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

Submitted by
Pollab Ahmed
163-35-1732
Department of Software Engineering
Daffodil International University

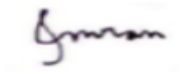
Supervised by
Ms. Syeda Sumbul Hossain
Lecturer (Senior Scale)
Department of Software Engineering
Daffodil International University

This Project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Science in Software Engineering.

Approval

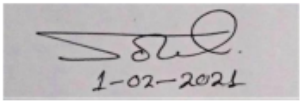
This Report titled **"A systematic mapping study on software process metrics in agile software development**", submitted by Pollab Ahmed, ID No: 163-35-1732 to the Department of Software Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS



Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



Internal Examiner 1

Md. Khaled Sohel
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Md. Maruf Hassan
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Prof. Dr. Mohammod Abul Kaashem
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering and Technology, Gazipur

Acknowledgement

At the beginning, I am thankful to the Almighty Allah for giving me the chance to learn and contribute to the world. I am also grateful to my supervisor, Ms. Syeda Sumbul Hossain, for her valuable advice, instructions and patience in helping me completing this paper. I would also like to thank our department previous head, Dr. Touhid Bhuyian and present head Dr. Imran Mahmud for their precious guidance.

The research work presented in this paper was conducted with the help of various digital libraries like Springer, ieeexplore and so on. I also like to thank them for helping me gathering useful resources and conducting my work.

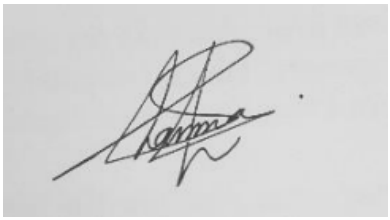
I would like to thank my all-batch mates (21th Batch) of the Software Engineering Department for inspiring me and providing me a great environment to work on my goal. All of my faculty members of the Department of Software Engineering and all students of this department should also be thanked from the core of my heart.

Finally, I would like to thank my family and friends for helping me by staying beside me in every situation of my life and provide me invaluable guidance.

Declaration

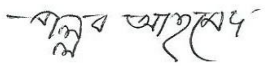
I hereby declare that I have taken this thesis under the supervision of **Ms. Syeda Sumbul Hossain, Lecturer (Senior Scale), Department of Software Engineering, Daffodil International University.** I also declare that neither this thesis nor any part of this has been submitted elsewhere for award of any degree.

Supervised by:



Ms. Syeda Sumbul Hossain
Lecturer (Senior Scale)
Department of Software Engineering
Daffodil International University

Submitted by:



Pollab Ahmed
ID: 163-35-1732
Batch: 21st Batch
Department of Software Engineering
Daffodil International University

Abstract

Context: Software process metrics are used for improving agile processes in software companies. This helps a lot in managing customers and in the growth of the company. Different studies found process metrics. But it is difficult to identify best suited metrics for every company.

Objective: As the number of process metrics is not a few, I am trying to identify the most suited process metrics that will benefit the software companies developing software using agile development models.

Methods: In this paper, systematic mapping study is used to identify potential agile process improvement metrics and finding the best metrics for using in the agile development process.

Results: By doing systematic mapping study, I have found only 13 papers on process metrics. I analyze them and find metrics that are highly regarded by fellow researchers. Finally, I arrange them in three categories.

Conclusion: The systematic mapping study provides an overview of agile software development metrics and further lets to dive into process metrics. It helps in finding process metrics to adopt in software companies and improve the process of agile software development.

Table of contents

List of tables

List of figures

List of charts

A systematic mapping study on improving agile software development process metrics

1. Introduction

Maintaining software development life cycle models (SDLC) throughout the software development is crucial for the success of the software. With this point of view various SDLC models were introduced and practiced. Among them, the agile development model had gotten the most popularity for its incremental development attitude. Agile works in iteration or in a small module where every iteration takes 2-4 weeks of time. And every iteration implements some functions. Jinzenji [10] 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 agile software development requires a lot of work for the development team as it requires to maintain multiple things at a time. Everything in agile is managed in phases and metrics come in help for this matter. Concas [2] implemented agile project phases with software metrics which provides a brief overview on how I can implement and ease our agile project development by applying metrics in different phases. Traditional software development models used a lot of metrics in its development phase which have 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 need and also fused some metrics with agile. Padmini [18] made a comparison on 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 [16] where they classified different software metrics that are used

in agile development. They are, Product Metrics, Process Metrics, Objective Metrics, Subjective Metrics, Resource Metrics, Project Metrics, Direct Metrics, and Indirect Metrics. In this paper, I am focusing on how I can improve agile process metrics.

Process metrics is further classified into different metrics and a lot of them are already being used by different companies according to their usage and need. But the problem is, I need to identify which metrics have the highest rate of improvement success in business, development and customer satisfaction. A lot of research had been done on this topic and many metrics also found out. I am doing a systematic mapping study in this context for finding the best metrics for agile software development process improvement.

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 [6]. 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 [21]. In general terms, it is a defining factor for agile process management.

By my systematic mapping study, I have found a 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 [8] 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 future insight of the business and predicting its shortcomings. Cost refers to the measurement of cost per function. Finally, Quality and Lean is for ensuring products quality assurance and working progress. Downey and Sutherland [6] 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 [17] 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 [23] had done a multiple case study on process metrics which provides effective operationalization is agile. Though agile works in big projects, many small projects are applying agile process metrics for better result. Springer [4] elaborated it in his paper.

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. To do so, I can either do systematic review or systematic mapping study. Kai Petersen et al. [19] 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 systematic review. That is the reason, I will use systematic mapping study in this research paper.

3.1 Systematic Mapping Study

Systematic mapping study is gaining popularity because of its efficiency in identifying the latest status of the research area and the amount of work had been done so far on that area. It can also help in pinpointing research gaps. Further, narrowing down of research papers by using search strings, it helps in saving time and cost for intensive researching. The significance of systematic mapping study in the software engineering field is recognized lately, said N. Condori-Fernandez [3]. Darnia Dicheva et al. [5] performed systematic mapping study for incorporating gamification

in study. To find as much research paper as possible, Emelie Engstrom et al. [7] used systematic mapping study for software product line testing. For getting the latest research status of Technical Debt and Technical Debt management and to know the concepts of it, a systematic mapping study was conducted by Zengyang Li et al. [14]. The procedure to follow when performing systematic mapping study is proposed in [18]. They are:

Procedure 1: Defining Research Questions

Defining research questions is the main goal of systematic mapping study as it manipulates the researchers in creating search queries and further studies. Moreover, goal-oriented study helps to find the result in a fast manner.

Procedure 2: Creating and executing search queries

Search queries are created with the goal keeping in mind. Through it, researchers can find potential papers or resources across different digital sources and databases.

Procedure 3: Applying inclusion and exclusion criteria for finding relevant resources

By following the research questions, inclusion and exclusion criteria is applied into found resources to get the highest amount of relevant research resources. It highly helps in reducing resources that are not relevant to the research area.

Procedure 4: Classifying resources through keywords

Resources can be classified by applying keyword finding techniques which follow two steps:

a) Only reading the title and abstract to find keywords that are relevant to the research saves a lot of time and effort.

b) If abstract and title does not help in finding research keywords, introduction and conclusion need to be read for gathering in depth concepts of the research.

Procedure 5: Extracting data and mapping studies

After gathering and classifying all the research resources, graphs are used to represent all the data to generate reports and mapping of studies.

3.2 Research Questions and methodology

For finding research questions and answers, several steps need to be taken. They are followed by:

Table 1: Mapping of Research Questions with Research Methodologies

Research Question	Research Step	Research Methodology
RQ1. What are the process metrics that are used by software companies stated in the current state of the art?	RS1.1 Finding out all the process metrics that are used in agile software projects.	Systematic Mapping Study
RQ2. Which process metrics have higher effectiveness in software process improvement?	RS2.1 Identify the metrics that have a higher effective ratio in software process improvement.	Systematic Mapping Study
RQ3. What is the impact of applying metrics in software process improvement?	RS3.1 Finding out and analyzing the metrics effectiveness in process improvement.	Systematic Mapping Study

4. Research Design

In this research paper, background study is the initial stage of finding out potential research papers that fulfils my criteria and getting to know the research gaps available in those papers. It is the input for my second stage, systematic mapping study. Where I follow the step-by-step process provided by systematic mapping study to explore more of the papers and finding out my research questions.

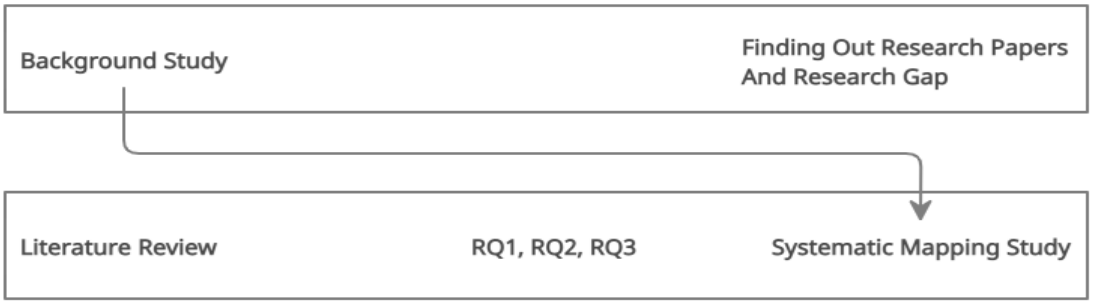


Figure 1: Study Design

4.1 Systematic Mapping Design

According to Kitchenham [19], getting an overview of a specific research area, systematic mapping study is a perfect method. It keeps my focus into the research area and helps to find relevant papers of my goal. Thus, reaching the goal becomes smoother.

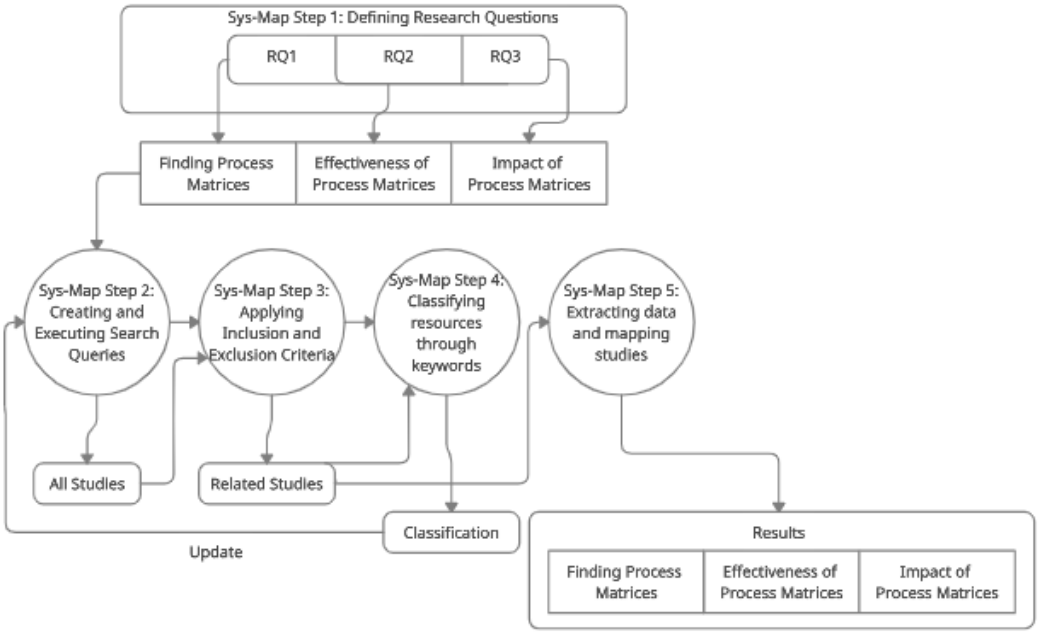


Figure 2: Design of Systematic Mapping

4.1.1 Systematic Mapping Step 1: Defining Research Questions

Research questions are the guide for research work. Getting answers to those questions narrows down my research work, tells the amount of effort I need and the path I need to follow. Budgen and other researchers [1] agree that the research question is the key point of every systematic mapping study. I have specified three research questions by following systematic mapping guidelines.

Table 2: Systematic Mapping Research Questions

Serial No.	Systematic Mapping Research Questions	Intention
RQ1	What are the process metrics that are used by software companies stated in the current state of the art?	Finding out process metrics used by software companies.

RQ2	Which process metrics have higher effectiveness in software process improvement?	Finding out the matrices that provide more effectiveness than others in process improvement.
RQ3	What is the impact of applying metrics in software process improvement?	Finding out the impact of process matrices in software companies

4.1.2 Systematic Mapping Step 2: Creating and executing search queries

Search queries are used according to the research scope and mediation between my research goal with other researches. Digital sources and databases are used for applying search queries. For systematic search, following procedures are performed.

Software process improvement, quality matric and agile represents 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 [9] 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. [9]

· For finding similar keywords, thesaurus or reference book is used.

· Incorporating Boolean operators like AND, OR etc. into search queries.

Table 3: Systematic Mapping Search Strategy

Data Items	Values
Databases	IEEE Xplore, Springer, Science Direct, ACM and Google Scholar
Scope Area	Software process improvement, quality matric, agile
Mediation	Keywords
Outcomes (Systematic mapping studying)	Identify the gap, following the research questions and the amount of contribution done in this research area.
Search Queries Formulation	Following strategy is applied into electronic databases for finding potential research papers. [9] · Between mediations, Boolean OR is used. · Between the research area and mediation, Boolean AND is used. · For finding only potential keywords, Boolean AND was also used in between them. · The example of my search query is given below: Software Process Improvement AND Quality Matric AND Agile

Reference Management System	I used Mendeley for referencing and Endnote for removing duplicate papers.
Year	2000 - 2020
Study Target	Journal and Conference Papers

Petticrew and Roberts [20] suggested that electronic databases cannot be the only source of literature and sometimes they may not be useful at all. That is why, manual and automated, both searches are used by me for maximum coverage of finding research papers. Conferences and researcher's websites are used for manual searching. For automated searching, my created search query is applied in different digital databases like IEEE Xplore, Springer, Scopus and Google Scholar.

4.1.3 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 does not go with my research goal. It is kind of like systematic literature studies [20]. According to Kitchenham, B. [12], introductory sentences and abstract hold my main focus for excluding research papers. The below steps are followed for finding relevant papers:

Step 1: The founded papers must be written in English, there is no duplication and peer reviewed.

Step 2: The purpose of the primary study must be on keywords software process improvement, quality matric and agile. Then, abstract and title have to be studied for further classification of the papers. If the motive of the paper is still unclear, introduction and conclusion can be studied for clarification.

Step 3: Only the relevant papers that are in full text and clear to all, can be included. If they are unavailable, they must be excluded.

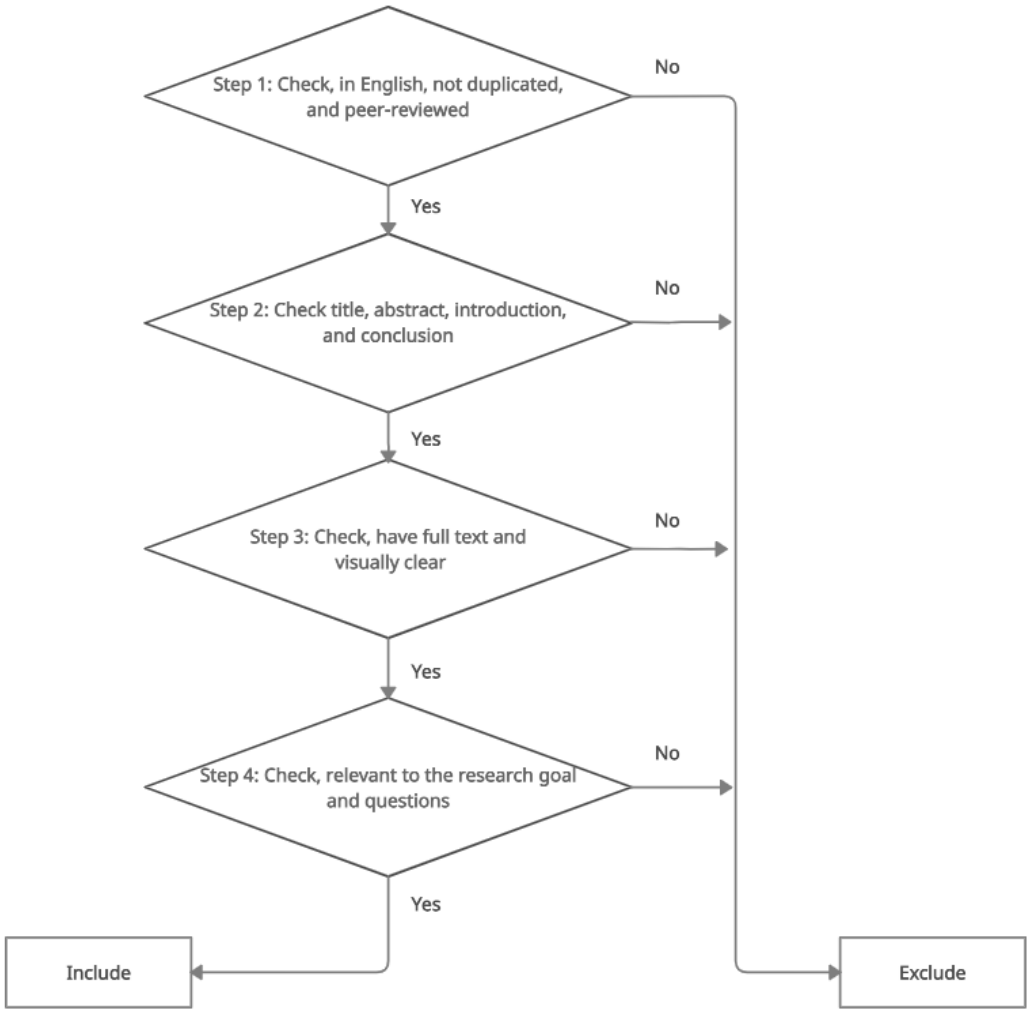


Figure 3: Inclusion & Exclusion Process

Step 4: The relevant papers must include its purpose and area of research.

Table 4: Inclusion & Exclusion Criteria

Criteria	Values

Inclusion	Relevant papers are: - · Interrelating with research goal and questions · Only written in English · In full text · Not duplicated · Peer reviewed · Including process improvement in agile or quality metrics in agile.
Exclusion	· Not written in English Language · Not in full text and unclear · Have duplicates · Inaccessible · Sessions containing editorials, prefaces, paper summaries, news, reviews, correspondence, discussions, comments, reader's letters and summaries of tutorials, workshops, symposium, panels, slides, and poster. · Cannot understand after going through title, abstract, introduction and conclusion.

4.1.4 Systematic Mapping Step 4: Classifying resources through keywords

Resources are classified according to the objective or goal of the research area. Kitchenham elaborated that [12], keyword finding can be done with only two steps. At the first stage, I go through the title and abstract to find potential keywords and ideas about the contribution of the research. It is the very base level of classification. If the title and abstract cannot provide enough details about the research, I need to go to the second stage. That is, trying to understand about the research by reading through introduction and conclusion. Thus, I try to figure out all the potential keywords and classify research papers accordingly.

4.1.5 Systematic Mapping Step 5: Extracting data and mapping studies

After completing classification, all the related studies are mapped according to the classification. Then, results 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.

5. Results and Analysis

5.1 Systematic Mapping Results

I collected agile software process improvement metrics from literature reviews, empirical studies and surveys which were published between 2010 to 2021. 38 papers are founded by me. I applied inclusion and exclusion criteria which leads me in finding 20 papers. After I have analyzed those papers on abstract, conclusion and deep studies for more specific results. Finally, I have found 13 papers.

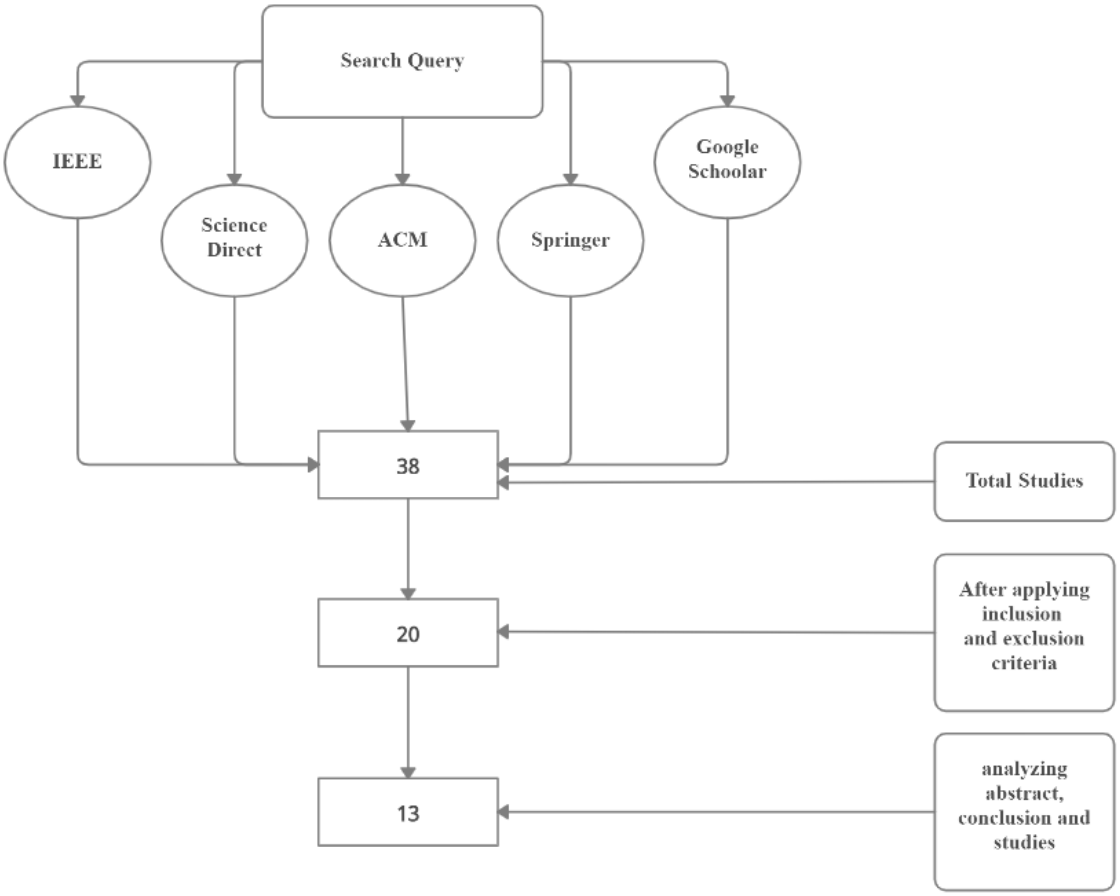


Figure 4: Systematic mapping results

From the study of systematic mapping, I have found more than fifteen metrics which have direct impact in the agile software development process and used by software companies in the current state of the art. This gives answer to my first research question. Though some of them have higher and some of them have lower impact rates, their presence makes the agile development process successful.

Table 5: Agile software development process metrics

#	Metric Name	Metric Description	Study References

1	Data quality cost	If the data is correct according to the work product or the accuracy of the data is measured in data quality. And for ensuring these qualities, I need to pay a huge sum of money	S6, S8, S11, S12
2	Functional quality cost	Every function in agile software development costs money. Further ensuring their quality needs to add extra cash from the client side.	S1, S6, S8, S11
3	Client Communication	From the requirement gathering to the software delivery, in every phase there will be a significant amount of client interaction with the project manager. Happening with the client affects the software process a lot.	S1, S6, S9
4	Forecast accuracy	Accurately predicting the outcome of every task related to the agile software development process. It can be defects, costs, delivery time etc.	S4, S6, S12
5	Estimation accuracy	Most of the time estimation accuracy talks about the estimated cost of the software. It's going well motivates the team highly and affects the development process.	S6, S11
6	Performance on delivery	When delivering the software, the performance of reporting to the client is a success factor. Representing every bit of information to the	S1, S5, S6, S8, S9, S11

		client is unnecessary. So, performing in delivery time is crucial.	
7	Coverage of unit test	Every software is developed with a lot of units. So, testing the units at the early stage reduces the amount of testing at the later stage. It highly increases the software development process.	S3, S5, S6, S7
8	Development Speed	This metric is dependent on many other metrics. But in general terms, it presents time the software takes for building from the beginning to the end. The less time it takes to build a software than other software's the development speed is that higher.	S1, S4, S6, S9, S11
9	Focus factor	The main purpose of developing and testing a software. Other than the focus factors a developing software can be gone off track.	S4, S6
10	Team assessment	The skills and capabilities of the team in a whole is measured in this metric. Agile development process highly depends on team capabilities.	S4, S9
11	Work adaptation rate	The rate of work from the requirements can be adapted by the team for the development purposes.	S1, S4, S6
12	Cycle completion rate	The amount of time it takes to complete a sprint cycle by the development team. Every sprint	S5, S6, S8, S9, S10

		time is calculated by the scrum master and assesses the team performance on that sprint or cycle.	
13	Severe defect rate	Defects have priorities and the highest priority of the defect is severe. For every software in agile, it is tested to find out if there is any severe defect in it.	S2, S6
14	Slipping defect rate	The ratio of the number of defects found in the development phase and the execution phase. The higher the ratio, the higher time it takes in the agile software development process.	S1, S6
15	Defect correction time	The amount of time taken in correcting defects. The lower the time, the higher capabilities the development team have. It is correlated with team assessment and defect correction efficiency	S2, S5
16	Clarity in requirements	As agile is an incremental development process, requirements change a lot. If the requirements are clear to understand and implemented by the development team, the development process takes less time.	S6
17	Defect correction efficiency	How well and fast defects can be solved by the development team. Team assessment plays a	S2, S6

		crucial role here and helps to reduce the development process time.	
18	Team communication	The communication between the team members affects the agile development process. Every person is not equally qualified. One is weak in one aspect and strong in another. Team members can help each other to overcome their weakness and contribute to the development process.	S13
19	System test	The overall system is tested for ensuring its functionalities are aligned with the client requirements and there is no major flaw in the system. It is actually done by the testers and team assessment plays a role here also.	S1, S5, D13

Now, if I summarize the above table in a column chart, I can easily overview the appearance of all the process metrics had studied and their impact on different companies.

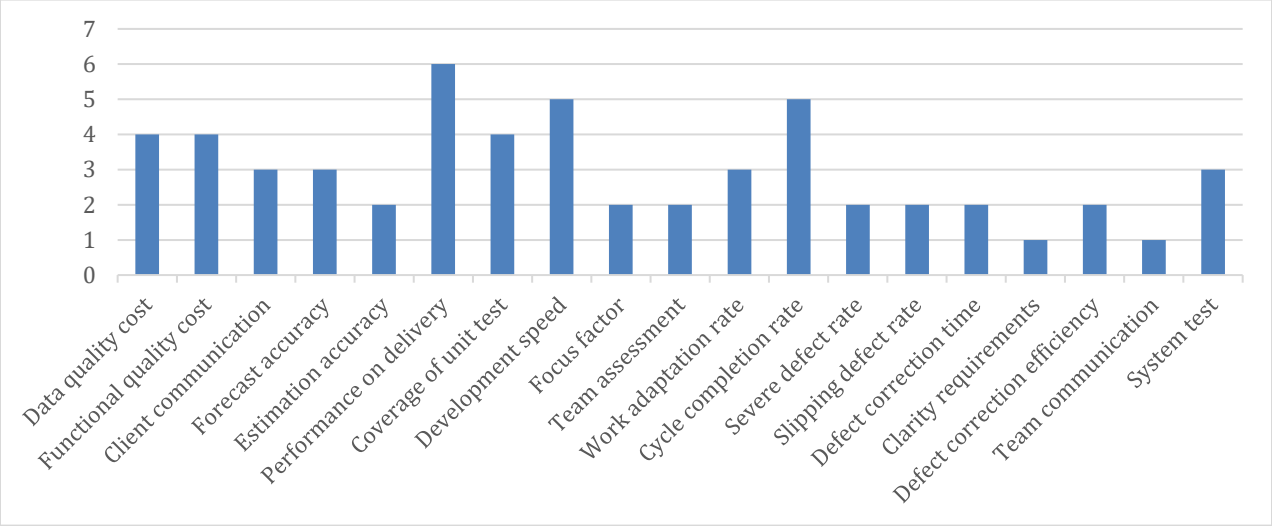


Chart 1: Number of appearances of process metrics

5.2 Systematic Mapping Analysis

Padmini and Bandara [18] made a detailed overview of process metrics where the percentage of the mostly used metrics in software companies were included. Their work matches a lot with my systematic mapping study. By following the Table: 5 above, I can see that some of the metrics are studied and practiced more than others. If I 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 effective metrics.

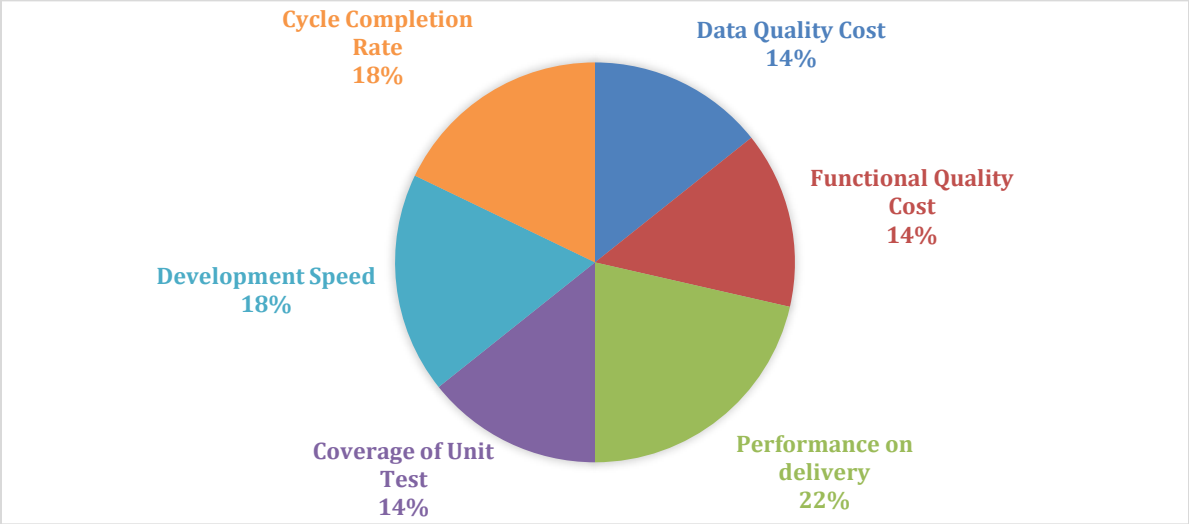


Chart 2: High-Level Effective Metrics

The high-level effective metrics are: Data quality cost, Functional quality cost, Performance on delivery, Coverage of unit test, Development speed, and Cycle completion rate. Development speed and Cycle completion rate greatly depends on team assessment.

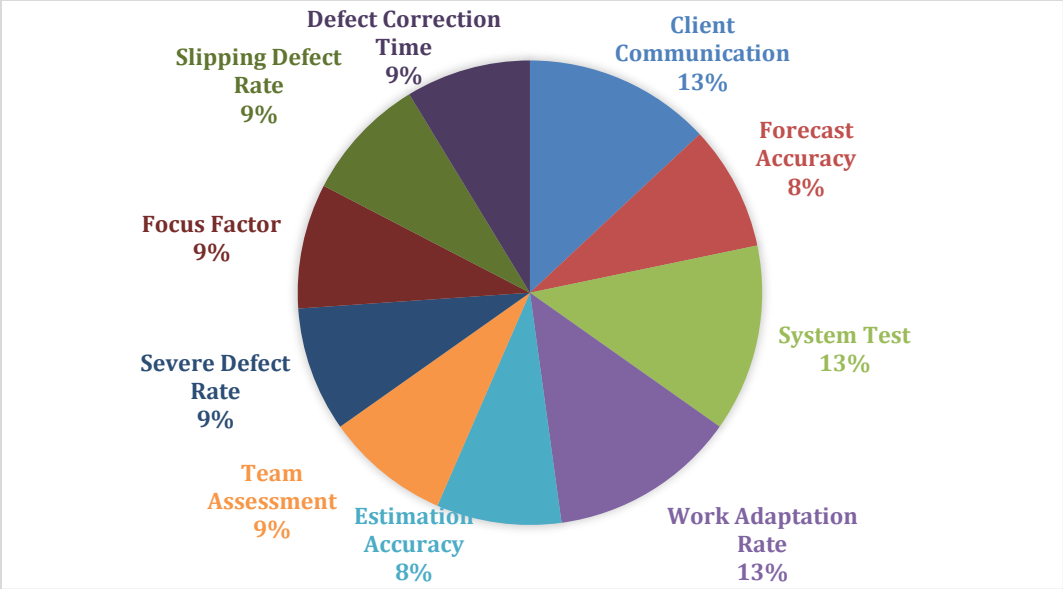


Chart 3: Mid-Level Effective Metrics

The mid-level effective metrics 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.

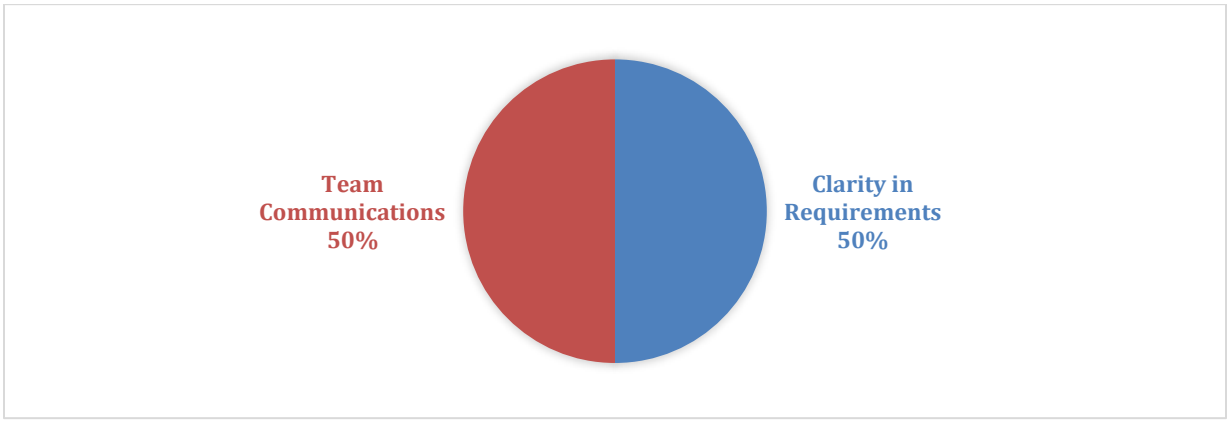


Chart 4: Low-Level Effective Metrics

Finally, the low-level effective metrics are: Clarity in requirements and Team communications. So, the -level effective metrics give the answer of my second research questions.

I can elaborate the effectiveness of process metrics in 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 leads 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. Specially, Functional quality cost, Development speed and Cycle completion rate is 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 lack of manpower or training sessions of the team members. Development speed also helps in monitoring the team performance. Moreover, it helps to show off the company's capabilities in developing certain categories of software with other companies.

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.

6. Conclusion and future work

Agile software development metrics are divided into different categories. In this research paper I focus on process metrics and make a comparative analysis through systematic mapping study. My analysis tells 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, researcher can easily find out the more studied area and less studied area in agile process metrics. On the other hand, applying these metrics according to their effectiveness in agile practicing companies will benefit them highly. I am hoping to do a case study on agile software development process metrics for getting actual scenarios on this area of work for my future studies. I will try to apply my found metrics on different companies to find out their effectiveness in a real-life environment.

7. Appendix

Appendix A: References of selected studies

Serial	Year	References
S1	2011	Gustafsson, J. (2011). Model of Agile software measurement: A case study
S2	2012	Korhonen, K. (2012). Supporting agile transformation with defect management in large distributed software development organisations.
S3	2012	Oza, N., & Korkala, M. (2012, March). Lessons Learned In Implementing Agile Software Development Metrics. In UKAIS (p. 38).
S4	2013	Downey, S., & Sutherland, J. (2013, January). Scrum metrics for hyperproductive teams: how they fly like fighter aircraft. In 2013 46th hawaii international conference on system sciences (pp. 4870-4878). IEEE
S5	2013	Mannila, J. (2013). Key performance indicators in agile software development.
S6	2015	Padmini, K. J., Bandara, H. D., & Perera, I. (2015, April). Use of software metrics in agile software development processes. In 2015 Moratuwa Engineering Research Conference (MERCon) (pp. 312-317). IEEE.
S7	2015	Jones, C. MSc in Software Development. (Introducing behavior-driven development and automated testing in a scientific software development

		SME)
S8	2016	Sneed, H. M., & Prentner, W. (2016, October). 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.
S9	2016	Pinto, N., Acuña, C., & Cuenca Pletsch, L. R. (2016). Quality Evaluation in Agile Process: A First Approach. In XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016).
S10	2017	Sandu, I. A., & Salceanu, A. (2017, March). 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.
S11	2018	Ram, P., Rodriguez, P., & Oivo, M. (2018, November). Software process measurement and related challenges in agile software development: a multiple case study. In International Conference on Product-Focused Software Process Improvement (pp. 272-287). Springer, Cham.
S12	2019	Rathore, S. S., & Kumar, S. (2019). A study on software fault prediction techniques. Artificial Intelligence Review, 51(2), 255-327.
S13	2019	Sandu, I. A., & Salceanu, A. (2019, March). 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.

8. References:

1. Budgen, D., Turner, M., Brereton, P., & Kitchenham, B. A. (2008, September). Using Mapping Studies in Software Engineering. In PPIG (Vol. 8, pp. 195-204).
2. Concas, G., Marchesi, M., Destefanis, G., & Tonelli, R. (2012). An empirical study of software metrics for assessing the hases of an agile project. International Journal of Software Engineering and Knowledge Engineering, 22(04), 525-548.
3. Condori-Fernandez, N., Daneva, M., Sikkel, K., Wieringa, R., Dieste, O., & Pastor, O. (2009, October). A systematic mapping study on empirical evaluation of software requirements specifications techniques. In 2009 3rd International Symposium on Empirical Software Engineering and Measurement (pp. 502-505). IEEE.
4. Choraś, M., Springer, T., Kozik, R., López, L., Martínez-Fernández, S., Ram, P., ... & Franch, X. (2020). Measuring and improving agile processes in a small-size software development company. IEEE access, 8, 78452-78466.
5. Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: A systematic mapping study. Journal of Educational Technology & Society, 18(3).
6. Downey, S., & Sutherland, J. (2013, January). Scrum metrics for hyperproductive teams: how they fly like fighter aircraft. In 2013 46th hawaii international conference on system sciences (pp. 4870-4878). IEEE.
7. Engström, E., & Runeson, P. (2011). Software product line testing–a systematic mapping study. Information and Software Technology, 53(1), 2-13.
8. Gustafsson, J. (2011). Model of Agile software measurement: A case study.
9. Jan, N., & Ibrar, M. (2010). Systematic mapping of value-based software engineering: A systematic review of value-based requirements engineering.
10. Jinzenji, K., Hoshino, T., Williams, L., & Takahashi, K. (2012, November). 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.
11. Jones, C. MSc in Software Development. (Introducing behavior-driven development and automated testing in a scientific software development SME)
12. Kitchenham, B. (2004). Procedures for performing systematic reviews. Keele, UK, Keele University, 33(2004), 1-26.

13. Korhonen, K. (2012). Supporting agile transformation with defect management in large distributed software development organisations.
14. Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. Journal of Systems and Software, 101, 193-220.
15. Mannila, J. (2013). Key performance indicators in agile software development.
16. Misra, S., & Omorodion, M. (2011). Survey on agile metrics and their inter-relationship with other traditional development metrics. ACM SIGSOFT Software Engineering Notes, 36(6), 1-3.
17. Oza, N., & Korkala, M. (2012, March). Lessons Learned in Implementing Agile Software Development Metrics. In UKAIS (p. 38).
18. Padmini, K. J., Bandara, H. D., & Perera, I. (2015, April). Use of software metrics in agile software development processes. In 2015 Moratuwa Engineering Research Conference (MERCon) (pp. 312-317). IEEE.
19. Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008, June). Systematic mapping studies in software engineering. In 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12 (pp. 1-10).
20. Petticrew, M., & Roberts, H. (2008). Systematic reviews in the social sciences: A practical guide. John Wiley & Sons.
21. Pfleeger, S. L. (2008). Software metrics: Progress after 25 years? IEEE Software, 25(6), 32-34.
22. Pinto, N., Acuña, C., & Cuenca Pletsch, L. R. (2016). Quality Evaluation in Agile Process: A First Approach. In XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)
23. Ram, P., Rodriguez, P., & Oivo, M. (2018, November). Software process measurement and related challenges in agile software development: a multiple case study. In International Conference on Product-Focused Software Process Improvement (pp. 272-287). Springer, Cham
24. Rathore, S. S., & Kumar, S. (2019). A study on software fault prediction techniques. Artificial Intelligence Review, 51(2), 255-327

25. Sandu, I. A., & Salceanu, A. (2017, March). 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
26. Sandu, I. A., & Salceanu, A. (2019, March). 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
27. Sneed, H. M., & Prentner, W. (2016, October). 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