

Global Journal of Engineering Education

Evaluation of learning analytics metrics and dashboard in a software engineering project course

Faisal A.T. Putra, Harry B. Santoso & Rizal F. Aji

Universitas Indonesia
Depok, Jawa Barat, Indonesia

ABSTRACT: Learning analytics is a technique to monitor a learning activity using metrics. The aim of this research was to find the impact of data filtering on metric quality, the learning analytics dashboard usability in a software development course project, and to compare a correlation-based dashboard with a randomly arranged dashboard. A quantitative method was applied, with the metric correlation set to lecturer scores; the system usability scale (SUS) was a tool for evaluating dashboards; and the Fisher-Irwin test and *t*-test were applied to compare dashboards. A qualitative method was applied to evaluate dashboard usability, with usability testing through *lost our lease*. A simple data filtering technique can improve metric quality except for code review metrics. As for usability, the learning analytics dashboard has a relatively good and acceptable SUS score of 70.75. Findings from the research reveal a significant difference between correlation-based and randomly arranged dashboards, whereas two other indicators suggest no significant differences are present.

Keywords: Dashboard, metric, learning analytics

INTRODUCTION

Since 2000, the Accreditation Board for Engineering and Technology (ABET) in Indonesia has formulated requirements for graduates in the engineering field. The requirements include the ability to communicate effectively and the ability to use necessary techniques, skills and modern engineering tools in engineering practice [1]. The requirements impact the design of a software development course to incorporate the use of collaborative tools and tasks that promote intensive teamwork. At the professional level, software development is rarely considered as an individual, but a team activity [2]. This is shown by the division of tasks and interactions among software development team members.

As a result, project-based learning (PBL) is normally used in a software development course, because the emphasis is on collaboration that can mimic real situations in the software industry. To promote easy interactions and monitoring activities among students and lecturers, many on-line tools for collaboration are used. This approach is known as computer-supported collaborative learning (CSCL) [3]. Computer-supported collaborative learning already is used by the Computer Science Faculty of Universitas Indonesia, especially in its software engineering project course.

Learning analytics techniques are often used to take advantage of data or activity records recorded through CSCL. Learning analytics is defined as a set of activities that measure, collect, analyse and report data about learners and the learning context, which they aim to know and understand, to optimise learning activities and environments [4]. One of the most important aspects of learning analytics are the metrics presented to the users: teachers or lecturers and students [5]. Metrics area set of functions or standards used to measure or evaluate an object. Metrics in learning analytics can be defined as a set of functions or standards that can be used for measuring or evaluating a student's activity and performance.

In the case of learning analytics in a software development project course, there are so many metrics that can be used, such as commit count, lines of code, code review count, code review comment count, issue count, assigned issue count, time to solve and file ownership [6]. The challenge is to determine which metrics are relevant to the course, and then to improve these metrics. Sometimes, the data recorded by a CSCL system, such as software repository and the code review system, contain undesired noise that decreases the metric quality. Such data need to be filtered to improve the metric quality.

One of the ways to present the metrics in learning analytics is by visualising the process on a learning analytics dashboard. Today, the trendy research topics in learning analytics revolve around how to visualise the metrics on a dashboard. Vozniuk et al are researchers who focus on the learning analytics dashboard [7]. As a result of their research, a portable learning analytics dashboard with widgets was proposed, so users could arrange their dashboards based on personal preferences.

To improve the portable learning analytics dashboard in the project course, the research findings reported here led to a recommended initial arrangement of widget positions on the learning analytics dashboard, which could then be personalised by users. In this context, one widget represents one metric. The authors proposes a portable learning analytics dashboard concept that is based on the correlation of metrics contained in the widgets. The widgets will be arranged based on the metric correlation, with the score from the lecturer as a gold standard. The aim of this research was to find out whether the proposed learning analytics dashboard was better than a randomly arranged learning analytics dashboard, by looking at users' tendency to change widget position and the number of users changing the widget position, as well as the differences between the recommended widget positioning on the dashboard and the desired widget positioning by the users.

LITERATURE REVIEW

Learning Analytics Metrics

Learning analytics is a field of study in technology enhanced learning (TEL) that is still developing. It is a set of activities that measure, analyse and report data about learners, and the context of their learning, to understand and optimise learning experiences and environments [4]. Based on that definition, knowing and understanding learners' condition and their learning environment based on data are at the core of learning analytics. Therefore lecturers, teachers, instructors and students should evaluate the learning activities.

The learning analytics cycle proposed by Clow consists of learners, data, metrics and interventions [5]. Once the data about students are available, they are measured by metrics, so users (students and lecturers or teachers) can interpret the data.

In this software development project course, the data can come from many sources: the version control system or software repository, the issue tracking system, wiki or blog and other sources. Studies about learning analytics metrics have been conducted in at least the past five years [2][6][8-12].

Learning Analytics Dashboard

A learning analytics dashboard is a technique by which to visualise learning analytics. A learning analytics dashboard is intended as a support for educators (teachers and lecturers), so as to have a better overview of learning activities, reflection or evaluation on learning activities, and to identify students who are isolated from the learning or student groups, and hence have a higher risk of failure [13]. In general, a learning analytics dashboard provides various metric visualisations of recorded and incorporated data. For example, Charleer et al implemented a learning analytics dashboard for a human computer interaction (HCI) course [14]. The metrics were shown on students' Web logs gathered using a really simple syndication (RSS) feed and twitter accounts using Twitter API.

There are only two studies indexed by Google Scholar related to learning analytics dashboards in a software development context and the two studies have a different focus. The first study by Charleer et al focuses on the use of badge, activity, and content as the visualised metrics [14]. The other study focuses on how to visualise the collaboration process in software development teams [15].

METHODOLOGY

The data sources for this study were five software projects from the 2016 Software Development Course. The Software Development Project course was selected, because the course used software repository (four groups using Github and one group using Gitlab), an issue tracking system (four groups using Trello and one group using Asana), and a code review system (four groups using Github and one group using Gitlab). In the 2016 class, there were six teaching assistants and two lecturers. The metric evaluation was performed by correlating the metric value with the score given by the lecturers. The lecturers' score was calculated from the documentation of the assessment based on completeness of mandatory tasks, elective tasks, affective scores and individual progress scores.

There are two steps: metric evaluation and dashboard evaluation. The metric evaluation consists of data collection, metric calculation with and without filtering and metric evaluation. The dashboard evaluation consists of analysis and design, dashboard development (for both the correlation-based and randomly arranged dashboards) and dashboard evaluation.

Learning Analytics Metrics Evaluation

The data sources of this study were five software projects from the 2016 Software Development Course class as described above. The evaluation of learning analytics metrics involves four main steps:

- data collection;
- metric calculation;
- data filtering;
- metric evaluation.

Data Collection

The data were collected using the application programming interface (API) of the CSCL system (Github, Gitlab, Trello and Asana). OAuth authentication using the authors' own accounts were required. To be able to access the data source, every team was required to add each author's account into the CSCL project.

Metric Calculation

The data collected were converted to metrics. The metrics used were commit count per student, lines of code per student, file ownership count per student, review request count per student, review comment count per student, issue count per student, assigned issue count per student, issue comment count per student and mean time to solve per student.

Data Filtering

Data filtering was performed on commit and review/issue messages. Filtering on commit messages would affect three metrics (commit count, lines of code and file ownership count). Filtering on issue/review messages would affect the following file metrics: issue count, issue comment count, issue assigned count, review request count and review comment count.

Five filtering methods were performed on commit messages:

1) commit limitation based on commit message character count;
2) commit limitation based on lines of code per commit;
3) commit limitation based on file ownership per commit;
4) simultaneous combination of commit message, lines of code and file ownership;
5) cut-off based on commit message character count/lines of code/file ownership by 5%, to 50% in 5% increments.

Review filtering was performed in two ways: data filtering, such as comment message/issue title minimal character count and manual annotation. Manual annotation was performed with comment annotation from two annotators who were experienced in performing assessment in software development. Annotation results were then compared to find the kappa coefficient score according to the annotations of two annotators. If the kappa coefficient score was higher than or equal to 0.60 or moderate [16], 50% of annotation differences use the first annotator result, whereas the other 50% will use the second annotator result. If the kappa coefficient is less than 0.60, the two annotators will meet and discuss the differences in the results.

Metric Evaluation

The metric evaluation is shown throughout the article. Generally, the steps performed in this metric evaluation were to determine the good and relevant metrics. The evaluation would also identify the irrelevant metrics in the software development project course. The relevant metrics are ones with a correlation score higher than 0.1 (very weak) and have *p*-value lower than or equal to 0.05 (significant).

Dashboard Evaluation

The dashboard evaluation was performed in three steps:

- analysis and design;
- dashboard development;
- dashboard evaluation.

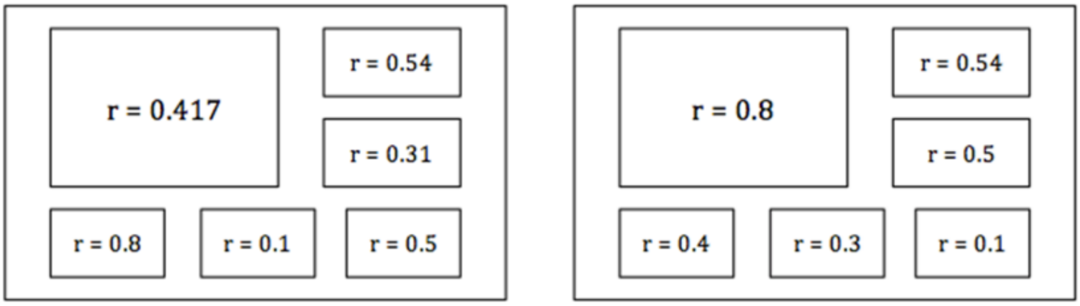
Analysis and Design

In this step, the data collection, metric calculation, and the dashboard mechanism were analysed to develop and design dashboard systems capable of presenting and visualising the metrics. The analysis and design method used the Web information system development methodology (WISDM) with the following steps: organisational analysis, information analysis, human-computer interface, technical design and work design [17].

Dashboard Development

After the dashboard systems were analysed and designed, the learning analytics dashboards were developed. The dashboards run on the current learning management system at the Computer Science Faculty in Universitas

Indonesia, viz. the student-centred e-learning environment (SCELE). Two kinds of dashboards in relation to widget positioning were developed: randomly arranged dashboard and correlation-based dashboard. The differences between the two dashboards are shown in Figure 1.



a) b)

Figure 1: Widget learning analytics dashboards; a) randomly arranged dashboard; b) correlation-based dashboard.

Dashboard Evaluation

After the two dashboards were developed, an evaluation was performed to find out the usability of the learning analytics dashboards and which dashboard was better: the correlation-based dashboard or the randomly arranged dashboard. To determine their usability, both learning analytics dashboards were tested in two ways: using both qualitative and quantitative approaches.

The qualitative approach was performed by using Web usability testing with the *lost our lease* framework proposed by Krug [18]. In usability testing with the *lost our lease* framework, users had to undertake tasks indirectly to run the dashboard. The quantitative approach used the system usability scale (SUS) [19] on four stakeholders: the lecturers, teaching assistants, students and system administrators.

The calculation to find the better dashboard was conducted using A/B testing. The A/B testing or split-half testing is a testing method used to compare two Web designs [20]. The A/B testing was conducted on 24 students involved in the metric analysis and evaluation. The recorded activities were:

1) The number of users who change the widget position.
2) The number of clicks on the button *change widget position*, which indicates the tendency to change the widget position.
3) The differences between the recommended position of widgets and the users' desired position of widgets.

A Fisher-Irwin test was used to evaluate 1 and 2, and a one-tailed *t*-test was used for 3 [20].

RESULTS

Learning Analytics Dashboard Results

Data were calculated using various predetermined metrics. Those metrics are commit count, lines of code, file ownership count, issue count, assigned issue count, review request count, issue comment count, review comment count and processing time. The results of the metric calculation are shown in Table 1.

Table 1: Metric profile.

No.	Metrics	Metric profile						Correlation	
		Mean	Min	Max	SD	n	Sum	*r*	*p*
1	Commit count	20.92	1	104	23.56	24	503	0.5758	0.0032
2	Lines of code	16320.78	1	77032	21811.41	24	391699	0.3636	0.0880
3	File ownership	292	1	1066	291.89	24	7008	0.5336	0.0072
4	Issue count	19.72	1	90	23.66	22	434	0.3644	0.0954
5	Assigned issue	3.94	1	14	3.88	17	67	0.2434	0.3463
6	Review request	3.7	1	8	2.31	10	38	-0.0554	0.8792
7	Issue comment count	14.7	1	44	14.76	15	221	0.5851	0.0172
8	Review comment count	1.33	1	2	0.57	3	4	-0.8660	0.3333
9	Processing time (thousands sec)	1354.80	57.74	3404.13	1105.23	13	17612.44	-0.1043	0.7343

After performing data filtering as explained in the methodology section, the best data filtering methods with the best correlation values are shown in Table 2.

Table 2: Metric value pre- and post-data filtering.

No.	Metrics	Correlation (Spearman)				Improvement	Fit (rank)
		Without data filtering		With data filtering			
		r	*p*	*r*	*p*		
1	Commit count	0.5758	0.0032	0.7609	0.0025	Exist	Yes (2)
2	Lines of code	0.3636	0.0880	0.5175	0.0232	Exist	Yes (5)
3	File ownership	0.5336	0.0072	0.8251	0.0009	Exist	Yes (1)
4	Issue count	0.3644	0.0954	0.5320	0.0230	Exist	Yes (4)
5	Assigned issue count	0.2434	0.3463	Not implemented		-	Yes (6)
6	Review request count	-0.0554	0.8792	-0.8365	0.0189	Exist, but negative	No
7	Issue comment count	0.5851	0.0172	0.7431	0.0023	Exist	Yes (3)
8	Review comment number	-0.8660	0.3333	-0.8660	0.3333	Not exist	No
9	Processing time	-0.1043	0.7343	Not implemented		-	No

In the case of metrics related to the commit or software repository, such as commit count, lines of code and file ownership, the best correlation value was obtained through the data filtering method with data elimination under the conditions as seen in Table 3.

Table 3: Metric value pre- and post-data filtering.

No.	Metrics	Minimal			Correlation	
		Char*	LoC*	FO*	*r*	*p*
1	Commit number	15	30	25	0.7609	0.0025
2	Lines of code	20	20	5	0.5175	0.0232
3	File ownership	25	30	25	0.8251	0.0009

* Char = commit message character, LoC = lines of code, FO = file ownership count

This research also used data filtering with a cut-off technique, as shown in Table 4. This method of filtering did not produce a better correlation score than the data filtering technique with the limitation of commit message character, lines of code and file ownership.

Table 4: Data filtering in software repository related metrics.

No.	Cut-off (%)	Commit number		Lines of code		File ownership	
		R	*p*	*r*	*p*	*r*	*p*
1	0	0.5758	0.0032	0.3636	0.0880	0.5336	0.0072
2	5	0.5758	0.0032	0.3636	0.0880	0.5336	0.0072
3	10	0.5758	0.0032	0.3636	0.0880	0.5336	0.0072
4	15	0.5758	0.0032	0.3636	0.0880	0.5341	0.0072
5	20	0.5758	0.0032	0.3636	0.0880	0.5358	0.0069
6	25	0.5758	0.0032	0.3642	0.0875	0.5358	0.0069
7	30	0.5758	0.0032	0.3541	0.0973	0.5358	0.0069
8	35	0.5758	0.0032	0.3541	0.0973	0.5358	0.0069
9	40	0.5758	0.0032	0.3541	0.0973	0.5358	0.0069
10	45	0.5758	0.0032	0.3541	0.0973	0.5358	0.0069
11	50	0.5772	0.0031	0.3541	0.0973	0.5358	0.0069

In the case of metrics related to the issue tracking system, the best metric value was gained from limiting the minimum number of characters to 30 for issue count and limiting the minimum number of characters of comments to 10 in issue comment count metrics. Results are shown in Table 5 in the Appendix. The usage of annotation made by humans (kappa score = 0.6014) did not result in a better correlation score than that of the character limitation method, as shown in Table 6.

Table 6: Data filtering with human annotation results.

Result	Issue comment		Review comment	
	r	*p*	*r*	*p*
Original	0.5851	0.0172	-0.8660	0.3333
Filtered by humans	0.6858	0.0047	-0.8660	0.3333

Based on the scores above, it is found that a variety of simple filtering techniques, such as character count limitation, file limitation and lines of code limitation can improve metrics related to commit/software repository, and hence improve metric relevance with these learning activities.

Learning Analytics Dashboard Evaluation Results

For this study, the analysis and design of learning analytics dashboards were performed with the WISDM framework. The results of the organisational analysis on the learning analytics dashboard are shown in Table 7. The informational design of dashboards was performed using the unified modelling language (UML).

Table 7: Organisational analysis of learning analytics dashboards.

No.	Stakeholder	Role
1	Lecturers	Creating, activating, and setting the type of assignments, monitoring student activities on assignments of learning analytics dashboards.
2	Teaching assistants	Helping and assisting lecturers in monitoring student activities on assignments using learning analytics dashboards.
3	Students	Performing assignments in accordance with the criteria set by lecturers, monitoring their own individual/team activities using the learning analytics dashboards.
4	Administrators	Installing assignments plugin and making sure that the assignment type is shown and running in the SCELE/LMS system.

After the analysis and design were performed, the learning analytics dashboard was developed under an on-line learning environment at the Computer Science Faculty in Universitas Indonesia, named SCELE. The SCELE runs using the Moodle learning management system (LMS). The results of the development of dashboards are shown in Figure 2.

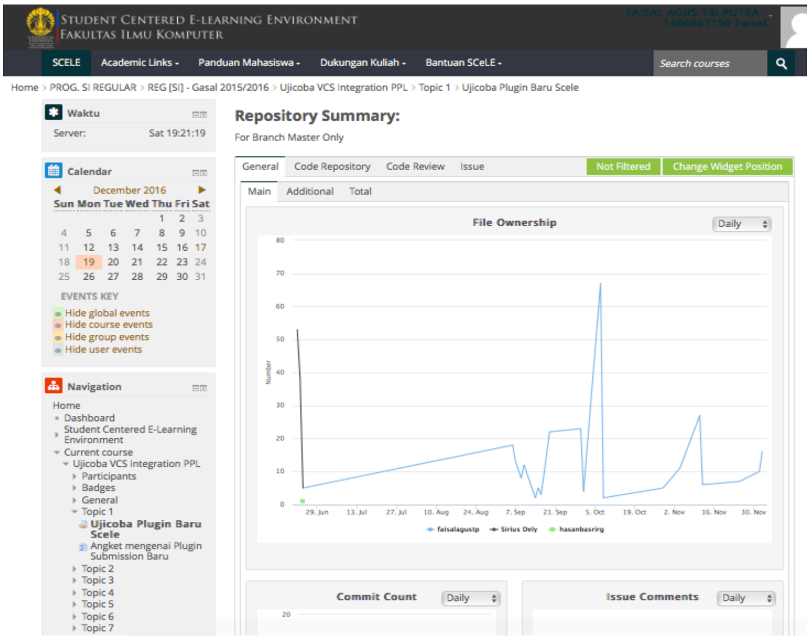


Figure 2: Screenshot of the learning analytics dashboards.

After the learning analytics dashboards were developed, they were quantitatively evaluated in terms of the usability aspect using SUS and were put under a test usability testing with the *lost our lease* framework to find out what the dashboards lack. Moreover, the evaluation of dashboards was conducted by comparing between the correlation-based learning analytics dashboard and the randomly arranged learning analytics dashboard. The SUS scores are shown in Table 8.

Table 8: Learning analytics dashboard SUS scores.

No.	Stakeholder	Mean score for the questions										Total
		1	2	3	4	5	6	7	8	9	10	
1	Lecturers	8.75	7.5	8.75	8.75	6.25	5	8.75	8.75	6.25	8.75	77.5
2	Teaching assistants	7.5	6.25	7.5	5	8.75	6.25	6.25	5	6.25	6.25	65
3	Students	8	6.5	7	7.5	8.5	6.5	6.5	6.5	7	5.5	69.5
4	Administrators	7.5	10	7.5	7.5	5	7.5	7.5	7.5	7.5	7.5	75
Mean		8	7	7.5	7.25	7.75	6.25	7	6.75	6.75	6.5	70.75

Generally, the learning analytics dashboard system is acceptable level with a good score (70.75). However, from the stakeholders' perspective, the highest score of acceptance of 77.5 was given by the lecturers and the lowest score of 65 was given by the teaching assistants. Out of the nine people who were the respondents of usability testing, all qualitatively regarded the system as good and proper to be used in the software engineering project course, with some notes for improvement.

The highlights of the notes obtained from the usability testing are detailed button improvement, pie chart facility, the metric name that was not clear in the tab *total*, the lack of notification before deleting a repository in assignments, the moving button widget that is not intuitive, the accordion that is not intuitive, and the lack of the representative graphic type.

The A/B testing for comparing learning analytics dashboard based on the correlation and randomly arranged view involved 24 students. Out of the 24 students, only 21 provided responses and accessed the learning analytics dashboards. Out of the 21 students, 11 students accessed the learning analytics dashboard based on the metric correlation and 10 students accessed the randomly arranged one. The detail of changes applied by the students is displayed in a 2 x 2 contingency table, as shown in Table 9. This indicates the number of respondents altering the dashboard position.

Table 9: 2 x 2 contingency table on the number of respondents changing the widget position.

2 x 2 contingency table	Change	Not change
Correlation-based	7	3
Randomly arranged	3	8

In addition, the tendency to change the widget position (including users who did not ultimately change the widget position, but clicked the button *change position* is shown in Table 10 below.

Table 10: 2x2 contingency table on the tendency to change the widget position on the learning analytics dashboards.

2 x 2 contingency table	Clicking the button	Not clicking the button
Randomly arranged	7	3
Correlation-based	6	5

The results from the Fisher-Irwin test on both contingency tables are shown in Table 11 below. The results show that there is no significant difference between the number of users who changed the widget position and the number of users with a tendency to change the dashboard position on both correlation-based and randomly arranged learning analytics dashboards.

As for the tendency to change the position, the *p* value is much higher than the significance limit (0.05), so it can be concluded that this bears no significance. As for the number of users who ultimately made changes to the dashboard position, the *p* value is also higher than 0.05, so it can be concluded that there are no significant differences.

Table 11: Dashboard evaluation using the Fisher-Irwin test.

Type	Score	*p*-value
Tendency to change	1.9444	0.6594
Changing the position	6.2222	0.0861

The similarity between the recommended widget dashboard and the widget position chosen by users on their dashboards is shown in Table 12. After performing testing using a one tailed *t*-test, the *p*-value was 0.0139, with a positive *t*-value (2.7098). Therefore, it can be concluded that the correlation-based dashboard is significantly better than the randomly arranged dashboard according to the similarity between the suggested dashboard arrangement and the users' desired arrangement.

Table 12: Widget positioning similarity score on randomly arranged dashboard and correlation-based dashboard.

Correlation-based group		Randomly arranged group	
50	100	100	100
100	66.67	33.33	33.33
100	100	100	16.67
100	66.67	16.67	33.33
66.67	100	66.67	
100		33.33	
Mean = 86.36		Mean = 53.33	

CONCLUSIONS

The findings from this study shows that simple data filtering, such as character limitation, lines of code and file ownership could improve the metric quality, as shown by the increasing correlation score between the metric value and the final score by the lecturers. But, the improvement does not apply for metrics related to the code review system. This is considered to be because of a lack of data for metrics related to the code review system.

Two learning analytics dashboards were developed: a dashboard with widget arrangement based on the metric correlation and a dashboard with random widget arrangement. The dashboards were evaluated based on three indicators: the number of users who change the dashboard, the number of users who have a tendency to change the dashboard, and the similarity between dashboards with suggested widget arrangement and dashboards with the widget position reflecting user desire. For the first two indicators, there is no significant difference between the two dashboards, with p-values equal to 0.65 and 0.08, respectively. Nevertheless, for the third indicator, the dashboard with the correlation-based arrangement is significantly better than the randomly arranged one, with a p-value equal to 0.01 and a positive t value.

Suggestions for further research result from the findings of this study. From the user's point of view regarding the usability, the acceptance level is quite good. But viewed from the standpoint of a point-to-point score, many respondents doubt the integrity, consistency and learnability of the system. One of the solutions is to allow enough time for customers to familiarise themselves with the system before performing the SUS test.

REFERENCES

1. Skvarenina, T., Incorporating and assessing ABET *soft skills* in the technical curriculum. *Proc. ASEE Annual Conf. and Expo.* (2008).
2. Dietsch, D., Podelski, A., Nam, J., Papadopoulos, P.M. and Schäf, M., Monitoring Student Activity in Collaborative Software Development, http://arxiv.org/abs/1305.0787
3. Coccoli, M., Stanganelli, L. and Maresca, P., Computer supported collaborative learning in software engineering. *Proc. IEEE Global Engng. Educ. Conf.*, IEEE, 990-995 (2011).
4. Ferguson, R., Learning analytics: drivers, developments and challenges. *Inter. J. of Technol. Enhanced Learning 4*, Inderscience (2012).
5. Clow, D., The learning analytics cycle. *Proc. Inter. Conf. on Learning Analytics and Knowledge*, ACM Press, New York, 134 (2012).
6. Mittal, M. and Sureka A., Process mining software repositories from student projects in an undergraduate software engineering course. *Proc. 36th Inter. Conf. on Software Engng.*, ICSE Companion, 344353 (2014).
7. Vozniuk, A., Govaerts, S. and Gillet, D., Towards portable learning analytics dashboards. *Proc. 2013 IEEE 13th Inter. Conf. on Advanced Learning Technologies*, IEEE, 412-416 (2013).
8. Santos, J.L., Govaerts, S., Verbert, K. and Duval, E., Goal-oriented visualizations of activity tracking. *Proc. Inter. Conf. on Learning Analytics and Knowledge*, New York: ACM Press, 143 (2012).
9. Robles, G. and Gonzalez-Barahona, J.M., Mining student repositories to gain learning analytics. An experience report. *Proc. IEEE Global Engng. Educ. Conf.*, IEEE, 1249-1254 (2013).
10. Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S. and Koller, D., Programming pluralism: using learning analytics to detect patterns in the learning of computer programming. *J. of the Learning Sciences*, 23, **4**, 561-599 (2004).
11. Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S.H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M.A., Sheard, J., Skupas, B., Spacco, J., Szabo, C. and Toll, D., Educational data mining and learning analytics in programming: literature review and case studies. *The 2015 ITiCSE on Working Group Reports*, Vilnius, Lithuania (2015).
12. Sørhus, S.K., Applying Learning Analytics in the Course TDT4100 at NTNU: NTNU (2015).
13. Verbert, K., Drachsler, H., Manouselis, N., Wolpers, M., Vuorikari, R. and Duval, E., Dataset-driven research for improving recommender systems for learning. *Proc. Inter. Conf. on Learning Analytics and Knowledge*, ACM Press, New York, 44-53 (2011).
14. Charleer, S., Santos, J.L., Klerkx, J. and Duval, E., Improving teacher awareness through activity, badge and content visualizations. *Proc. Inter. Conf. on Web-based Learning*, Springer, 143-152 (2014).
15. Vivian, R., Tarmazdi, H., Falkner, K., Falkner, N. and Szabo, C., The development of a dashboard tool for visualising online teamwork discussions. *Proc. ICSE'15*, IEEE, 380-388 (2015).
16. Viera, A.J. and Garrett, J.M., Understanding interobserver agreement: the kappa statistic. *Family Medicine*, 37, **5**, 360-363 (2005).
17. Vidgen, R., Constructing a Web information system development methodology. *Infor. Systems J.*, 12, **3**, 247-261 (2002).
18. Krug, S., *Don't Make Me Think. A Common Sense Approach to Web Usability*, Indianapolis: Que Corp, (2000).
19. Sauro, J. and Sauro, J., Measuring Usability with the System Usability Scale (2011), 26 August 2018, https://measuringu.com/sus/
20. Sauro, J. and Lewis, J.R., *Quantifying the User Experience: Practical Statistics for User Research*. Elsevier (2012).

Faisal Agus Tri Putra received his Bachelor degree in education in computer science from Indonesia University of Education and a Master of Computer Science from Universitas Indonesia. His research is oriented towards learning analytics, vocational education, gaming and software engineering. At present, he is working as a lead software engineer at Solve Education, a non-profit organisation that focuses on helping children and youth around the globe receive quality, effective education by using mobile game technology and gamification. He is also working as a game and learning analyst at the same organisation.

Dr Harry Budi Santoso is a faculty member at the Faculty of Computer Science in Universitas Indonesia (UI). He received his BS and MS in Computer Science from Universitas Indonesia (UI), and his PhD from the Department of Engineering Education at Utah State University, in the USA. He has taught several courses at UI (e.g. computer-assisted instruction, computer and society, human-computer interaction). He is also Head of the Digital Library and Distance Learning Laboratory. His research interests includes learning personalisation, metacognition, multimedia content, user experience and distance learning.

Dr Rizal F. Aji received his doctorate from the University of Indonesia. His reasearch interests include e-learning, computer network and information security. He is a lecturer and IT manager within the Faculty of Computer Science at the University of Indonesia. He teaches the course IT Infrastructure, Cloud Computing and Information Security.

APPENDIX

Table 5: Data filtering on issue and review-related metrics.

No.	Min Char	Issue created		Review request		Issue comment		Review comment		Issue created*		Review request*	
		R	*p*	*r*	*p*	*r*	*p*	*r*	*p*	*r*	*p*	*r*	*p*
0	0	0.3644	0.0954	-0.0554	0.8792	0.5851	0.0172	-0.8660	0.3333	0.3644	0.0954	-0.0554	0.8792
1	5	0.4330	0.0565	-0.6240	0.1341	0.5670	0.0211	-0.8660	0.3333	0.3716	0.0886	0	1
2	10	0.4409	0.0516	-0.6240	0.1341	0.5684	0.0216	-0.9990	nan	0.3619	0.0973	-0.1897	0.6248
3	15	0.4409	0.0516	-0.6240	0.1341	0.4941	0.0611	nan	nan	0.3472	0.1134	-0.4700	0.3459
4	20	0.4409	0.0516	-0.6240	0.1341	0.7431	0.0023	nan	nan	0.4288	0.0523	-0.9487	0.0513
5	25	0.4156	0.0683	-0.6240	0.1341	0.6924	0.0180	nan	nan	0.5320	0.0230	-0.9486	0.0513
6	30	0.3991	0.0812	-0.6182	0.1388	0.5439	0.1300	nan	nan	0.4684	0.0579	-0.9486	0.0513
7	35	0.3726	0.1160	-0.8365	0.0189	0.5439	0.1300	nan	nan	0.4606	0.0628	-0.8944	0.1055
8	40	0.3726	0.1160	-0.8023	0.0540	0.3592	0.3820	nan	nan	0.4083	0.1164	-0.8660	0.3333
9	45	0.4231	0.0710	-0.6789	0.1380	0.6486	0.1149	nan	nan	0.4146	0.1103	Nan	Nan
10	50	0.4218	0.0719	-0.6789	0.1380	0.1091	0.8158	nan	nan	0.4018	0.1228	Nan	Nan