAB was used to program the classifier; it connects easily with GitHub.Table 1 keyword associated with each application domain.

The keywords enter the first part of the classifier, which is the parser. For each repo, the parser will analyze the readme file, searching for keyword matches. If there are no keywords that match, then the parser will search the repo name for matches. The classifier structure is depicted in figure 2 below. For example, in a business application, the keywords bitcon, business, or e-store should appear in the name or description parts of the readme file. Whenever there is a match, the repo name and URL will be stored in an Excel file under the business application domain.

The classification results (for all 554 datasets) are in six Excel files, each with a specific application domain. The result of the classification shows that the majority of the repos were learning repos, followed by mobile applications, business, gaming, and utility.

From the OSS ecosystem perspective, the learning repos do not maintain all eight factors of the ecosystem, thus, such repos have been eliminated from the study. Also, such an application domain witnessed unusual growth during the COVID-19 pandemic; new repos became popular rapidly, and some old repos stayed in competition. From there, business applications and mobile applications are the most popular application domains, which were the main focus of this study, the other application domains werenot considered here.

Table 2: Keywords associated with each application domain

Application domain	Keywords
Mobile	Android

	Ios or ios
	Tablet or ipad
	Mobile phone
	Iphone
Business	Apps or app
	Bitcon
	Business
Learning	e-store
	Guide
	Learning
	Books
	Lectures
	Course or courses
Library and utility	Languages
	Adds on
	utility
Gaming	Libraries
	Gaming
	Puzzle
	strategy
	Boardgame
	Play or play it now or play it
Others	Any repos contain words not in the keywords

After the elimination, the resulted dataset consists of 224 repos (118 repos for business applications). 106 repos for mobile apps). For the sustainability test, a second revisit to the dataset in 2022 has been done, to test the status of the repos (are they still popular?), 200 repos out of 224 remain popular in 2022. Those were considered the datasets for the sustainability study.

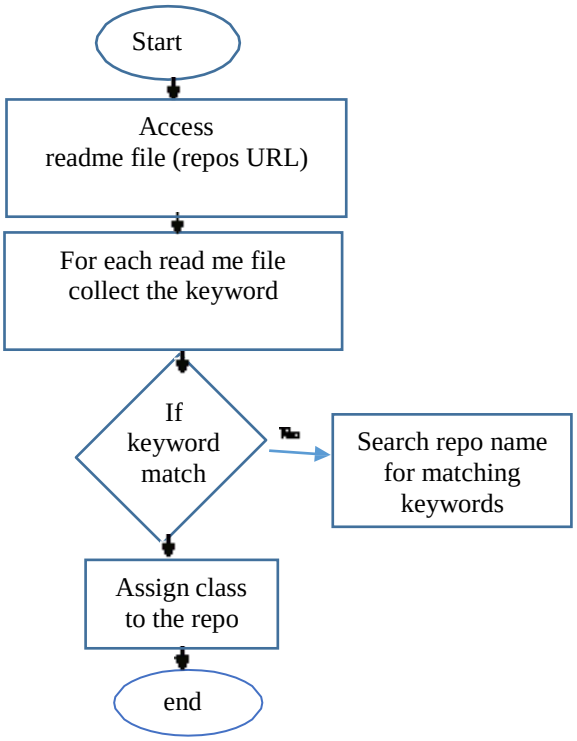

Fig. 2 shows the classification algorithm used in this study.

3.3. Extract Factors that affect sustainability

In the extraction process, the 200 observed repos were further processed using statistical methods to identify factors that play central roles in repos sustainability. GitHub provides a number of parameters (eight factors) that help the developer track the activities of his or her repo. Those factors are suggested by (Ahmed Majid Taha, 2018; Andrea Capiluppi, 2019; Baharuddin et al., 2018; Chen et al., 2014; Farias et al., 2019; Franco-Bedoya, 2015; Konstantions Manikas, 2013; Munaiah et al., 2017; Song & Kim, 2018). The study, used the same framework suggested by Alshomali(Alshomali, 2018) to investigate how that framework will perform in the case of mobile applications and how it will perform for business applications.

Eight GitHub factors, associated with each repo, have been observed, which are: fork, watch, stars, pull, issues, contributors, release, and commits. These values were extracted during the first collection of the dataset, after three years, the values were checked again to make sure the repos were on top (sustainingthe same level of popularity). To ensure the validity of the dataset, an ANOVA approach was used, and the result of the ANOVA test showed the presence of a significant difference. In other words, our dataset was valid. As stated in (Andrea Capiluppi, 2019), the application domain influences the repository activity level and popularity, hence, for the most popular application domain, the strength path among those eight values has been calculated twice. The first time, the calculation was done using the dataset for the business application domain, while the second one used mobile application. Tables 3 and 4 illustrate the results of applying the structural path method to mobile and business repositories consecutively. Note that the stars in the tables refer to P values where the existence of one star(*) before the value means p< 0.05, two stars(**) mean p< 0,01, and three stars(***) mean p <0.000.

In tables 3 and 4, cells with an orange background represent the best elements that help to increase the commits. When the software has more commits, it means more change will be saved to the software (the software has been updated according to the commit message), in other words, the repos that contain the software are still active. The largest value of the commits, means the software is sustained. It could be seen from the two tables that there is a strong path. This indicates that, the factors in the strong path arethe ones that influence the activity level of the repos. In other words, increasing those factors will ensure the sustainability of the software. For mobile apps (table 3), the strongest path was stars-issues, issue- release, and release-commits, while in business applications, the path was watch-pull and pull-commits.

Table 3 shows the path strength to each GitHub attribute for mobile application software.

	Fork	Watch	Stars	Contributors	Pulls	Issues	Release	Commits
Fork		***0.9	***0.87	-0.4			-0.1	
Watch	***0.9		***0.9	0.53	0.34			
Stars	***0.87	***0.9				0.6		-0.3
contributors	-0.4							0.32
Pulls				0.3				0.35
Issues				0.27	0.43		0.78	
Release								0.49

Table 2 shows the path strength to each GitHub attribute for business application software.

	Fork	Watch	Stars	Contributors	Pulls	Issues	Release	Commits
Fork		***0.93	***0.87		0.36	0.36		0.19
Watch	***0.93		***0.91		0.58			
Stars	***0.87	***0.91				-0.26		0.27
contributors								0.17
Pulls				0.46				0.64
Issues				-0.24	0.43		0.39	
Release								0.10

4. Result and Discussion

The study attempts to answer two main research questions, which relate to each other. The research questions were:

- What is the most popular application domain in the software market?
- What factors affect the sustainability of software?

The dataset used in this study consists of 568 popular GitHub repos. Popularity represents the number of stars and forks (forks relate to downloads from repos), which means we seek not only one measure of popularity but also need to make sure that OSS is still in use and users download it.

Those repos were collected, pre-processed, classified, and observed for four years starting early in 2019. The dataset size for this study was 554 for the first question. Then, after the second revisit, we ended up with

200 repos (eliminating the learning repos and repos that got out of gaining more stars (not popular any more or not active). The 200 repos used to calculate the affected factors in Oss sustainability The results were as follows:

a.What is the most popular application domain in software markets?

After the preprocessing step, 554 repos were good enough to enter the classifier. To answer this question, a classifier had been built using keyword analysis techniques. For each of the 554 repos, the readme file and repos name (extracted from repos documentation) were entered into our classifier (see figure 2).

MATLAB was very useful in working with repos to auto-extract the required files. The reason behind using application domain as a basis for classification was the effect of application domain on the reposlife cycle (Andrea Capiluppi, 2019). The most popular application domains were:

Learning, followed by mobile applications, then business applications, gaming, and utility was thelast. There wasn't too much of a gap between mobile and business applications.

b. What are the factors that affect software sustainability?

To answer the second question, the dataset needs to be further filtered. Thus, only two applicationdomains have been chosen for the sustainability factor test: mobile and business application domains. Learning repos do not have all the attributes of Github, there aren't too many pull requests or issues because of the nature of such applications. It is for the user to read and learn, not to interact, develop, or enhance, as there is no software or real code there. Hence, it will fail when matching it to the reference model. Also, this repo is highly affected by the pandemic of COVID-19; only the star and fork attributes change to higher numbers; other attributes stay mostly the same.

After selecting two application domains for the sustainability test, the dataset is reduced to 224 repos. This dataset will be tested again in 2022 to examine its activity and popularity; 200 repos survive this test.The factors that affect business and mobile application sustainability are watchers, pull requests, stars, issues, and releases. Those factors influence the commits. A commit is a record that saves changes; any edit in repos will show up in commits.

To better interpret the results, the theory of planned behaviors was used (NORRIS F. KRUEGER, 1993). Users in business apps watch the software, then decide how to tailor the applications to their needs. After that, they share the changes and send a pull request. For software to be sustainable, those pull requests need to be resolved. Hence, for business applications, more people seeing the apps means more people will download them, and as a result, the repos will still be active (commits will increase).

Thus, OSS developers have to care about who watches their apps and try to increase watchers (stars are not that important as people observe the applications and then decide upon them; not all users have to ratethe apps using stars). This is why watchers are considered more important in business applications than stars or forks.

On the other hand, stars play a central role in the mobile application domain, followed by issues, and resolving such issues leads to a new release. In mobile apps, users report more issues, which encourages the production of new releases. And that's true, as many mobile apps are dynamic and fast-growing;every few months, new releases of devices occur in the market. Mobile users download apps that have highly rated stars, more releases mean the apps have evolved to comply with platform requirements or other reported issues in older releases.

CONCLUSIONS

In this study, GitHub was used as the main source of the dataset, GitHub is the largest and most popular OSS platform hosts many OSS repositories [30]. The most popular OSS repos. have been chosen as the dataset. The study took a long time (about three years to finish, starting at the beginning of 2019 and ending in 2022). Each year, the datasets (used for sustainability factor calculation) are examined, to confirm that the repos are still active and popular.

The aim of this study was the identification of the internal factors that affect OSS sustainability. The study of related works reveals the influence of application domain in the OSS lifecycle (Andrea Capiluppi, 2019; Fronchetti et al., 2019; Jansen, 2014), thus the two research questions this studyattempts to answer were:

- What is the most popular application domain in the software market?
- What are the factors that affect software sustainability?

Related to the investigation of the most popular application domain in OSS, learning repos was the first application domain mostly available and demanded, and mobile apps, considered a dynamic type of business application, came in second, followed by business, then gaming, and utility. Application domains are identified using a classifier.

The learning application domain was eliminated and no further processed for sustainability as it didn't have all the required internal factors. During the COVID-19 pandemic, learning repos grew rapidly, new repos became popular rapidly, and some old repos stayed in competition.

To test the sustainability of the repos, only 200 business and mobile software products were considered using statistical analysis, and multiple regression resulted in two tables (3 and 4).

From Table 3, the strongest path, which represents the major factor that leads to increased commits, means that the repos still active (in other words, they affect the sustainability) of mobile app software are stars, issues, and releases. Watchers are the key components to focus on.

Sustainable mobile apps required working on new releases and quickly resolving issues. But for business applications, only two factors are considered important (watchers and pulls), which developers need to keep an eye on.

The learning application domain was eliminated and no further processed for sustainability as it didn't have all the required internal factors. During the COVID-19 pandemic, learning repos grew rapidly, new repos became popular rapidly, and some old repos stayed in competition.

To test the sustainability of the repos, only 200 business and mobile software products were considered using statistical analysis, and multiple regression resulted in two tables (3 and 4).

From Table 3, the strongest path, which represents the major factor that leads to increased commits,means that the repos still active (in other words, they affect the sustainability) of mobile app software are stars, issues, and releases. Watchers are the key components to focus on.

Sustainable mobile apps required working on new releases and quickly resolving issues. But for business applications, only two factors are considered important (watchers and pulls), which developers need to keep an eye on.

`Acknowledgments

The authors would like to thank the Mustansiriyah University, College of Sciences, Computer Sciences

Department for all the support and encouragement that helped to accomplish this study.

REFERENCES

[1] Ahmed Majid Taha, A. A. A., Atheer Akram Abdulrazzaq, A.s. Albahri. (2018). Identifying the Affecting Factors for Adoption of Open Source Software in IT Community. Journal of Engineering and Applied Sciences, 13(14), 5771-5780. doi: 10.3923/jeasci.2018.5771.5780

[2] Alshomali, M. A. A. (2018). Open source software GitHub ecosystem: a SEM approach. (PhD), James Cook University, the College of Business, Law & Governance.

[3] Andrea Capiluppi, N. A. (2019). The Relevance of Application Domains in Empirical Findings. oHeal, May 2019, Montreal, Canada. doi: 10.1145/nnnnnnn.nnnnnn

[4] Bahamdain, S. S. (2015). Open Source Software (OSS) Quality Assurance: A Survey Paper. Procedia Computer Science, 56, 459-464. doi:10.1016/j.procs.2015.07.236

[5] Bahamdain, S. S., Open Source Software (OSS) Quality Assurance: A Survey Paper. International Workshop on Enterprise Web Application Dependability , 56 p. 460, (2015).

[6] Baharuddin, M. F., Izhar, T. A. T., & Shoid, M. S. M. (2018). Adoption of Open Source Software (OSS) and Organization Performance in the Library. International Journal of Academic Research in Business and Social Sciences, 8(9). doi:10.6007/IJARBSS/v8-i9/4591

[7] Bai, S., Liu, L., Meng, C., & Liu, H. (2023). Automating discussion structure re-organization for GitHub issues.

[8] Expert Systems with Applications, 225, 120024. doi:10.1016/j.eswa.2023.120024

[9] Bird, C., Gourley, A., Devanbu, P., Swaminathan, A., & Hsu, G. (2007). Open Borders? Immigration in Open Source Projects. 6-6. doi:10.1109/msr.2007.23

[10] Chen, F., Li, L., Jiang, J., & Zhang, L. (2014). Predicting the number of forks for open source software project. Paper presented at the Proceedings of the 2014 3rd International Workshop on Evidential Assessment of Software Technologies, Nanjing, China. https://doi.org/10.1145/2627508.2627515

[11] Coelho, J., & Valente, M. T. (2017). Why modern open source projects fail. 186-196. doi:10.1145/3106237.3106246 Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub. CSCW '12: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, 1277-1286. doi:10.1145/2145204.2145396

[12] Kinikar, H. A. ., Patil, A. K.-., Thipse, S. S. ., & Jadhav, T. A. . (2023). PCCI-DI Combustion Simulation for Significant Reduction of NOX and PM for GENSET Engine. International Journal of Membrane Science and Technology, 10(2), 306-316. https://doi.org/10.15379/ijmst.v10i2.1200

[13] Farias, V., Wiese, I., & Santos, R. (2019). What Characterizes an Influencer in Software Ecosystems? IEEE Software, 36(1), 42-47. doi:10.1109/ms.2018.2874325

[14] Franco-Bedoya, O. (2015). Open Source Software Ecosystems: Towards a Modelling Framework. 451, 171-179. doi:10.1007/978-3-319-17837-0_16

[15] Fronchetti, F., Wiese, I., Pinto, G., & Steinmacher, I. (2019). What Attracts Newcomers to Onboard on OSS Projects? TL;DR: Popularity. 556, 91-103. doi:10.1007/978-3-030-20883-7_9

[16] Jansen, S. (2014). Measuring the health of open source software ecosystems: Beyond the scope of project health. Information and Software Technology,, 56(11), 1508-1519.

[17] Jiang, J., Lo, D., He, J., Xia, X., Kochhar, P. S., & Zhang, L. (2016). Why and how developers fork what from whom in GitHub. Empirical Software Engineering, 22(1), 547-578. doi:10.1007/s10664-016-9436-6

[18] Johannes Wachs, M. N., William Schueller, Axel Polleres. (2022). The Geography of Open Source Software: Evidence from GitHub. Technological Forecasting & Social Change, 176(121478).

[19] Khalil, W., Haider, S., Al-Shamayleh, A. S., Akhunzada, A., & Gani, A. (2023). Risk Factors and practices for the Development of Open Source Software from Developers' Perspective. IEEE Access, 1-1. doi:10.1109/access.2023.3267048

[20] Konstantions Manikas, K. M. H. (2013). Software ecosystems – A systematic literature review. Journal of System and Software, 86(5), 1294-1306.

[21] Mens, T., Claes, M., & Grosjean, P. (2014). ECOS: Ecological studies of open source software ecosystems. 403-406. doi:10.1109/csmr-wcre.2014.6747205

[22] Mohammad Azeez Alshomali, J. R. H., Jason Holdsworth, SingWhat Tee. (2017). GitHub: Factors influencing project activity levels Proceedings of The 17th International Conference on Electronic Business, ICEB, Dubai, UAE,, 116-124.

[23] Kaewsaeng-on R, AL-Takhayneh SK, Jam FA, Chang B-L, Pradana M and Mahmood S (2022) A three wave longitudinal study of school innovation climate and entrepreneurship teachers' acceptance to technology: Moderating role of knowledge sharing and knowledge hiding. Front. Psychol. 13:1028219. doi: 10.3389/fpsyg.2022.1028219

[24] Munaiah, N., Kroh, S., Cabrey, C., & Nagappan, M. (2017). Curating GitHub for engineered software projects. Empirical Software Engineering, 22(6), 3219-3253. doi:10.1007/s10664-017-9512-6

[25] NORRIS F. KRUEGER, J., ALAN L. CARSRUD2. (1993). Entrepreneurial intentions: - Applying the theory of planned behaviour ENTREPRENEURSHIP h RPGIONAL DEVELOPMENT, 5, 315-330.

[26] Palvia, V. M. a. P. (2012). Factors Affecting the Success of Open Source Software. Journal of Systems and Software., 85(4).

[27] Peterson, K. (2014a). The GitHub Open Source Development Process. Peterson, K. (2014b). The GitHub Open Source Development Process. Pipinellis, A. (2018). GitHub Essentials - Second Edition (Second ed.): Packt.

[28] Prana, G. A. A., Treude, C., Thung, F., Atapattu, T., & Lo, D. (2018). Categorizing the Content of GitHub README Files. Empirical Software Engineering, 24(3), 1296-1327. doi:10.1007/s10664-018-9660-3

[29] Rouder, J. N. (2016). The what, why, and how of born-open data. Behav Res Methods, 48(3), 1062-1069. doi:10.3758/s13428-015-0630-z

[30] Song, J., & Kim, C. (2018). What Is Needed for the Sustainable Success of OSS Projects: Efficiency Analysis of Commit Production Process via Git. Sustainability, 10(9), 3001. doi:10.3390/su10093001

[31] Valiev, M. (2021). External Factors in Sustainability of Open Source Software. (PhD), Carnegie Mellon University Pittsburgh, PA 15213.

[32] Wang, W., Poo-Caamano, G., Wilde, E., & German, D. M. (2015). What Is the Gist? Understanding the Use of Public Gists on GitHub. 314-323. doi:10.1109/msr.2015.36

[33] Xanthopoulos, S., & Xinogalos, S. (2013). A comparative analysis of cross-platform development approaches for mobile applications. 213-220. doi:10.1145/2490257.2490292

DOI: https://doi.org/10.15379/ijmst.v10i4.1859