

Software Process Metrics in Agile Software Development: A Systematic Mapping Study

Syeda Sumbul Hossain(✉), Pollab Ahmed, and Yeasir Arafat

Daffodil International University, Dhaka, Bangladesh
{syeda.swe,pollab35-1732,yeasir35-1501}@diu.edu.bd

Abstract. Software process metrics are being used for improving agile processes in software companies. This helps a lot in managing customer satisfaction and the growth of the company. Different studies have stated different process metrics and practitioners are using various metrics in terms of software process improvement. However, it is difficult to find out the best-suited metrics for every company as per the project nature. A systematic mapping study is used to identify potential process improvement metrics and finding the best metrics in agile development. We have found only 13 papers on process metrics. Finally, we have classified them into three categories assuming their impacting factors. This study provides an overview of agile software development process metrics that will help the practitioners to adopt efficient process metrics in Global Software Development (GSD).

Keywords: Software process improvement · Global Software Development (GSD) · Agile · Distributed software development · Process metrics · Systematic mapping study

1 Introduction

Maintaining software development life cycle models (SDLC) throughout the software development is crucial for the success of the software and also for the business. With this point of view, various SDLC models were introduced and practiced and the agile development model had got the most popular for its incremental development attitude. Agile works in an iteration or in a small module where every iteration takes 2–4 weeks. And every iteration implements some functions. Jinzenji [8] explained iterative software development in a very understanding manner. The main benefit of agile is that it can incorporate any valid functionalities at any time of the development process. Improving the agile software development process requires a lot of work for the development team as it requires maintaining multiple things at a time. Everything in agile is managed in phases and metrics come in help for this matter. Concas [3] implemented agile project phases with software metrics which provides a brief overview on how we

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12957, pp. 15–26, 2021.
https://doi.org/10.1007/978-3-030-87013-3_2

can implement and ease our agile project development by applying metrics in different phases. Traditional software development models used a lot of metrics in their development phase which has a great impact in managing, organizing, and improving the development process. Many companies implemented those metrics in agile but found that they are not fully capable of utilizing agile development. So, they tried to create new metrics according to their business and development needs and also fused some metrics with agile. Padmini [14] made a comparison of traditional software metrics and agile software metrics. A lot of literature reviews, case studies on companies, and empirical studies have been done for finding the metrics which will provide the highest amount of benefit in agile software development. A survey was done by Sanjay Misra and Martha Omorodion [12] where different software metrics are classified as Product Metrics, Process Metrics, Objective Metrics, Subjective Metrics, Resource Metrics, Project Metrics, Direct Metrics, and Indirect Metrics.

Process metrics are further classified into different metrics and a lot of those are already being used by different software companies according to their usages and needs. A lot of research had been done on this topic and many metrics also found out. It will be worth the findings if we can summarize the most common metrics that might be effective mostly in software process improvement. From that, we are motivated to identify the most common metrics that are stated in the literature through a systematic mapping study.

The further sections of this study are assembled as follows. Related works are presented in Sect. 2. In Sect. 3, the overall research methodology is described which is followed by results and analysis in Sect. 4. And finally, the conclusion is furnished in Sect. 5.

2 Related Work

Measuring the agile software development process is not that much different than traditional software development. But it still has some differences. For a software company practicing agile, measuring its projects under development and the quality of the products they create and serve is very much essential. It helps to understand the project's progression rate and manage it for further improvement [4]. Metrics are used for this aspect so that better visibility and insight can be found. It also helps for analyzing how well it is doing in development and did [16]. In general terms, it is a defining factor for agile process management. By my systematic mapping study, we have found very little study on agile process metrics. And it is also seen that many of the studies done in recent years which refer to agile are gaining popularity and it needs a lot of concentration from the researchers. Gustafsson [5] made a category of five metrics as Lean, Business value, Cost, Quality, and Predictability. Predictability and Business value refers to different surveys on business for finding a future insight of the business and predicting its shortcomings. Cost refers to the measurement of cost per function. Finally, Quality and Lean are for ensuring product quality assurance and working progress. Downey and Sutherland [4] found some

remarkable metrics which are very much beneficial for making decisions on managerial aspects. There are nine of them and they are, Focus Factor, Percentage of Found Work, Percentage of Adopted work, Velocity, Work Capacity, Targeted Value Increase, Win/Loss Record, Accuracy of Forecast, and Success at Scale. Oza and Korkala [13] divided agile software development into three categories, productivity level, economic level, and code level. Though they further divided this into seven, their primary motive is to find out good metrics for improving team performance in the agile process. Ram [18] had done a multiple case study on process metrics that provides effective operationalization is agile. Though agile works in big projects, many small projects are applying agile process metrics for better results [2]. According to ISO/IEC 9126:1991, software quality metric is a metric of quantity and the value of a feature of any software product can be measured by it. Metrics play a vital role in software development as it ensures the validity of certain factors of the product under development and already developed. So, finding out proper metrics for agile process improvement is a must for the growth of software development companies.

3 Research Methodology

For finding potential information about certain research areas, reviewing previous research papers is a must and well-established path. Kai Petersen et al. [15] performed a thorough analysis and discussion on systematic review and systematic mapping study where he suggested that systematic mapping study is better in finding potential information than a systematic review. We are also motivated to perform a systematic mapping study from one of our prior works [6].

3.1 Systematic Mapping Study

According to Kitchenham [15], getting an overview of a specific research area, a systematic mapping study is a perfect method. It keeps my focus on the research area and helps to find relevant papers. We have followed the overall mapping study process from [6,15]. Figure 1 represents the overall research methodology of this study.

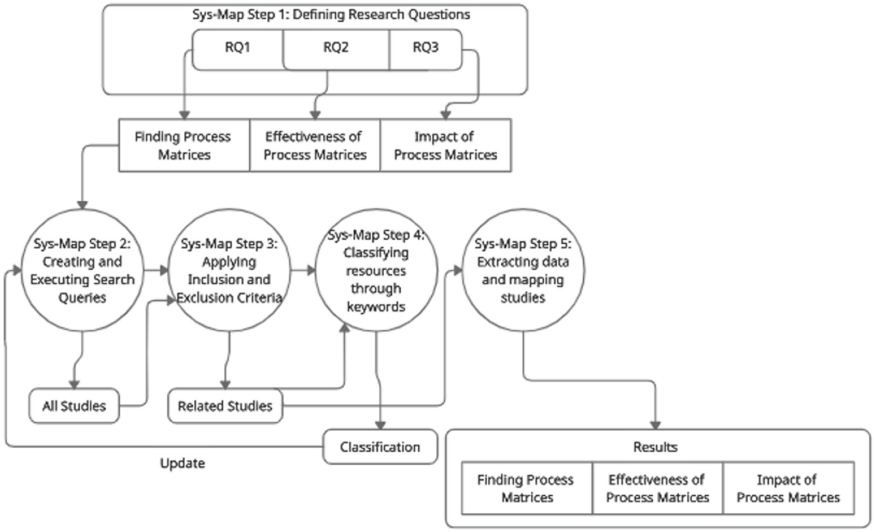

Fig. 1. Design of systematic mapping

Systematic Mapping Step 1: Defining Research Questions. Budgen and other researchers [1] agreed that the research question is the key point of every systematic mapping study. We have also specified research questions for conducting a systematic mapping study. We were driven by the following research questions:

– *Which software process improvement metrics are reported in the current state-of-the-art that has succeeded in terms of business, development, and customer satisfaction in global agile software development?*

Table 1 is presenting the breakdown of our Research Question.

Table 1. Systematic mapping research questions

Serial No.	Systematic Mapping Research Questions	Objective
RQ1	What are the process metrics that are used by software companies stated in the current state of the art?	To find out the process metrics used by software companies.
RQ2	Which process metrics have higher effectiveness in software process improvement?	To find out the matrices that provide more effect than others in process improvement.
RQ3	What is the impact of applying metrics in software process improvement?	To find the impact of process matrices in software companies

Systematic Mapping Step 2: Creating and Executing Search Queries. In this stage, we have designed our search strategy to define the searching process, digital databases, searching timeline. Then we have created our search string based on the research question and research scope. Digital sources and databases are used for applying search queries. For systematic search, the following procedures are performed. Software process improvement, quality metrics, and agile represent a multi-dimensional area of context. By observing, it is found that every study consists of keywords for referencing their terms and values. Snowballing approach [7] is proved to be the best approach for gathering and analyzing all the potential keywords. Search queries are specified by the following steps:

- All the potential keywords are identified by observing research titles, abstracts, and index terms of a paper.
- Including research area and mediation into the search queries find most relevant research papers.
- For finding similar keywords, thesaurus or reference book is used.
- Incorporating Boolean operators like AND and OR into search queries.

Table 2 shows the summary of the search strategy.

Table 2. Summary of search strategy

Data Items	Values
Databases	IEEE Xplore, Springer, Science Direct, ACM, Google Scholar
Scope Area	Software process improvement, quality matric, agile
Mediation	Keywords
Outcomes	Identify the gap, following the research questions and the amount of contribution done in this research area.
Reference Management	Mendeley
Year	2000 - 2020
Study Target	Journal and Conference Papers

Systematic Mapping Step 3: Applying Inclusion and Exclusion Criteria for Finding Relevant Resources. Inclusion and exclusion criteria are applied for finding relevant research papers and omitting papers that do not go with this research goal. According to Kitchenham, B. [9], introductory sentences and abstract hold were the main focus for excluding research papers. The below steps are followed for finding relevant papers:

- Papers must be written in English, there is no duplication, and peer-reviewed.

– The purpose of the primary study must be on keywords followed by abstract and title to be studied for further classification of the papers. If the motive of the paper is unclear, the introduction and conclusion can be studied for clarification.
– Only the relevant papers that are available in full text and clear were included.

Figure 2 represents the Inclusion & Exclusion Process of this mapping study.

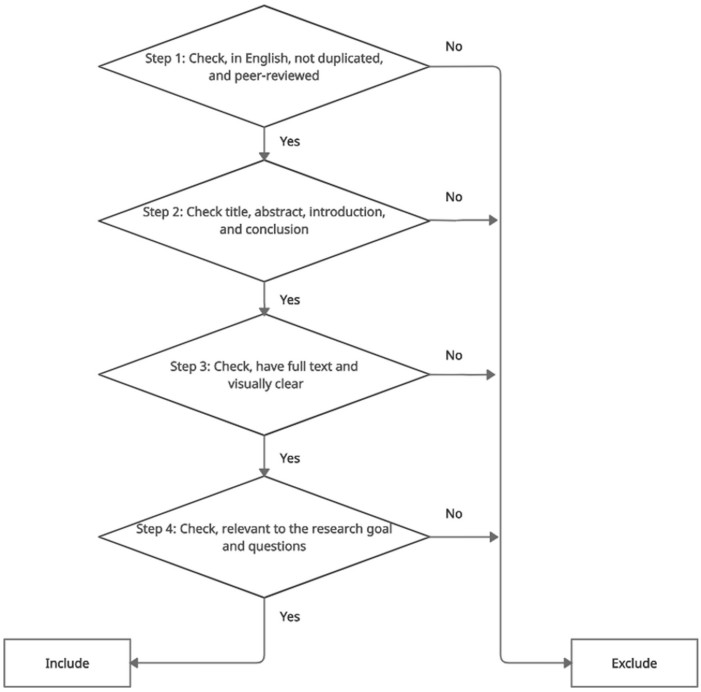

Fig. 2. Inclusion & exclusion process

Systematic Mapping Step 4: Classifying Resources Through Keywords. Resources are classified according to the objective or goal of the research area. Kitchenham elaborated that [9], keyword finding can be done with only two steps. At the first stage, we have gone through the title and abstract to find potential keywords and ideas about the contribution of any research. If the title and abstract could not provide enough details about the research, we have moved to the further stages by going through the introduction and conclusion to classify research papers accordingly.

Systematic Mapping Step 5: Extracting Data and Mapping Studies. All the related studies are mapped and reported visually by graphs and figures. They are listed by different perspectives like challenging issues, mitigation approaches, advantageous points, disadvantageous points, years, etc. Duplicate data are omitted from lists. Finally, they are summarized for a clearer overall view of the whole systematic mapping study.

4 Results and Analysis

4.1 Systematic Mapping Results

Initially, we have found 38 papers that were published between 2010 to 2021. After applying inclusion and exclusion criteria along with the snowballing process, we have selected 20 papers. After investigating all the 20 papers we have conclusively determined 13 papers that are most relevant to our research. Figure 3 represents the Systematic Mapping Results.

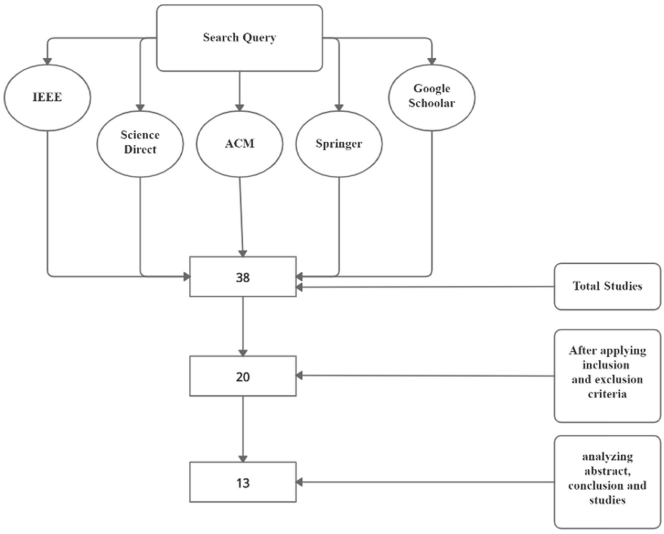

Fig. 3. Systematic mapping results

4.2 Process Metrics

From the study of systematic mapping, we have found a total of 19 software process metrics that have a direct impact on the agile software development process and used by software companies in the current state of the art. This answers our research question RQ1. Though some of them have higher and some of them have lower impact rates, their presence makes the agile development process successful. Table 3 summarizes all process metrics that are stated in different literature.

Table 3. Summary of process metrics

Metric Name	Metric Description	Study References
Data Quality Cost	Time, budget and resource to collect real data	[14,18,19,22]
Functional Quality Cost	Time, budget and resource to ensure functional quality	[5,14,18,22]
Client Communication	Effective communication with client in every phase	[5,14,17]
Forecast Accuracy	Clear outcome and risk measure	[4,14,19]
Estimation Accuracy	Accurate time, budget and resource estimation	[14,18]
Performance on Delivery	Reporting to client at delivery	[5,11,14,17,18,22]
Coverage of Unit Test	Code or function coverage at early development phase	[11,13,14]
Development Speed	Core development time	[4,5,14,17,18]
Focus Factor	Purpose of developing and testing a software	[4,14]
Team Assessment	The skills and capabilities of the team members	[4,17]
Work Adaptation Rate	The ratio of the requirements can be adapted by the team for the development purposes	[4,5,14]
Cycle Completion Rate	Ratio of complete a sprint successfully	[11,14,17,20,22]
Severe Defect Rate	Critical defects which have high risk	[10,14]
Slipping Defect Rate	The ratio of the number of defects found in the development and the execution phase	[5,14]
Defect Correction Time	Defect fixing turn around time	[10,11]
Clarity in Requirements	Understand-ability of the requirements to the development team	[14]
Defect Correction Efficiency	Ratio of defect correction and rejection	[10,14]
Team Communication	The communication between the team members	[21]
System Test	System test for ensuring client requirements	[5,11,21]

Table 3 is summarized in a column chart with an overview of the appearance of all the process metrics had been stated in different studies (Fig. 4).

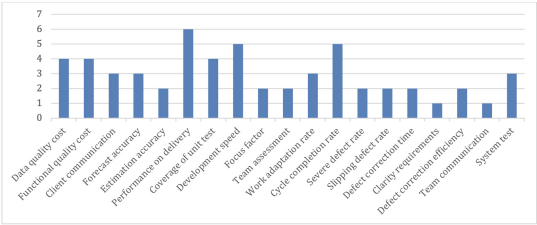

Fig. 4. Study frequency in terms of process metrics

4.3 Systematic Mapping Analysis

Padmini and Bandara [14] made a detailed overview of process metrics where the percentage of the most used metrics in software companies were included. We have categorized the metrics of the above table into three effective metrics according to the company usages and research studies, they can be referred to as high, mid, and low-level impact metrics. We can elaborate on the effectiveness of process metrics from two different perspectives. The first one goes to the companies which are practicing agile methodologies and the other goes to the research communities which are studying agile software development process improvements. Every company tends to follow the steps which lead them to earn more money. High-level effective metrics are highly regarded by every company and they are kind of a standard that a company should follow for improving the development process. Especially, Functional quality cost, Development speed, and Cycle completion rate are a must to maintain. By applying these metrics, companies can easily calculate the functional requirements and its quality maintenance cost. Every sprint cycle completion time can be observed which helps to measure team performance and find a lack of manpower or training sessions of the team members. Development speed also helps in monitoring the team's performance. Moreover, it helps to show off the company's capabilities in developing certain categories of software with other companies.

High-Level Impact Metrics. The high-level effective metrics Fig. 5 are Data quality cost, Functional quality cost, Performance on delivery, Coverage of unit tests, Development speed, and Cycle completion rate. Development speed and Cycle completion rate greatly depend on team assessment.

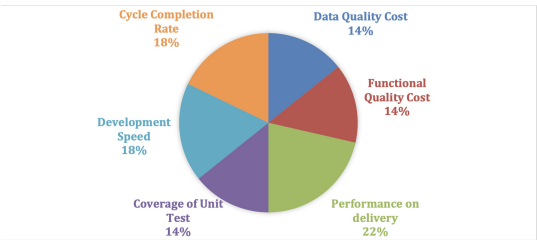

Fig. 5. High level impact metrics

Mid-Level Impact Metrics. The mid-level effective metrics Fig. 6 are Client communication, Forecast accuracy, System test, Work adaptation rate, Estimation accuracy, Team assessment, Severe defect rate, Focus factor, Slipping defect rate, Defect correction time, and Defect correction efficiency.

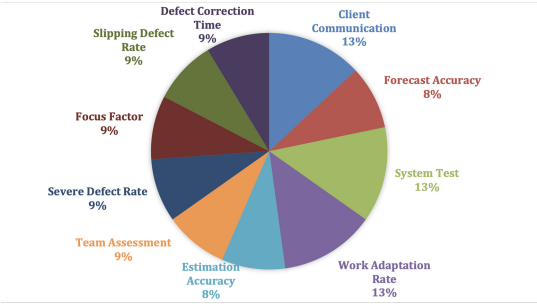

Fig. 6. Mid level impact metrics

Low-Level Impact Metrics. Finally, the low-level effective metrics Fig. 7 are Clarity in requirements and Team communications. So, the -level effective metrics answer our second research question.

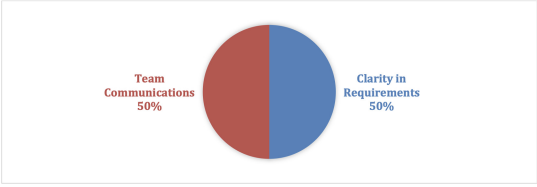

Fig. 7. Low level impact metrics

5 Conclusions

Agile software development metrics are divided into different categories. In this work, we focus on process metrics and make a comparative analysis through a systematic mapping study. This analysis shows that some metrics have a high effective ratio and some have not. Some metrics are constantly practiced by companies and researchers and some are less practiced. By following this systematic mapping study, a researcher can easily find out the more studied area and less studied areas of agile process metrics. On the other hand, applying these metrics according to their effectiveness in agile practicing companies will benefit them highly. This systematic mapping study tells the most studied areas and the lowest studied areas of agile process metrics. It also shows in which area the

researcher should give their focus for improvement. Based on the findings from this study, we will conduct a case study research on how practitioners focus on improving process quality by imposing different process quality metrics.

Acknowledgements. We thank Mr. Razzak M Abdur, Ph.D., who is working as Software Quality Coordinator at Ocuco Limited, Dublin, Ireland for his continuous support towards our research journey.

References

1. Budgen, D., Turner, M., Brereton, P., Kitchenham, B.A.: Using mapping studies in software engineering. Ppig. **8**, 195–204 (2008)
2. Choraś, M., et al.: Measuring and improving agile processes in a small-size software development company. IEEE Access **8**, 78452–78466 (2020)
3. Concas, G., Marchesi, M., Destefanis, G., Tonelli, R.: An empirical study of software metrics for assessing the phases of an agile project. Int. J. Softw. Eng. Knowl. Eng. **22**(04), 525–548 (2012)
4. Downey, S., Sutherland, J.: Scrum metrics for hyperproductive teams: how they fly like fighter aircraft. In: 2013 46th Hawaii International Conference on System Sciences, pp. 4870–4878. IEEE (2013)
5. Gustafsson, J.: Model of agile software measurement: a case study (2011)
6. Hossain, S.S., Arafat, Y., Amin, T., Bhuiyan, T.: Requirements re-usability in global software development: a systematic mapping study. In: Gervasi, O., et al. (eds.) ICCSA 2020. LNCS, vol. 12252, pp. 960–974. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58811-3_68
7. Jan, N., Ibrar, M.: Systematic mapping of value-based software engineering: a systematic review of value-based requirements engineering (2010)
8. Jinzenji, K., Hoshino, T., Williams, L., Takahashi, K.: Metric-based quality evaluations for iterative software development approaches like agile. In: 2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops, pp. 54–63. IEEE (2012)
9. Kitchenham, B.: Procedures for performing systematic reviews. Keele UK Keele Univ. **33**(2004), 1–26 (2004)
10. Korhonen, K.: Supporting agile transformation with defect management in large distributed software development organisation (2012)
11. Mannila, J.: Key performance indicators in agile software development (2013)
12. Misra, S., Omorodion, M.: Survey on agile metrics and their inter-relationship with other traditional development metrics. ACM SIGSOFT Softw. Eng. Notes **36**(6), 1–3 (2011)
13. Oza, N., Korkala, M.: Lessons learned in implementing agile software development metrics. In: UKAIS, p. 38 (2012)
14. Padmini, K.J., Bandara, H.D., Perera, I.: Use of software metrics in agile software development process. In: 2015 Moratuwa Engineering Research Conference (MERCon), pp. 312–317. IEEE (2015)
15. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12, pp. 1–10 (2008)
16. Pfleeger, S.L.: Software metrics: progress after 25 years? IEEE Softw. **25**(6), 32–34 (2008)

17. Pinto, N., Acuña, C., Cuenca Pletsch, L.R.: Quality evaluation in agile process: a first approach. In: XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016) (2016)
18. Ram, P., Rodriguez, P., Oivo, M.: Software process measurement and related challenges in agile software development: a multiple case study. In: Kuhrmann, M., et al. (eds.) PROFES 2018. LNCS, vol. 11271, pp. 272–287. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03673-7_20
19. Rathore, S.S., Kumar, S.: A study on software fault prediction techniques. Artif. Intell. Rev. **51**(2), 255–327 (2017). https://doi.org/10.1007/s10462-017-9563-5
20. Sandu, I.A., Salceanu, A.: Metrics improvement for phase containment effectiveness in automotive software development process. In: 2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), pp. 661–666. IEEE (2017)
21. Sandu, I.A., Salceanu, A.: System testing in agile sw development of the electronic components based on software from the automotive industry. In: 2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE), pp. 1–4. IEEE (2019)
22. Sneed, H.M., Prentner, W.: Analyzing data on software evolution processes. In: 2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 1–10. IEEE (2016)