



Software Process Improvement and Estimation: A Systematic Literature Review.

Anas Usman Inuwa¹, Mohammed Kabir Dauda², Maryam Abdullahi Musa³, and Abdulmajid Babangida Umar⁴

¹Department of Mathematical Sciences, Faculty of Science, Taraba State University Jalingo, Nigeria

^{2,3}Department of Mathematical Sciences, Faculty of Science, Abubakar Tafawa Balewa University Bauchi, Nigeria

⁴Department of Computer Science, Yusuf Maitama Sule, University Kano, Nigeria

anaspullo@yahoo.com

ARTICLE INFO

History:

Received: 1st May, 2022

Accepted: 5th June, 2022

Received in revised form:

10th July, 2022

Published: 28th August, 2022

Keywords

Component, Database, Data extraction, Software evaluation, Software metrics, Search process, Software process Improvement.

ABSTRACT

Software process improvement and estimation is a vital issue in IT and software industries, by drawing attention of developers and system analyst to wonder on how to improve quality and productivity of the products. Currently, SLR on empirical studies of agile software development are few results found discussion specifically about process improvement, rather they only presented suggestions on self and quality improvement. The studies on software process improvement will address the issues regarding success stories, factors and barriers for assisting companies towards implementing software. The objective of this study is to comprehensively conduct a review of several available techniques, models, frameworks and approaches for “software process improvement”, software evaluation, and software process validation, success stories, problem and challenges encountered along the line. As a result, Systematic Literature Review (SLR) method was used on 47 selected primary studies were reviewed and thoroughly analyzed. It was realized that there is high demand to develop more techniques that are very specific to software process improvement and estimation for quality and productivity of the products. From more analysis and our readings of the selected articles were found that 60% of the articles are industrial reports and about 20% are case study researches, while others constitute the remaining 20%. However, the distinction between case study research and industrial reports, by considering the study was planned empirically and been carried out with clear research goals and questions alongside with metrics/measure.

© 2022 TAJET All rights reserved

1.0 Introduction

Systematic literature review (SLR) is nowadays gaining interest of many researchers in so many fields; software engineering is not an exception. SLR gives you an in-depth knowledge in a particular field, the previous work and solutions carried out on a particular topic, success stories, drawbacks and future research trends in the topic at hand. It is quite not much being practiced in the field of software engineering, and until now, there is only one guide on performing SLR in software engineering that was compiled by Kitchenham early 90's, which was adopted from medical field SLR guidelines [1, 2].

Software engineering is a field dealing with the production and enhancement of traditional practices to systems and more technologically wise applications, simplifying methods, approaches and boosting business processes by saving time, human labor, and cost reduction and other resources. It is now the backbone and entity of existence of systems and services. Due to this high importance attached to software and systems, it is deemed necessary to measure and quantify the process of its development, evaluate the steps and the products, and find ways to improve and simplify the production

process. This will eventually lead to proper utilization of resources, increase market competition.[3, 4].

Measurement is a field and a process attributed to finding the quality, by setting up questions that will help in evaluating the quality and set-up metrics/steps that must be used in order to achieve the measurement process. It is used in characterizing, evaluating and predicting software entities, including products, process and resources in order to assess, monitor, control, and improve quality, accuracy of estimation and productivity. Tom DeMarco said “You cannot control what you cannot Measure” [2, 3]. There are so many definitions of measurement. However ISO/IEC defines measurement as “a process for establishing, planning, performing and evaluating measurement, within an overall project, enterprise or organizational measurement structure[4]”. Therefore, software industries are implementing measurement programs as part of their development processes.

However, developing measurement templates require the software industries to follow and adopt certain standards for consistency and continuity. In planning the measurement process, most of the industries use standards like: “ISO/IEC 15939 (2007),



CMMI, ISO/IEC 25021, ISO 9126 standard family and ISO/IEC 14598-x [4],” and the use of procedures which are based on their experiences, heuristics (like Nelson 10 heuristics), and company’s goal, standard or guide, planning models, tools, stakeholders involved in inputting or making use of data [7, 8].

Implementation of measurement processes has been a challenge for the past number of years as reported by Briand et al. in the year (1996). They found that measurement process is lacking clear objectives, sustainability, valid measurement instruments, resource utilization and proper budgeting. They also realized most of the measurement processes could not stand for more than two years [6]. But they are able to find out that the reasons for the failure to be lack of relationship between the measurement process and business goals of the particular organization, the organizational transactions, and poor merging of the measurement process and corrective actions in overcoming the issues at hand. On the other hand, Basili and his team found that: Metrics should be very sound, and the data.

Collected must be a complete one, should also be consistent with users requirement and goals, which are hint for making a good measurement process. Importance of measurement process, have been highlighted by Kitchenham. Measurement/metrics are to be formalized as part and person of development process in any given Software Company. Factors like: measurement patterns, tool support, GQM oriented metrics, are presumed to be keys to successful implementation of measurement process in software development process [7].

There are a number of techniques, methods, and models for software estimation, process improvement and software evaluations in literatures. It is now very popular that most measurement algorithms are based on GQM (goal-question-metric). Others include, Goal question indication metric (GQIM), goal argument metric (GAM), and Measurement information model found in ISO/IEC 15939:2007. Researchers raised the need for enhancing and extending the existing techniques and models due to the weaknesses realized in the models. In this article, SLR on software process improvement and estimation is represented, where a lot of publications were gathered on: Software evaluation, software process improvements, and software estimation, details will be found in the next section. So, for the purpose of this study, the following research questions were formulated:

RQ1: What techniques, methods and models are available in the literature, concerning: software evaluation, software improvement process and software estimation?

RQ2: What are the challenges, problems faced by the existing techniques?

RQ3: What are the recommendations from the literature suggested to improve software improvement and estimation process?

RQ4: What are the factors will lead to failure in the future research?

In general, SLR are meant to give a researcher a deep information/knowledge on existing problems/solutions, related work and research trend on the topic at hand, for further enhancement and improvement. Hence, this study is also not an exception to that. In addition, recommendations and some useful information will be found in the appendix tables. The provided research questions will assist the future researchers in identifying the critical areas, challenges in the aspects of software process improvement (SPI) and estimation for product quality.

2.0 Related Works

This section describes the summary of related work regarding software improvement and software estimation which are closely related to the research. Danh Nguyen-Cong and De Tran-Cao, conducted an SLR on Effort Estimation Studies in Agile, Iterative and Incremental Software Development. Their aim was to find research and common trends, the existing gaps and recommendations for further research. Which found that there is a need for an in-depth research on estimating effort, analyzing the effects of previous and recent project data, the use of standard, use of composite model, and further empirical and validation of model on Agile, Iterative and Incremental Software Development Process [3].

A.K.M. Moinul Islam and Chow KianCheng conducted an SLR on: Evaluation and Measurement^[SEP] of Software Process Improvement. They defined software process improvement as: “a systematic approach to increase the efficiency and effectiveness of a software development organization and to enhance software products [38]”. They Identified: “Seven distinct evaluation strategies, wherein the most common one, as “Pre-Post Comparison,” being applied in 49 percent of the inspected papers. Quality was the most measured attribute (62 %), followed by Cost (41 %), and Schedule (18 %). By looking at measurement perspectives, “Project” represents the majority with 66 percent [3]”. They concluded by saying that the challenges of software process improvement have to do with some confounding factors including pre-post comparison, in accurate description of the context of the evaluation. And measurement/metrics to evaluate short and mid-term effects of software process improvements initiates is prevailing whereas the long-term measurements with respect to customer satisfaction and ROI (return on investment) is not prevailing[11-13].

Marco Kuhrmann and Philipp Diebold conducted studies on the topic: How Does Software Process Improvement Address Global Software Engineering? The studies was carried out on 30 articles, and their findings showed that the analyzed papers delivering substantial discussion of cultural models and how to improve the

models can be used to address better the software process improvement process programs which are in a multi-national environments. They also found out how agile approaches should be implemented in companies operating in a global scale. At the end, they presented success stories, factors, and barriers for assisting companies towards implementing software improvement process and the context of global software engineering [10].

Célio Santana and Fabiola Queiroz conducted an SLR on: Software Process Improvement in Agile Software Development. They identified and characterized software process improvement in an agile environment through SLR with about 423 published articles within the range of 2001-2003, characterized according to SPI aspects, where by distinct approaches of agile software process improvement were identified, and the most popular one's is found to be "traditional approaches of SPI in agile environment", however there are novel approaches like improving of people's behavior and practices as innovative in SPI considerations. Lastly they presented an argument on: "what are the roles of SPI in a context with people and interactions are more valued than process and tools? [2-16].

Dyba and Dingsoyr performed an SLR on empirical studies of agile software development and not much result are found talking specifically about process improvement, rather they only presented suggestions on self and quality improvement. They also conducted another SLR on 12 "agile software development and suggested that many aspects, such as, process improvement still needs further research [2-18].

Unterkaalmsteiner and his team conducted an SLR identifying and characterizing measurement techniques and evaluation strategies which are used in assessing the impacts of different SPI initiatives. And: "They found seven distinct evaluation strategies, whereas the most measured attribute was Quality (62%), and followed by Cost (41%) and Schedule (18%). They conclude that evaluation validity of SPI initiatives is challenged by the scarce consideration of potential confounding factors and the inaccurate descriptions of the evaluation context. Measurements to assess short and mid-term impact of SPI initiatives prevail, whereas long-term measurements tend to be less used [2-20].

3.0 Systematic Literature Review

Planning the SLR: The SLR was planned, and shall see in the following steps, described in details

Reason for the SLR: The reason and essence for conducting this study is to gather and examine various literatures in the past and at the recent time discussing on: software process improvement, software estimation and measurement, software evaluation, and to explore the success and issues/problems found along the line, and lastly to suggest and give a roadmap for future research in the field.

The Search Strings: The following search terms were used in looking up related work, journal articles, conference papers, books, and magazines talking about: software, evaluation, process improvement, estimations, and measurement.

The phrases are: ("Software Metrics + Software Measurement"), (Software Process improvement AND Evaluation), "Software Process Improvement" AND Systematic Literature Review), (SLR + Software Metrics), ("Systematic Literature Review in Software Process Improvement), (Evaluating software development process and Improvements"), (Software Estimation Techniques and Process), (Software Metrics AND Systematic Literature Review).

In either case, the articles were selected based on the titles, abstract, introduction, conclusion, findings and future research recommendations, year of publication, and the language used (in this research, only articles published in English language were considered). And also, verified that no same publication being published in more than one journal and no two publications bears the same title, but one author can be found having two or more publications, which are related and being the continuity of the previous publication or article, as this practice is well accepted by the research world.

Methodology/flow for the SLR: Figure 1 shows the SLR process methodology/flow.

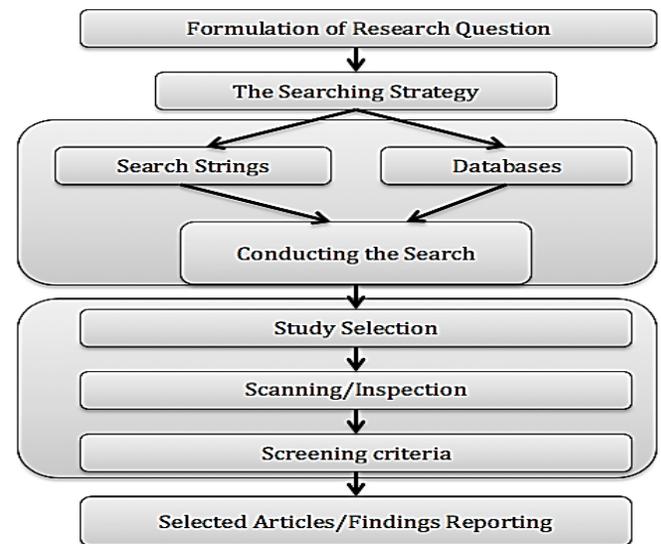


Figure 1. The SLR Process flow/methodology [11]

Research Questions: Table 1 below shows the research questions for this study, and the motivations behind choosing or formulating them:

Table 1. The Research questions and the motivations

ID	Question	Rationale behind
RQ1	What techniques, the methods and models available in the literature, concerning software evaluation, software improvement process and software estimation?	To understand the state of art in Software process and improvement, estimation and evaluation from the literatures selected.
RQ2	What are the challenges, problems faced by the existing techniques?	Here we need find and examine the problems and issues with the current techniques and see how they are tackled.
RQ3	What are the recommendations from the literature suggested to improve software improvement and estimation process?	In here, we shall be gathering all recommendations made by authors of the respective selected studies in a compressive manner and to make a summary of all.
RQ4	What factors lead to failure, and how to avoid them in future research?	To obtain the causes in order to take solution measure.

Search Process: The appropriate search terms are used in selecting and finding the primary study articles. Hence, by including the term “Software engineering”, from a single reasonable article obtained, ten others can be obtained from its references and cited-by list. On the other hand, attention was made in the research questions, due to the fact that the SLR is determined to answer them. The synonyms, and alternatives of the key terms were also considered, the use of logical terms like: AND and OR were also applied.

Database Consulted: The databases consulted are the most common electronic journals and articles repositories, including: Google search, Google Scholar, the ACM Digital libraries, Springer link, (and other Springer databases) Science Direct, Web of Science (that is ISI), IET libraries, and some other university library articles and books within the context of software measurement. For every single search, there are many found result of which we selectively downloaded randomly for screening at inclusion and exclusion stage as will be explained next. However, the articles are mixed up that’s to say the set of primary selected studies includes: journal articles, book chapters, and presentation and conference proceedings.

The Exclusion Criteria: Reports that were made in languages other than English were excluded, studies with complicated

mathematical formulas and thus difficult to be understood, are also excluded. Therefore, all irrelevant journal articles, book that are outside the scope of software engineering weren’t considered.

The inclusion criteria: These are the criteria used in selecting the primary most relevant paper. Specifically the work focused on articles/papers dealing with our outlined research questions, and relevant to process improvement in software development, software evaluation, and software metrics. The following are the more specific criteria for inclusion in the primary studies.

- a. The general criteria are:
 - The studies are either; reviews, SLR, or in measurement domain
 - The study is relevant and related to domain of software evaluation, metrics, or estimation
 - The study included a kind of data, experiment, survey, or statistical representation.
- b. Criteria which are in-line with research questions:
 - The study might have/contain analysis result, measurement and planning models, tools or practices
 - The study might have highlighted about problems/challenges, and suggested possible solution
 - Road map for future research.
 - And within 2010-2016, however there may be few articles out of the range, but found relevant to the studies.

Assessment/Quality of the primary studies: In conducting SLR one need to select quality papers found in recognized database (IEEE, Science Direct etc.), which will help in conducting a good SLR report. For example: are the aims and objectives of the studies clearly presented? Are the findings applicable to academics or industrial perspectives? Is there good relationship between the research questions and the contents of the paper? And lastly were the findings in-line with the objectives of the study?

Data Extraction: At this stage a table was used that contained the primary studies, showing: Author, title of the paper, the model being proposed, and recommendations with personal remarks on the relevancy of the research. Hence the studies are either: Case study (dealing with particular area or concept), Industrial Report (from a particular industrial work), Experiment (performed on a particular subject), Action research (with clear goals and well defined methodology as in academics), and Survey research (here the use of data or sample is required).

4.0 Conducting the SLR

Below are the search and the selection process for the primary performed studies as illustrated in the Figure 2.

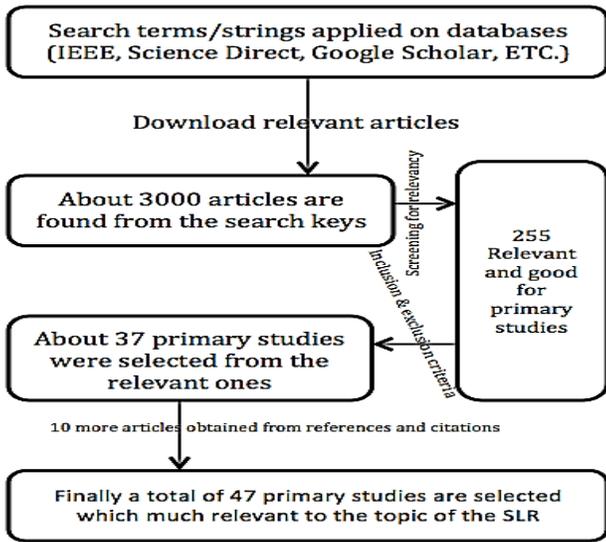


Figure 2. The Search and the selection process for primary studies[4]

However the figure only shows number of articles and conference papers, but classification and sources of the articles are given in the following Table 2, (Search engines, Journals consulted, and conference papers are as shown in the table).

NB. Book chapters, magazines and are not included.

Table 2. Primary study search, Journals and Databases

S/N	Journal Name/Database name	No. of Articles on search
1.	Springer	1000
2.	IEEE	1300
3.	Science Direct	400
4.	ACM digital library	50
5.	Google Scholar	1450
6.	Scopus	38

N.B the table above shown the primary search journals and database which were used according to different range of years. Sometimes, it can even get zero result, which can be due to the search term used. At long run, the number reduced as selection for relevancy and the criteria for inclusion and exclusion paying attention to the range of years (2010-2016, however few others can be found out of the range). The number go on and on reducing until 37, and later 10 more potential ones were obtained in the citations and reference list of some papers. However we shall see the classification of the 47 finally selected primary studies, as to whether they are: journal publications, conferences, and other sources of articles and reports.

At firstly, let’s see the distribution of our selected articles, over the years: see the bar chart below:

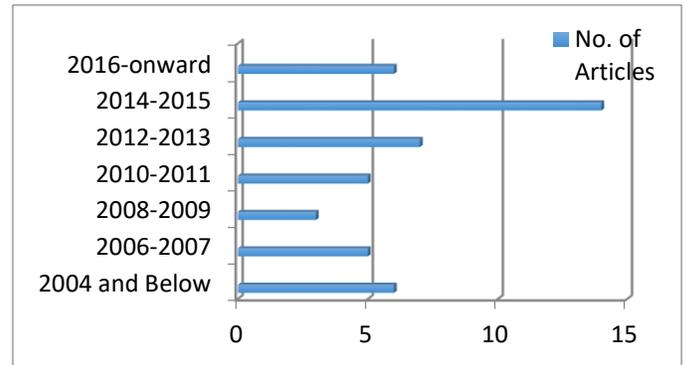


Figure 3. The Distribution of primary studies in the span of years

From mare analysis and our readings of the selected articles we found that 60% of the articles are industrial reports and about 20% are case study research, while others constitute the remaining 20%. However, the distinction between case study research and industrial reports, by considering the study was planned empirically and been carried out with clear research goals and questions alongside with metrics/measure (the GQM approach), most of the studies are dealing with software process estimation and on process improvements. However in either case we focused on the context/scope, the specific problems and experiences, talking about success stories, causes of failure or factors, other challenges faced, and suggestions on improving their GQM for the betterment and improvement of quality in their products. There are actually few of the articles that compared the software process measurement, estimation, and improvement with available tools, models or practices [3, 9, 21, 22]. The reason might be that some studies many studies are lacking clear concept description like: the organization involved, business they do, the size, and skills of its staff, and lastly productivity, robustness and stakeholders. In addition, what are the measurement details, time frame, methodology for analysis (automated or manual), relationship between the measurement process and the activities improvement, Hence, this is a serious drawback on making software process improvement a success. There is a need for standardization and definition of steps and qualities expected and targeted to boost effective and reliable software products resulting from the improved standardization of measurement and estimation processes. You may also find some studies lacking a clear and defined research approach, method or technique, but personal structuring or a mere reporting (no tools, no charts, tables or any statistical representation) [5, 7]. Now, the

answers will be presented to our research questions, which are the purpose of this SLR, and based on the studies with mere analysis on the collected journal and conference articles and papers respectively.

RQ1: What techniques, the methods and models available in the literature/publications, concerns: software evaluation, software improvement process and software estimation?

RQ2: What are the limitations of the existing techniques?

To outline the techniques, methods/approach, purpose, measurement entity focus, area where it is applied, and the limitations/challenges. We deemed it necessary to have our findings result presented in a tabular format, bearing the above listed stuffs as the headings/titles of respective columns. In either case we tried to cover research questions RQ1 and RQ2 in the same Table, and the table is in such a way that it is classified based on: Software Process Improvement, Software Estimations and however, it is too large to be placed in here, therefore, it will be found in the appendix section.

Toward discuss on software Process Improvement: Software process Improvement (SPI): Here there are a lot of measurements and planning frameworks/models for software process improvement, example: The QIP framework/model which defines an approach admitting the use of goal question metric, in the improvement of software development process which are applicable to organizations and projects. Another one is the AMI model proposed with similar objectives as the QIP but specific to CMMI. The approach comprises of four basic steps including [2, 15].

- a. Assess [analyzing problem assessment with relevant goal definition]
- b. Analyzing [what are the GQM for the identified problem?]
- c. Metric [data collection and analysis] lastly,
- d. Improvement [what are the comparison between the planned and the actual [1]]”.

However, OPEN approach was proposed to effectively utilize the GQM and the CMMI models. See the Table.

RQ3: What are the recommendations from the literature suggested to improve software improvement and estimation process?

Furtado and Oliveira (2012), recommend development of framework software acquisition and related services, in-line with the recommendations and good practices of the existing models for process improvement. Matalonga & Feliu (2011), recommended that training is an effective tool for institutionalizing and improving software development process. Cheng, et.al, (2012), recommended that the SE development process should focus more on user/customer satisfaction by involving the users in the process and to pay attention on ROI (return on investment). Elham, et al. (2014)

suggested that classification of software projects should “be based on underlying attributes to localize the development effort estimation process in a widely used model called analogy-based estimation (ABE) [33].

RQ4: What factors lead to failure and how to avoid them in future research?

There are so many factors and reasons leading to software failure, which are at the result of the process failure. The following are some of the factors obtained from the literatures:

- a. Unrealistic or un articulated process and in turn project goals,
- b. In accuracy in the estimation of required resources,
- c. In appropriate definition of system and process requirement,
- d. Lack of proper documentation of achievement, progress and status information,
- e. Lack of defined measure for managing risks
- f. Poor communication between users and developers, that is the development process does not adopt user-centered approach,
- g. The use of immature or outdated technology,
- h. The inability of the team to handle the projects complexity, in other words: lack of defined metrics for handling project complexity,
- i. The adoption of sloppy development practices,
- j. Poor project management,
- k. Politics from the stakeholders and personal of interest, and lastly,
- l. Commercial and business pressures.

These are the most common factors leading to software process failure, and on the other hand, avoiding all the aforementioned could lead a correct, accepted software development process improved, and will yield a qualitative and correct product. There are many other factors that are not covered here. Whenever software fails it’s grossly affect the organization, individual, company, country or any field where it was used. It leads to loss of resources, if not loss of lives, depending on the area of its application.

Conclusion

In conclusion, this study on software process improvement and estimation will address the issues regarding success stories, factors and barriers for assisting companies towards implementing software. Therefore, this work has result in identifying the areas were need to improve quality and productivity of the product. Hence, all the stated objectives were achieved through the formulation of the research questions. In which systematic literature review was conducted with about 47 selected based studies were observed. It was realized that there is a high demand to develop

more techniques, which are very specific to software process improvement and a tool for estimation and evaluation of software improvement process.

References

- [1] Kitchenham, B., and Ebse, C., (2007).“Guidelines for performing Systematic Literature Reviews in Software Engineering Executive summary.
- [2] Kitchenham, B., Brereton, O.P., Budgen, D.,Turner, M., Bailey, J. and Linkman, S (2009): “Systematic literature reviews in software engineering – A systematic literature review,” *Inf. Softw. Technol.*, vol. 51(1): 7–15.
- [3] Lakshmi. P and Maheswari, T.L (2016): “An effective rank approach to software defect prediction using software metrics, *10th Int. Conf. Intell. Syst. Control*, pp. 1–5.
- [4] Zhang, F., Hassan, A.E., Mcintosh, S., and Zou, Y. (2016): “The Use of Summation to Aggregate Software Metrics Hinders the Performance of Defect Prediction Models,” vol. XX, no. Xx, pp. 1–16.
- [5] Tahir. T, Rasool, G. and Gencel, C. (2016) “A systematic literature review on software measurement programs,” *Inf. Softw. Technol.*, vol. 73, 101–121.
- [6] Nguyen-Cong, D. and Tran-Cao, D. (2013)“A review of effort estimation studies in agile, iterative and incremental software development,” *Proc. - RIVF Int. Conf. Comput. Commun. Technol. Res. Innov. Vis. Futur. RIVF* pp. 27–30.
- [7] Kumar, D. and Kumari, M. (2015) “Component Based Software Engineering: Quality Assurance Models, Metrics,” *4th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir.*, pp. 1–6.
- [8] Gürbüz, H. G. (2016) “Software Metrics for Green Parallel Computing of Big Data Systems.”
- [9] Lee, T., Nam J., Han D., Kim, S. and In, H. (2016).“Developer Micro Interaction Metrics for Software Defect Prediction,” *IEEE Trans. Softw. Eng.*, vol. 99, 1–1
- [10] Moreno-león J, Carlos J, and Román-gonzález M.,(2016) “Comparing Computational Thinking Development Assessment Scores with Software Complexity Metrics,” *2016 IEEE Glob. Eng. Educ. Conf.*, no. April, pp. 1040–1045
- [11] Unterkalmsteiner M., Gorschek T., Moinul Islam A. K. M., Cheng C. K., Permadi R. B., Feldt R., Member S., Gorschek T., Islam A. K. M. M., Cheng C. K., Permadi R. B., and Feldt R., (2012) “Evaluation and Measurement of Software Process Improvement — A Systematic Literature Review,” *IEEE Trans. Softw. Eng.*, vol. 38, no. 2, pp. 398–424,
- [12] Unterkalmsteiner M., Gorschek T., Islam a K. M. M., Cheng C. K., Permadi R. B., and Feldt R., (2011)“Evaluation and Measurement of Software Process Improvement - A Systematic Literature Review,” *IEEE Trans. Softw. Eng.*, vol. X, pp. 1–29,
- [13] Kitchenham B., Pretorius R., Budgen D., Brereton O. P., Turner M., Niazi M., and Linkman S., (2010)“Systematic literature reviews in software engineering – A tertiary study,” *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805,.
- [14] Kuhrmann M. and Diebold P., (2016) “How Does Software Process Improvement Address Global Software Engineering ?,”.
- [15] Santana C., Cerviño M., Caetano D., and Gusmão C., (2010))“Software Process Improvement in Agile Software Development: A Systematic Literature Review,” *Work. Bras. Métodos Ageis*, pp. 30–42,.
- [16] Schramm J., Dohrmann P., Rausch A., and Ternite T., (2014) “Process model engineering lifecycle: Holistic concept proposal and systematic literature review,” *Proc. - 40th Euromicro Conf. Ser. Softw. Eng. Adv. Appl. SEAA 2014*, pp. 127–130,.
- [17] Dybå T. and Dingsøy T., (2008.)“Empirical studies of agile software development: A systematic review,” *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859,
- [18] Andrian R., Hendradjaya B., and Sunindyo W. D., (2016) “Software Assessment Model Using Metrics Products for e-Government In The G2B Model,” *4th Int. Conf. Inf. Commun. Technol. Bandung*, vol. 4, no. c,.
- [19] Bhatia S., (2015)“International Journal of Science and Engineering Estimation of Software Development Effort with Machine Learning Approaches : A Review,” vol. 1, no. 1, pp. 23–26,.
- [20] Hirama K., (2016) “Software Complexity Analysis Based on Shannon ´s Entropy Theory and C & K Metrics,” vol. 14, no. 5, pp. 2485–2490,.
- [21] Zhao F., Nian G., Jin H., Yang L. T., and Zhu Y., (2015) “A Hybrid eBusiness Software Metrics Framework for Decision Making in Cloud Computing Environment,” pp. 1–11,.
- [22] Dutta P., Lehmann C., Odedra D., Singh D., and Pohl C., (2015)“Tracking and Quantifying Developmental Processes in C. elegans Using Open-source Tools,” *J. Vis. Exp.*, no. 106, p. e53469,.
- [23] Ali A.,(2016)“Prediction using different Software Metrics and Particle Swarm Optimization (PSO),”.
- [24] Bilgaiyan S., Mishra S., and Das M.,(2016) “A Review of

- Software Cost Estimation in Agile Software Development Using Soft Computing Techniques,” *2016 2nd Int. Conf. Comput. Intell. Networks*, pp. 112–117.
- [25] Popovic J., Bojic D., and Korolija N., (2015)“Analysis of task effort estimation accuracy based on use case point size.,” *IET Softw.*, vol. 9, no. 6, pp. 166–173.
- [26] Dejaeger K., Verbeke W., Martens D., and Baesens B., (2012)“Data mining techniques for software effort estimation: A comparative study,” *IEEE Trans. Softw. Eng.*, vol. 38, no. 2, pp. 375–397.
- [27] Rath S. K., Acharya B. P., and Satapathy S. M., (2016)“Early stage software effort estimation using random forest technique based on use case points,” *IET Softw.*, vol. 10, no. 1, pp. 10–17.
- [28] Xiao X. and Dohi T., (2013) “Estimating software intensity function based on translation-invariant poisson smoothing approach,” *IEEE Trans. Reliab.*, vol. 62, no. 4, pp. 930–945.
- [29] Shahzad M. and Liu A. X., (2014)“Fast and Accurate Estimation of RFID Tags,” *Networking, IEEE/ACM Trans.*, vol. PP, no. 99, p. 1.
- [30] Khatibi V. Bardsiri and Khatibi E., (2015)“Model to estimate the software development effort based on in-depth analysis of project attributes,” *IET Softw.*, 2015, Vol. 9, Iss. 4, vol. 9, no. 4, pp. 109–118.
- [31] Mittas N., Papatheocharous E., Angelis L., and Andreou A. S., (2015) “Integrating non-parametric models with linear components for producing software cost estimations,” *J. Syst. Softw.*, vol. 99, pp. 120–134.
- [32] González-Ladrón-de-Guevara F, Fernández-Diego M, and Lokan C, (2016) “The usage of ISBSG data fields in software effort estimation: A systematic mapping study,” *J. Syst. Softw.*, vol. 113, pp. 188–215.
- [33] [33] Afzal W., Alone S., Glocksien K., and Torkar R.,(2016)“Software test process improvement approaches: A systematic literature review and an industrial case study,” *J. Syst. Softw.*, vol. 111, pp. 1–33.
- [34] Idri A, Hosni M, and Abran A, (2016) “Improved Estimation of Software Development Effort Using Classical and Fuzzy Analogy Ensembles,” *Appl. Soft Comput.*, vol. 49, pp. 990–1019.
- [35] Baumann F, Milutinovic A, and Roller D, (2016) “Software Engineering Inspired Cost Estimation for Process Modelling,” *Int. J. Soc. Behav. Educ. Econ. Bus. Ind. Eng.*, vol. 10, no. 2, pp. 549–559.
- [36] Tanveer B, Guzmán L, and Engel U. M, (, 2016) “Understanding and improving effort estimation in Agile software development,” *Proc. Int. Work. Softw. Syst. Process - ICSSP '16*, pp. 41–50.
- [37] Al Bulushi, (2015)“Governing Public - Private Tensions of Open Source Software Communities in the Higher Education Sector,” pp. 1–6.
- [38] Garces G. A, Bonjour E, and Rakotondranaivo A, (2016) “New product acceptability evaluation and improvement model with knowledge reuse**This work is financed by a grant from the French Ministry of Higher Education and Research (Ministere de l’Enseignement superieur et de la Recherche, MESR).,” *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1104–1109.
- [39] Rahmani H, Sami v, andKhalili A, (2016)“CIP-UQIM: A unified model for quality improvement in software SME’s based on CMMI level 2 and 3,” *Inf. Softw. Technol.*, vol. 71, pp. 27–57.
- [40] Kumar C. and Yadav D. K, (2015)“A Probabilistic Software Risk Assessment and Estimation Model for Software Projects,” *Procedia Comput. Sci.*, vol. 54, pp. 353–361.
- [41] Galinac Grbac T, Car Z, and Huljenic D, (2012)“Quantifying value of adding inspection effort early in the development process: a case study,” *IET Softw.*, vol. 6, no. 3, p. 249.
- [42] Burgess C. J, Lefley M, and Le M (2001)“Can genetic programming improve software effort estimation? A comparative evaluation,” *Inf. Softw. Technol.*, vol. 43, no. 14, pp. 863–873.
- [43] Wittig G. and Finnie G, (1997) “Estimating software development effort with connectionist models,” *Inf. Softw. Technol.*, vol. 39, no. 7, pp. 469–476.
- [44] Dolado J. J, Rodriguez D, Harman M, Langdon W. B, and Sarro F, (2015)“Evaluation of estimation models using the Minimum Interval of Equivalence,” *Appl. Soft Comput. J.*, vol. 49, pp. 956–967.
- [45] Elnaffar S, (2016) “Using Software Metrics to Predict the Difficulty of Code Writing Questions,” no. April, pp. 513–518.
- [46] Popović B, Balota A, and Strujić D, (2016) “Visual representation of predictions in software development based on software metrics history data,” pp. 370–375.
- [47] Sharma A. and Kushwaha D. S, (2012)“Estimation of Software Development Effort from Requirements Based Complexity,” *Procedia Technol.*, vol. 4, pp. 716–722.
- [48] Choudhari J. and Suman U, (2012)“Story Points Based Effort Estimation Model for Software Maintenance,” *Procedia*

- Technol., vol. 4, pp. 761–765..
- [49] Lopez-Novoa U, Sáenz J, Mendiburu A, Miguel-Alonso J, Errasti I, Esnaola G, Ezcurra A, and Ibarra-Berastegi G, (2015)“Multi-objective environmental model evaluation by means of multidimensional kernel density estimators: Efficient and multi-core implementations,” *Environ. Model. Softw.*, vol. 63, pp. 123–136..
- [50] Tsumoto S. and Hirano S, (2016)“Data Mining Oriented Software Quality Estimation,” *Procedia Comput. Sci.*, vol. 91, no. Itqm, pp. 1028–1037.
- [51] Panda A, Satapathy S. M, and Rath S. K, (2015) “Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points,” *Procedia Comput. Sci.*, vol. 57, pp. 772–781..
- [52] Darwish S. M, (2016)“Software test quality rating: A paradigm shift in swarm computing for software certification,” *Knowledge-Based Syst.*, vol. 110, pp. 167–175.
- [53] Bin Ali N, Petersen K, and De França B. B. N, (2015)“Evaluation of simulation-assisted value stream mapping for software product development: Two industrial cases,” *Inf. Softw. Technol.*, vol. 68, pp. 45–61..
- [54] Huang J, Li Y.-F, and Xie M, (2015)“An empirical analysis of data preprocessing for machine learning-based software cost estimation,” *Inf. Softw. Technol.*, vol. 67, pp. 108–127.
- [55] Banerjee T, Gadou M, and Ranka S, (2016) “Sustainable Computing : Informatics and Systems A genetic algorithm based approach for multi-objective hardware / software co-optimization,” *Sustain. Comput. Informatics Syst.*, vol. 10, pp. 36–47.
- [56] Xiong B, Yang K, Zhao J, Li W, and Li K, (2016)“Performance evaluation of OpenFlow-based software-defined networks based on queuing model,” *Comput. Networks*, vol. 102, pp. 172–185..
- [57] Marculescu B, Feldt R, Torkar R, and Poulding S, (2015)“An initial industrial evaluation of interactive search-based testing for embedded software,” *Appl. Soft Comput. J.*, vol. 29, pp. 26–39..
- [58] Bansal A, (2017)“Computer Languages , Systems & Structures Empirical analysis of search based algorithms to identify change prone classes of open source software,” *Comput. Lang.*, vol. 47, pp. 211–231.
- [59] Wei J, Ji Y, and Yu C, (2014) *Improvement of Software Defined Radio based RSSI localization with bias Abstract* : 47(3) IFAC.
- [60] Oyamada M. S, Zschornack F, and Wagner F. R, (2008) “Applying neural networks to performance estimation of embedded software,” *J. Syst. Archit.*, 54(1–2): 224–240.
- [61] Gonzalez-Huerta J, Insfran E, Abrahão S, and Scanniello G, (2015) “Validating a model-driven software architecture evaluation and improvement method: A family of experiments,” *Inf. Softw. Technol.*, 57(1): 405–429.
- [62] Seo Y. S, Bae D. H, and Jeffery R, (2013) “AREION: Software effort estimation based on multiple regressions with adaptive recursive data partitioning,” *Inf. Softw. Technol.*, 55(10): 1710–1725.
- [63] Minku L. L. and Yao X, (2013)“Ensembles and locality: Insight on improving software effort estimation,” *Inf. Softw. Technol.*, 55(8): 1512–1528.
- [64] Sulayman M, Urquhart C, Mendes E, and Seidel S, (2012)“Software process improvement success factors for small and medium Web companies: A qualitative study,” *Inf. Softw. Technol.*, vol. 54(5): 479–500.
- [65] Vavpotic D. and Bajec M, (2009) “An approach for concurrent evaluation of technical and social aspects of software development methodologies,” *Inf. Softw. Technol.*, 51(2): 528–545.
- [66] Morgenshtern O, Raz T, and Dvir D, (2007)“Factors affecting duration and effort estimation errors in software development projects,” *Inf. Softw. Technol.*, (49): 827–837.
- [67] Araújo R. D. A, Oliveira A. L. I, Soares S, and Meira S, (2012)“An evolutionary morphological approach for software development cost estimation,” *Neural Networks*, vol. 32, pp. 285–291.
- [68] Senderovich A, Weidlich M, Yedidsion L, Gal A, Mandelbaum A, Kadish S, and Bunnell C. A, (2016)“Conformance checking and performance improvement in scheduled processes : A queuing-network perspective,” *Inf. Syst.*, vol. 62, pp. 185–206.
- [69] Lung C, Zhang X, and Rajeswaran P, (2016)“The Journal of Systems and Software Improving software performance and reliability in a distributed and concurrent environment with an architecture-based self-adaptive framework,” *J. Syst. Softw.*, vol. 121, pp. 311–328.