A Versatile Dataset of Agile Open Source Software Projects

Vali Tawosi, Afnan Al-Subaihin, Rebecca Moussa, Federica Sarro
{vali.tawosi,a.alsubaihin,rebecca.moussa.18,f.sarro}@ucl.ac.uk
University College London
London, UK

ABSTRACT

Agile software development is nowadays a widely adopted practise in both open-source and industrial software projects. Agile teams typically heavily rely on issue management tools to document new issues and keep track of outstanding ones, in addition to storing their technical details, effort estimates, assignment to developers, and more. Previous work utilised the historical information stored in issue management systems for various purposes; however, when researchers make their empirical data public, it is usually relevant solely to the study's objective. In this paper, we present a more holistic and versatile dataset containing a wealth of information on more than half a million issues from 44 open-source Agile software, making it well-suited to several research avenues, and cross-analyses therein, including effort estimation, issue prioritization, issue assignment and many more. We make this data publicly available on GitHub to facilitate ease of use, maintenance, and extensibility.

CCS CONCEPTS

• **Software and its engineering**;

KEYWORDS

Agile Development, Open-Source Software, Data Mining

ACM Reference Format:
Vali Tawosi, Afnan Al-Subaihin, Rebecca Moussa, Federica Sarro. 2022. A Versatile Dataset of Agile Open Source Software Projects. In *Proceedings of MSR '22: Proceedings of the 19th International Conference on Mining Software Repositories (MSR 2022).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The early 2000s has witnessed a surge of the adoption of Agile Software Development alongside the release of the *Agile Software Development Manifesto* in 2001 [12]. Agile techniques boast a faster response to unanticipated alterations that can arise during development such as changes in user requirements, development environments and delivery deadlines; typically contrasted with traditional 'plan-based' project development, which operates under the assumption that software is specifiable and predictable [11]. Agile Software Development is currently among the most common software development methods in project management [24].

MSR 2022, May 23–24, 2022, Pittsburgh, PA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Managing agile software development is commonly aided by an issue tracking tool, which allows agile teams to log and organize outstanding development tasks (e.g., bug fixes, functional and non-functional enhancements), in addition to hosting meta-data related to these tasks. Issue tracking tools, such as Jira [1], provide a trove of historical information regarding project evolution that promise great value for Empirical Software Engineering research. Such data has been employed to address many software engineering problems such as effort estimation [9, 28], task prioritization [13, 15, 31], task assignment [20], task description enhancement [7], iteration planing [8] and exploring social and human aspects [21, 22, 32, 33]. However, the data made available by previous empirical studies is usually mainly relevant solely to the study's objective. Therefore, we aim at paving the way for a more holistic and versatile dataset containing a wealth of information on open-source software projects, which can serve as a single source for many possible research avenues, and enable novel investigations on the inter-play of multiple factors as well as draw observations across multiple research studies.

We call this dataset the TAWOS (**T**awosi **A**gile **W**eb-based **O**pen-**S**ource) dataset. It encompasses data from 13 different repositories and 44 projects, with 508,963 issues contributed by 208,811 users. The dataset is publicly hosted on GitHub [29] as a relational database, and designed such that it is amenable to future expansions by the community. Prospective contributors are welcome to join our effort to maintain, grow and further enhance the database by issuing a pull request on Github.

2 DATASET DESCRIPTION

2.1 Data Extraction

This dataset was mined during the latter half of October 2020. The mining process targeted 13 major open source repositories: *Apache, Appcelerator, Atlassian, DNNSoftware, Hyperledger, Lsstcorp, Lyrasis DuraSpace, MongoDB, Moodle, MuleSoft, Spring, Sonatype, and Talendforge.* Most of these repositories were employed by previous work and they all used Jira as an issue management platform, which ensures uniformity of structure and availability of information. From each of these repositories, projects were selected such that they adopt iterative development and record story points for their issues, thus suggesting that they follow an Agile methodology. We considered projects that have at least 200 issues with recorded story point entries, in order to have enough data to enable statistical analyses resulting in meaningful conclusions.

A total of 904 projects from the aforementioned repositories were considered, among which we selected the 44 that satisfy the collection constraints. To extract issue information, we used the Jira REST Java Client (JRJC) [3]; JRJC was used alongside our own tool, implemented in Java, to extract further features that are not implemented in JRJC (see Section 2.5).

2.2 Data Storage

The final dataset is modeled and stored as a relational database. This enables users of the dataset to employ SQL for easy horizontal and vertical data sampling in addition to allowing easier future expansion. We elected to host the dataset in the MySQL Database Management System as it is lightweight and ubiquitous. The database can be downloaded from a GitHub repository together with the instructions on how to install and use it [29].

2.3 Data Characteristics

The TAWOS dataset contains 508,963 issues from 44 project. The projects are diverse in terms of different project characteristics. Each project contains issues that range from 313 to 66,741 issues. The projects span different programming languages, different application domains and different team geographical locations. Table 1 shows the number of various elements for each of the projects contained in the dataset currently. Those include the number of: all issues, issues categorised as bug report, distinct users (i.e. bug report contributors, etc.), developers, change logs and comments, links to other issues, components, sprints, versions, and the number of issues with story points assigned.

2.4 Data Structure

Figure 1 shows the Entity-Relationship Diagram of the database. The core entity is the `Issue` table, which holds the main information about an issue report. Some of its fields are directly extracted from the issue report such as the issue type (e.g., story, bug, improvement), status (e.g., open, in progress, closed) , description, etc., whereas others are derived from the information stored and/or the events that occurred during the issue's lifecycle. We elaborate on these derived fields in Section 2.5.

Other important tables are `Comment` and `Change_Log` tables. Comments hold the documented discussions of the team around the issue development. Change logs hold all the changes made by the users on the issue report, by recording the field that received the change, the previous value, the next value and the nature of the change. Both these tables store the chronological order of the events in the `Creation_Date` field. Information about the Sprints, Versions, and Components of the issues are also stored in separate tables. The `Issue_Links` table captures the links between the issues. The `User` table stores all the distinct users who interacted with each project, in addition to linking the events and information to their authors and user roles. Any personally identifiable information of users like their usernames and emails are redacted from this dataset.

2.5 Computed and Derived Fields

To further enrich the dataset, we have augmented the mined data with several additional features that are computed or derived from the source Jira repositories as described below.

Issue Description Text and Code. The `Description` field holds the long description of the user story or bug report which can contain natural text interleaved with code snippets or stack-traces. To facilitate processing, we separate the code snippets/stack traces and the natural text describing the issue into the `Description_Code` and `Description_Text` fields respectively. We maintain the original description in the `Description` field. Same is done for the `Comment` field, from which we extract the `Comment_Code` and `Comment_Text`. This is motivated by previous work showing that code tokens may have different meaning from those found in natural language text, hence ought to be analysed separately [23, 26, 28].

Resolution Time. The field `Resolution_Time_Minutes` stores the time span (in minutes) between when an issue is marked for the first time as "*In Progress*" and when it is marked as "*Resolved*". This period can be considered as an approximation of the time taken by the development team to resolve the issue. This is usually the target variable used for bug resolution/fixing time estimation [14, 18, 27]. Other proxies for time are provided, such as `In_Progress_Time` and `Total_Effort_Time`, indicating, respectively, the implementation time and the development (including code review and testing) time.

SP Estimation Date: This field records the time when the `Story_Point` field of the Jira issue report was populated by the developer. This information might be useful, for example, for studies on software effort estimation, in order to properly take into account the chronological order of the estimates and avoid unrealistic usage of the data as described in previous studies [6, 17, 25].

Date and Time. The date and time stored in different Jira repositories may have different timezones, as the projects usually have contributors from all around the World. Therefore, we converted and stored all dates and times to a unified timezone, namely the Coordinated Universal Time (UTC).

Field Change Flag. It is important to keep track of the changes developers made to some of the issue fields. For example, the title and description of the issue are two important pieces of information used by recent automated approaches to produce effort estimates [9], therefore it is important to know whether these fields have been edited after the initial estimate was done. The `Title_Changed_After_Estimation` and `Description_Changed_After_Estimation` fields store this flag. We also provide a flag that shows whether the SP has been changed after the initial estimate. Note that these flags are based on the change logs of the issue.

Change Type in Change Log. This field is calculated to categorise change log updates into one of five categories: "STATUS" indicates a change from one status to another in the Jira workflow of a given issue; "DESCRIPTION" indicates a change to the issue title or description; "PEOPLE" indicates that the user (Change_Log.Field='assignee' or/and 'reporter') of the issue was changed; "STORY_POINT" indicates that the Story Point field of the issue was updated. Any other changes were categorised as "OTHER".

2.6 Extensibility and Maintainability

The TAWOS database is designed such that it is easily extensible by attaching additional information to the corpus. This can help facilitate studying different problems and/or aspects of the same problem. Sharing and managing the dataset as a GitHub repository, enables us to update, expand and enrich its content, whether by us or by the community as external contributions (i.e., pull requests). Github also guarantees that the information can be safely stored long-term, thus preventing the issues often faced in previous work where the data provided are not reachable anymore (e.g., due to use of volatile storing platform such as institutional webpages which change when researchers move to another institution).

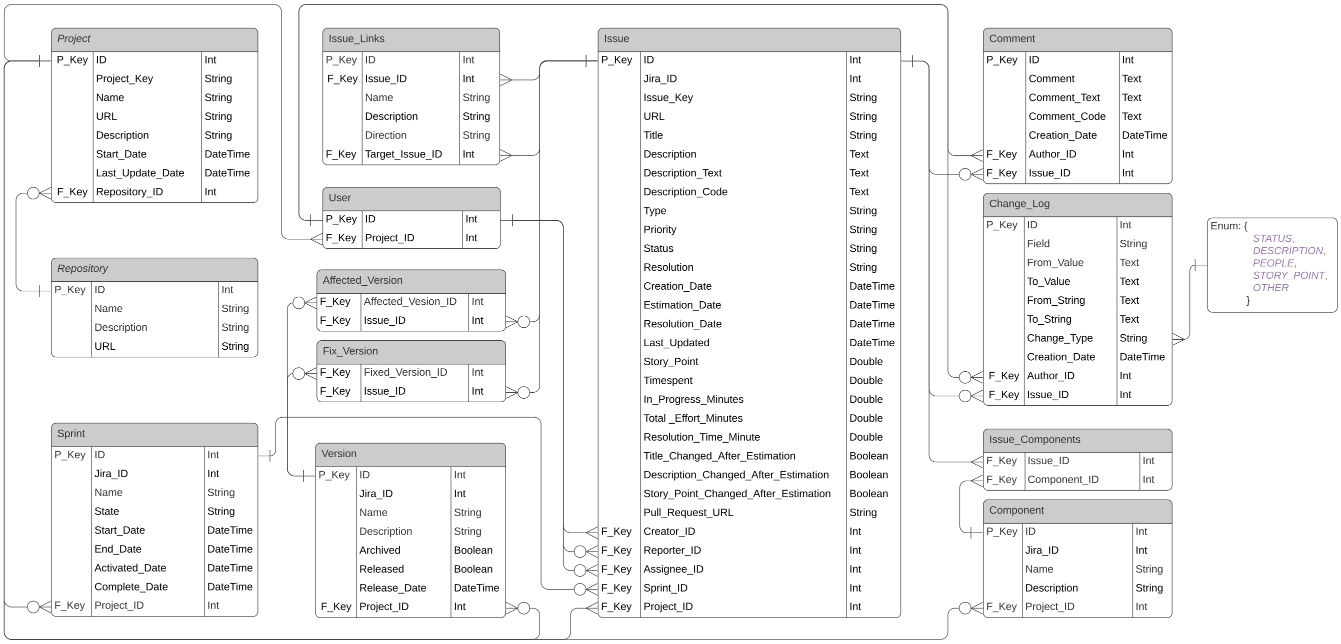

Figure 1: Entity-Relationship Diagram (ERD) for the TAWOS Issues Database.

3 ORIGINALITY AND RELEVANCE

Previous studies have extracted information from issue reports managed in Jira to build predictive models for Story Point (SP) estimation in agile software projects [9, 23, 26], however not all of them have made their data public [23, 26]. Choetkiertikul et al. [9] shared their data in a replication package [2], however, it only consists of features considered in their study (i.e., the issue key, title, description, and story point of the mined issues).

The dataset presented herein encompasses all the projects considered in previous studies[1] [9, 23, 26] augmented with more issues and features.[2] Furthermore, it includes 28 additional projects, which have never been used by any of these previous studies.

Our dataset has been recently used by Tawosi et al. [30] who analysed a total of 31,960 issues from 26 projects stored in TAWOS in order to replicate and extend the work by Choetkiertikul et al. [9]. This set of issues has also been used in a recent study on the effectiveness of clustering for SP estimation [28].

We believe that the TAWOS dataset can help expedite the research in the area of Agile software development effort estimation. In addition to providing a unified benchmark for such studies, it also helps circumvent the challenges faced, and the time consumed, when mining such data from the web. For example, we note that Choetkiertikul et al. [9] could not mine the same data used in the study by Porru et al. [23] likely because the repositories mined had changed during the time period between the two studies.

[1]The only exception is the MuleStudio project used by Choetkiertikul et al. [9], for which we could not find the data source on-line.

[2]The TAWOS dataset has 485,650 more issues in total, and 46,411 more issues with Story Points compared to the one shared by Choetkiertikul et al. [9]. It also contains more issues for each of the 16 projects included in Choetkiertikul et al. [9]'s dataset.

The use of different data in similar studies hinders the immensely useful opportunity to draw observations from across different studies performed at different times around a certain subject matter. We hope that our dataset can help the community tackle this challenge. Although our dataset has been primarily designed to aid in software engineering estimation tasks, it also includes information relevant to other software engineering research, and it is designed to be expanded by other contributors. This allows and promotes the investigation of a wider range of SE aspects as discussed in the next section.

4 RESEARCH OPPORTUNITIES

In addition to benefiting effort estimation studies, the TAWOS dataset promises value to many other areas of software engineering research, including developer productivity studies, iteration planing and task scheduling.

An important research topic in Requirement Engineering is **requirement prioritization** [4, 16, 31] and, especially in an agile setting, the selection of issues for the next iteration [10, 19]. The TAWOS dataset can support such studies by providing a large collection of issues, with known priorities and iterations (i.e., Sprints and Releases) coupled with various aspects providing a full-picture view of the issues, projects and assignees. Additionally, as the dataset makes historical project evolution from multiple repositories available, it enables cross-project analysis.

The TAWOS database provides information about the versioning of the software under development. This information includes the name, description, and release date of the version, and whether it is archived or released. Versions connect to issues via two relations: Affected versions, and Fix versions. The former is the version where a bug or problem was found; whereas the latter is the version where

Table 1: Descriptive statistics of the TAWOS dataset.

Repository	Project Name	Project Key	# Issues	# Bugs	# Users	# Developers	# Change Log	# Comments	# Links	# Components	# Sprints	# Versions	# Story Points
Atlassian	Crowd	CWD	4,311	1,841	2,663	105	62,408	7,440	2,624	50	44	227	214
	Confluence Cloud	CONFCLOUD	23,409	10,071	24,064	513	321,439	64,655	7,694	147	477	17	352
	Software Cloud	JSWCLOUD	11,702	3,505	15,187	211	201,512	30,143	4,492	33	74	68	318
	Jira Cloud	JRACLOUD	25,669	8,339	30,020	557	295,951	74,473	8,176	66	59	170	361
	Confluence Server	CONFSERVER	42,324	25,477	30,755	422	1,608,633	125,591	23,401	104	565	1,121	662
	Atlassian Software Server	JSWSERVER	12,862	6,007	15,468	182	304,682	35,400	5,724	44	70	433	351
	Jira Server	JRASERVER	44,165	20,630	36,585	462	1,162,959	130,457	22,020	115	50	598	380
	Bamboo	BAM	14,252	6,050	7,092	107	256,321	28,638	6,330	115	14	391	528
	Clover	CLOV	1,501	531	347	20	25,812	2,259	338	15	48	63	387
	FishEye	FE	5,533	2,896	2,371	74	112,723	8,914	2,044	9	109	245	240
Apache	Mesos	MESOS	10,157	4,891	1,282	252	108,349	30,152	6,342	42	227	87	3,272
	MXNet	MXNET	1,404	373	156	50	49,295	384	90	9	41	0	209
	Usergrid	USERGRID	1,339	349	97	37	15,435	1,535	270	15	38	8	487
Appcelerator	Command-Line Interface	CLI	645	399	165	29	10,956	2,233	188	12	98	145	374
	Titanium Mobile Platform	TIDOC	3,059	1,344	421	62	81,454	7,712	710	6	217	261	1,297
	Aptana Studio	APSTUD	8,135	6,152	3,365	15	107,961	19,138	1,606	49	12	91	890
	Appcelerator Studio	TISTUD	5,979	3,455	654	63	147,215	19,880	4,051	56	163	126	3,406
	The Titanium SDK	TIMOB	22,059	15,742	3,170	161	483,361	83,252	11,120	52	301	568	4,665
	Appcelerator Daemon	DAEMON	313	123	36	5	4,062	469	90	44	62	20	242
	Alloy Framework	ALOY	1,519	646	386	30	36,312	4,491	586	15	118	172	315
DNN Tracker	DotNetNuke Platform	DNN	10,060	7,319	1,092	33	197,067	32,015	3,766	143	NA	70	2,594
Hyperledger	Blockchain Explorer	BE	802	164	149	64	8,621	1,634	300	0	47	0	373
	Fabric	FAB	13,682	3,562	1,283	457	151,811	23,056	5,312	26	142	55	636
	Indy Node	INDY	2,321	826	133	59	40,111	5,884	1,626	6	76	26	681
	Sawtooth	STL	1,663	318	174	56	15,800	576	454	29	22	4	966
	Indy SDK	IS	1,531	396	177	92	21,842	2,971	602	10	75	30	720
Lsstcorp	Lsstcorp Data management	DM	26,506	2,551	277	211	310,891	71,744	19,722	259	396	4	20,664
Lyrasis	Lyrasis Dura Cloud	DURACLOUD	1,125	374	32	12	11,559	1,443	264	14	7	86	666
MongoDB	Compass	COMPASS	1,791	737	484	17	23,617	2,077	820	87	91	77	499
	Java driver	JAVA	3,560	1,028	1,439	35	42,995	11,018	772	35	46	107	238
	C++ driver	CXX	2,032	502	409	39	30,193	4,756	838	13	56	70	224
	MongoDB Core Server	SERVER	48,663	22,342	8,837	452	1,030,545	136,823	40,084	37	NA	444	784
	Evergreen	EVG	10,299	2,636	300	67	204,228	16,939	2,866	6	NA	26	5,402
Moodle	Moodle	MDL	66,741	41,355	12,230	554	1,298,195	481,606	52,356	97	151	373	1,594
Mulesoft	Mule	MULE	11,816	5,421	1,449	146	233,760	16,627	3,622	129	311	274	4,170
	Mule APIkit	APIKIT	886	467	123	34	16,137	744	154	19	124	96	473
Sonatype	Nexus	NEXUS	9,912	5,975	2,896	82	168,909	26,159	3,956	91	143	167	1,845
Spring	DataCass	DATACASS	798	166	205	10	7,070	919	226	11	54	154	243
	XD	XD	3,707	610	189	31	43,227	4,120	940	18	66	37	3,705
Talendforge	Talend Data Quality	TDQ	15,315	6,288	708	131	249,243	33,438	8,590	88	144	245	1,843
	Talend Data Preparation	TDP	5,670	2,180	320	48	107,565	6,187	3,388	10	79	68	813
	Talend Data Management	TMDM	9,137	6,374	478	110	173,623	31,071	5,438	31	76	141	297
	Talend Big Data	TBD	4,624	2,731	553	98	70,596	5,447	1,348	35	46	149	344
	Talend Enterprise Service Bus	TESB	15,985	4,451	590	118	169,426	17,929	2,228	40	90	371	1,000
		Total	508,963	237,594	208,811	6,313	10,023,871	1,612,399	267,568	2,232	5,029	7,885	69,724

a feature is released or a bug is fixed. This information can be used to track the bug's lifecycle and possibly if the link to the pull request which resolves the bug is presented in the `Pull_Request_URL` field, it can be tracked to the code. This information opens up avenues of research in **software testing and maintenance**.

The TAWOS dataset also contains information on the developer assigned to a given issue, in addition to various information regarding resolution time and the assignee's statistics. Such data enables, for example, the use of machine learning models to help automatically recommend the best developer for a new issue. Additionally, the dataset provides other useful information that can be considered for optimising **task assignment**, for example, considering developers' work load [5]. The dataset also provides the issue status transitions, which can be used to analyse activities and events to predict the **time to fix a bug, or bug triage** [14, 18, 27].

5 FINAL REMARKS

We have indicated just some of the research avenues the TAWOS dataset could be exploited for. We envision that the wealth of information provided, coupled with the ability for other researchers to participate in the growth of the dataset, will enable novel research endeavours on the inter-play among several and different aspects of open-source agile software projects. For example, if a researcher uses our dataset to analyse the corpus of issue comments with regard to developers affects (e.g., emotions, sentiments, politeness), they can extend the dataset by issuing a pull request and thereby augmenting the existing data with the results of their investigation (e.g., augment the comments written by developers with emotions such as surprise, anger, sadness and fear). This data can be re-used in subsequent research investigating the inter-play between, for example, developer emotions and productivity.

We invite potential users of the database to consult our on-line documentation [29] before use in order to understand possible limitations and select data that best fits the aim of their investigations. We plan to curate and expand our dataset by adding other projects and features, and encourage the research community to join our effort in growing and enriching it, in order to open the door for novel research avenues.

ACKNOWLEDGMENTS

Vali Tawosi, Rebecca Moussa and Federica Sarro are supported by the ERC grant no. 741278.

REFERENCES

[1] [n.d.]. Jira Issue & Project Tracking Software | Atlassian. https://www.atlassian.com/software/jira

[2] [n.d.]. Source Code and data for "A Deep Learning Model for Estimating Story Points" · GitHub. https://github.com/SEAnalytics/datasets/tree/master/storypoint/IEEETSE2018

[3] [n.d.]. The Jira REST Java Client, Version 5.2.0. https://mvnrepository.com/artifact/com.atlassian.jira/jira-rest-java-client-app/5.2.0

[4] Rami Hasan Al-Ta'ani and Rozilawati Razali. 2016. A framework for requirements prioritisation process in an agile software development environment: empirical study. *International Journal on Advanced Science, Engineering and Information Technology* 6, 6 (2016), 846–856.

[5] Wisam Haitham Abbood Al-Zubaidi, Patanamon Thongtanunam, Hoa Khanh Dam, Chakkrit Tantithamthavorn, and Aditya Ghose. 2020. Workload-aware reviewer recommendation using a multi-objective search-based approach. In *Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering*. 21–30.

[6] Abdul Ali Bangash, Hareem Sahar, Abram Hindle, and Karim Ali. 2020. On the time-based conclusion stability of cross-project defect prediction models. *Empirical Software Engineering* 25, 6 (2020), 5047–5083.

[7] Oscar Chaparro, Jing Lu, Fiorella Zampetti, Laura Moreno, Massimiliano Di Penta, Andrian Marcus, Gabriele Bavota, and Vincent Ng. 2017. Detecting missing information in bug descriptions. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. 396–407.

[8] Morakot Choetkiertikul, Hoa Khanh Dam, Truyen Tran, Aditya Ghose, and John Grundy. 2017. Predicting delivery capability in iterative software development. *IEEE Transactions on Software Engineering* 44, 6 (2017), 551–573.

[9] Morakot Choetkiertikul, Hoa Khanh Dam, Truyen Tran, Trang Pham, Aditya Ghose, and Tim Menzies. 2019. A Deep Learning Model for Estimating Story Points. *IEEE TSE* 45, 7 (2019), 637–656. https://doi.org/10.1109/TSE.2018.2792473

[10] Juan J Durillo, Yuanyuan Zhang, Enrique Alba, Mark Harman, and Antonio J Nebro. 2011. A study of the bi-objective next release problem. *Empirical Software Engineering* 16, 1 (2011), 29–60.

[11] Tore Dybå and Torgeir Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50, 9 (2008), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

[12] Martin Fowler, Jim Highsmith, et al. 2001. The agile manifesto. *Software development* 9, 8 (2001), 28–35.

[13] Carlos Gavidia-Calderon, Federica Sarro, Mark Harman, and Earl T. Barr. 2021. The Assessor's Dilemma: Improving Bug Repair via Empirical Game Theory. *IEEE Transactions on Software Engineering* 47, 10 (2021), 2143–2161. https://doi.org/10.1109/TSE.2019.2944608

[14] Mayy Habayeb, Syed Shariyar Murtaza, Andriy Miranskyy, and Ayse Basar Bener. 2017. On the use of hidden markov model to predict the time to fix bugs. *IEEE Transactions on Software Engineering* 44, 12 (2017), 1224–1244.

[15] Yuekai Huang, Junjie Wang, Song Wang, Zhe Liu, Dandan Wang, and Qing Wang. 2021. Characterizing and Predicting Good First Issues. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–12.

[16] Maliheh Izadi, Kiana Akbari, and Abbas Heydarnoori. 2020. Predicting the Objective and Priority of Issue Reports in a Cross project Context. *arXiv preprint arXiv:2012.10951* (2020).

[17] Matthieu Jimenez, Renaud Rwemalika, Mike Papadakis, Federica Sarro, Yves Le Traon, and Mark Harman. 2019. The importance of accounting for real-world labelling when predicting software vulnerabilities. In *Procs. of ESEC/FSE*. 695–705.

[18] Youngseok Lee, Suin Lee, Chan-Gun Lee, Ikjun Yeom, and Honguk Woo. 2020. Continual prediction of bug-fix time using deep learning-based activity stream embedding. *IEEE Access* 8 (2020), 10503–10515.

[19] Lingbo Li, Mark Harman, Emmanuel Letier, and Yuanyuan Zhang. 2014. Robust next release problem: handling uncertainty during optimization. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 1247–1254.

[20] Senthil Mani, Anush Sankaran, and Rahul Aralikatte. 2019. Deeptriage: Exploring the effectiveness of deep learning for bug triaging. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. 171–179.

[21] Marco Ortu, Giuseppe Destefanis, Bram Adams, Alessandro Murgia, Michele Marchesi, and Roberto Tonelli. 2015. The JIRA Repository Dataset: Understanding Social Aspects of Software Development. In *Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering* (Beijing, China) *(PROMISE '15)*. Association for Computing Machinery, New York, NY, USA, Article 1, 4 pages. https://doi.org/10.1145/2810146.2810147

[22] Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. 2016. The emotional side of software developers in JIRA. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. IEEE, 480–483.

[23] Simone Porru, Alessandro Murgia, Serge Demeyer, Michele Marchesi, and Roberto Tonelli. 2016. Estimating story points from issue reports. In *PROMISE*. 1–10.

[24] Project Management Institute. 2017. *Success Rates Rise - 2017 9th Global Project Management Survey*. Technical Report. https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf

[25] Federica Sarro, Rebecca Moussa, Alessio Petrozziello, and Mark Harman. 2020. Learning From Mistakes: Machine Learning Enhanced Human Expert Effort Estimates. *IEEE Transactions on Software Engineering* (2020).

[26] Ezequiel Scott and Dietmar Pfahl. 2018. Using developers' features to estimate story points. In *ICSSP*. 106–110.

[27] Reza Sepahvand, Reza Akbari, and Sattar Hashemi. 2020. Predicting the bug fixing time using word embedding and deep long short term memories. *IET Software* 14, 3 (2020), 203–212.

[28] Vali Tawosi, Afnan Al-Subaihin, and Federica Sarro. 2022. Investigating the Effectiveness of Clustering for Story Point Estimation. In *SANER 2022*. IEEE.

[29] Vali Tawosi, Afnan Alsubaihin, Moussa Rebecca, and Federica Sarro. [n.d.]. The TAWOS dataset. https://github.com/SOLAR-group/TAWOS.git

[30] Vali Tawosi, Rebecca Moussa, and Federica Sarro. 2022. Deep Learning for Agile Effort Estimation Have We Solved the Problem Yet? arXiv:2201.05401 [cs.SE]

[31] Qasim Umer, Hui Liu, and Inam Illahi. 2019. CNN-based automatic prioritization of bug reports. *IEEE Transactions on Reliability* 69, 4 (2019), 1341–1354.

[32] Andric Valdez, Hanna Oktaba, Helena Gómez, and Aurora Vizcaíno. 2020. Sentiment analysis in jira software repositories. In *2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT)*. IEEE, 254–259.

[33] Ting Zhang, Bowen Xu, Ferdian Thung, Stefanus Agus Haryono, David Lo, and Lingxiao Jiang. 2020. Sentiment analysis for software engineering: How far can pre-trained transformer models go?. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 70–80.