Hasil.7. Implementation

by Herman Yuliansyah

Submission date: 12-Aug-2023 06:36PM (UTC+0700)

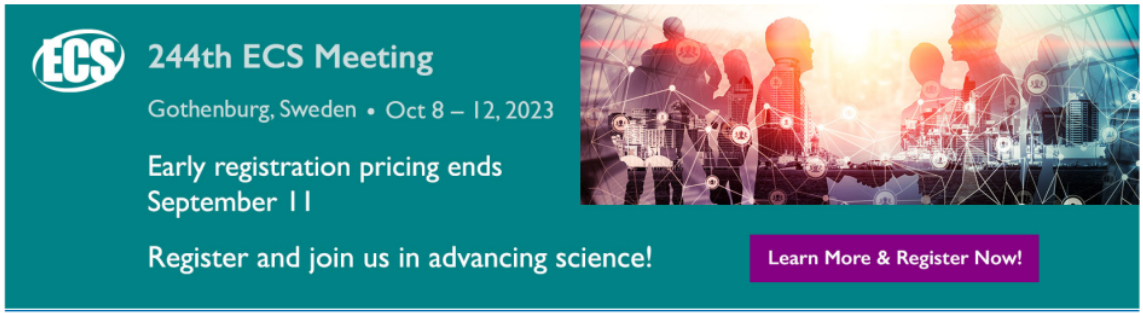
Submission ID: 2144781374

File name: 7. Implementation of use case point as software effort estimation in Scrum Framework.pdf (970.22K)

Word count: 4067

Character count: 21311

Implementation of use case point as software effort estimation in Scrum Framework



Implementation of use case point as software effort estimation in Scrum Framework

H Yuliansyah[1], S N Qudsiah[2], L Zahrotun[3] and I Arfiani[4]
[1,2,3,4] Department of Informatics, Universitas Ahmad Dahlan
Jl. Prof Soepomo, Janturan, Umbulharjo, Yogyakarta, DI. Yogyakarta, Indonesia

[1]herman.yuliansyah@tif.uad.ac.id

Abstract. Estimation is a mechanism to estimate and provide a value for software development projects. Use Case Point (UCP) is a method of software estimation effort in software development based on Use Case model. Scrum is a framework for software development that can solve complex problems and always change. The aims of this study are finding the implementation of UCP as a method of estimating efforts software development with Scrum framework. The methodology of this research is identifying the estimation process with UCP and examine collaboration with Scrum Framework. The results of UCP and Scrum Framework collaborations will be simulated using a web-based project management system. The result of this research is an implementation of UCP and Scrum can be done by associate the User Stories on the Product Backlog with the Use Case on UCP and by modifying User Story attribute to Functionality and adding transaction attribute when defining the User Storie on Product Backlog.

1. Introduction

Estimation is an anticipation of the resource that requires to finishing the software development by observing the environment and the method [1]. With software estimation, we can find software size estimates, software development effort, software development cost and software development schedule for the software project in an environment by using defined methods, tools, and techniques [1]. A good software estimation method will produce optimal resource by utilization and improve the quality of the final products to reach higher level of customer satisfaction [2].

Use Case Point (UCP) is a method to estimate the required effort (Estimation Effort) in completion of software development [3][4]. Modeling software with Use Case is the basis of Estimation Effort used in UCP method [5]. Estimation Effort calculation is multiplication between UCP value and value of Effort Rate (ER) [6]. The value of ER is a prediction value that calculated based on previous project statistics and this is an important value to identify the number of human resources for a project [6]. Several previous studies have defined ER values in UCP. Karner proposed an ER value of 20 Man Hours per UCP [6]. Schneider & Winters has proposed a value of 20, 24, 36 Man Hours Per UCP [7]. Clemmons has proposed a value of 18 Man Hours per UCP [4]. Ochodeck has proposed a value of 4 to 25 Man Hours per UCP [3]. Subriadi has proposed a value of 8.2 Man Hours per UCP [8]. Sholiq has proposed a value of 4.41 Man Hours per UCP [9].

Scrum is a framework with a very simple process to manage complex projects [10]. Besides Extreme Programming (XP) and Kanban, Scrum is also included in Agile Software Development (ASD) [11]. Some methods of effort estimation on Scrum framework include Expert Judgment, Planning Poker [12][13][14], Story Point [15][16], Bucket System [17], T-shirt Size [17], Relative Mass Valuation [17], and Dot Voting [17].

Planning poker introduced by James Grenning, described further by Cohn, is a game that team members can play to estimate the tasks that exist in software development [13][18][19]. Planning poker uses cards to estimate stories with values of 1, 2, 3, 5, 7, 10. This value represents the number of days required to be ideal for programming. The poker planning estimation results from experts will be more

accurate with the actual effort required and will be more accurate than the combination of statistics and individual estimates [14].

Story Point is an effort estimation based on the size of a task to represent consensus between team and task effort [16][20]. A point is assigned to each Story Point. This point is relative. This is because a two-pointed story will assume to take twice the effort of a story. Story Points can be assigned based on the effort involved, complexity and inherent risks of developing features [15].

Bucket system is a way to estimate the number of tasks in a group of people in small to medium quantities and done quickly. T-shirt size is an estimation method that uses non-numeric estimation technique/not with numbers, this is done to prevent confusion in interpreting the number of hours is required for an attempt. T-shirt size will estimate a task with the title extra-small, small, medium, large, extra-large, or double extra-large. Relative mass valuation is a quick way to get through the task and estimate everything because it is related to each other. To use this approach, first, write a card for each task. Then, you have to create a large table to move tasks each other. Dot Voting is a technique that allows participants to choose preferences among a set of items. This is done by placing a dot on an item higher than the other items. Items, with higher priority dots, are higher than items with fewer dots. [17].

A variety of estimation methods have been applied in the context of ASD, although most of these methods do not yield good predictive accuracy values. However, Story Point and Use Case Point is a technique approaching with software requirements specification method such as user stories and use case. This specification is the most commonly used [12]. Based on the description, the story point is a method that has been widely implemented in Scrum Framework minus Use Case Point [20]. Therefore, he novelty in this article is how to implement Use Case Point with Scrum Framework. This is an improvement to utilize the software effort estimation in software development.

2. Method

This research will be done in 3 stages. First stage is to identify the estimation process with UCP. This identification is done by reviewing the related literature in the estimation process with UCP. This identification will know the components required in the estimation process with UCP.

The second stage is to examine the mechanism of implementation of Scrum framework to find UCP and Scrum *integration*. The result of this stage is to find the part that can be an intersection and can be associated with UCP and Scrum. These results will be the system needs to be developed into a simulation application.

The last stage is to develop a simulation application. This simulation application is a web-based project management system. This application will give an overview of the implementation process of UCP as Software Effort Estimation method in Scrum Framework.

3. Results and Discussion

The UCP estimation process begins by measuring system functionality based on the use case model in a count called Unadjusted Use Case Point (UUCP). To calculate UUCP is done by giving the value of each actor on Use Case and give weighting to each Use Case based on the complexity of the function to be implemented from Use Case. The results of the actor rating are simple, average or complex values with reference to Table 1 and each Use Case is the reference to Table 2.

Table 1. Assessment of Weighted Actor [9].

Complexity	Definition	Weight
SIMPLE	If it represents with application programming interface.	1
AVERAGE	If it has interaction through a protocol OR human with a line terminal.	2
COMPLEX	If it interaction through GUI.	3

Table 2. Assessment of Weighted Use Cases [9].

Complexity	Definition	Weight
SIMPLE	If it has 3 or less transactions.	5
AVERAGE	If it has 3 to 7 transactions	10
COMPLEX	If it has more than 7 transactions	15

The next estimation process is to calculate the Technical Complexity Factor (TCF) and the Environmental Factor (EF) that will affect the completion of the software process. EF will help to estimate how efficient the project. The TCF component will contribute to the complexity shown in Table 3 and the EF components that will contribute to the efficiency shown in Table 4.

Table 3. Assessment of TCF - Technical Complexity Factor [9].

F_i	Factors Contributing to Complexity	W_i
F_1	Distributed systems.	2
F_2	Application performance objectives, in either response or throughput.	1
F_3	End user efficiency (on-line).	1
F_4	Complex internal processing.	1
F_5	Reusability, the code must be able to reuse in other applications.	1
F_6	Installation ease.	0.5
F_7	Operational ease, usability.	0.5
F_8	Portability.	2
F_9	Changeability.	1
F_{10}	Concurrency.	1
F_{11}	Special security features.	1
F_{12}	Provide direct access for third parties	1
F_{13}	Special user training facilities	1

Table 4. Assessment of EF - Environmental Factor [9].

F_i	Factors Contributing to Complexity	W_i
F_1	Familiar with Objectory	1.5
F_2	Part time workers	-1
F_3	Analyst capability	0.5
F_4	Application experience	0.5
F_5	Object oriented experience	1
F_6	Motivation	1
F_7	Difficult programming language	-1
F_8	Stable requirements	2

The value of UCP can be calculated by multiplying between UUCP, TCF, and EF values. After that multiply, the number of UCP with Mean Resources needed per UCP (MR) or Effort Rate (ER).

Based on the calculation of the software development estimation process with UCP, it can be seen that from the process UUCP calculation is related to the analysis and system design with UML Use Case. This relation is the value of Weighted actors derived from the level of complexity of the actor type and Weighted use cases obtained from the level of complexity of the number of transactions that may arise from a Use Case. The calculation of UUCP will be focused later on in integration with Scrum.

To find the integration of UCP with Scrum, the first step is to examine the process model or process sequence of the Scrum framework. Figure 1 describes the Scrum framework process model. In Figure 1, there are several activities that occur in Scrum: determining product backlog, determining sprint backlog, implementation (coding) and daily Scrum, sprint review, and sprint retrospective [21].

Product Owner defines the product backlog [22]. The Product Owner arranges and lists all the goals that will be produced in the software. The List of Product Backlog is sorted by priority. Product Backlog is not limited and will be the main reference or source of all the features that must exist in the

software to be developed. Product Backlog will be used during Sprint Planning Meeting to determine the tasks that will be done as Sprint Backlog.

The determination of the Sprint Backlog is determined by reference to a predefined Product Backlog. The Developer Team implements the Sprint Planning Meeting to determine the Sprint Backlog to be performed by the Developer Team. In Sprint Planning Meeting also determine the start time and estimated time of each Sprint work by the Developer Team in hours and determine Sprint Goal in this application. Sprint Planning Meeting will generate Sprint Backlog, which will be ready to be completed.

After Sprint Planning Meeting is completed and generates Sprint Backlog and Sprint Goal, the Developer Team will complete the Sprint Backlog that has been obtained from the Sprint Planning Meeting. During the software development process, the Developer Team meets every day for a maximum of 15 minutes to deliver and discuss the performance of Sprint Backlog that has been done and what will be done and the constraints that occur during the completion of the performance. Daily Scrum is done every day and at the same hour. Scrum Master and Developer Team attended daily Scrum.

Sprint Review is done at the end of Sprint Backlog completion. Sprint Review activity is carried out to review work that has been done by the Developer Team within a certain time. The Sprint Review was attended by the Scrum and Stakeholder Teams and was held for a maximum of four hours. The Scrum teams and Stakeholder collaborate to discuss Sprints that have been worked on and completed by the Developer Team. Sprint Retrospective is done to allow the Scrum Team to review itself and make plans for future work.

Based on the process can be concluded that the determination of Product Backlog and Sprint Backlog in Sprint Planning Meeting is a planning phase in Scrum, while the work of Sprint Backlog and Daily Scrum is the implementation phase of software development and Sprint Review is the evaluation phase of software development. So in this analysis found that estimation with Use Case Point (UCP) will be able to be integrated in Scrum framework at Product Backlog and Sprint Backlog determination phase.

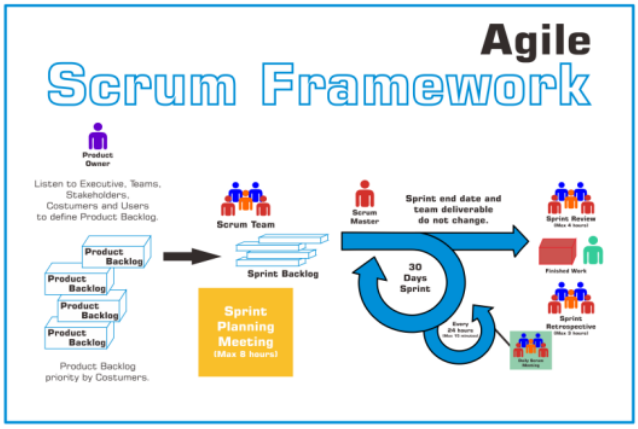


Figure 1. Scrum framework

To clarify the integration process, then I will make a simulation of the integration process. Figure 2 is a simple Use Case example illustrating an application. According to Alan Dennis, Barbara Haley Wixom and David Tegarden, a Use Case is a major part of system functionality and is labelled with a descriptive verb phrase or noun [23].

In Product Backlog determination in Scrum, Product Backlog list can be written by creating User Story. According to Mike Cohn writing User Story can explain a function that will be something of value to the user of the software[18]. Alistair Cockburn in Mike Cohn's book proposes the Use Case writing template among the most commonly used as in Figure 3 [18]. Figure 4 is the equivalent of Writing User Story.

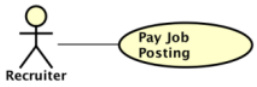
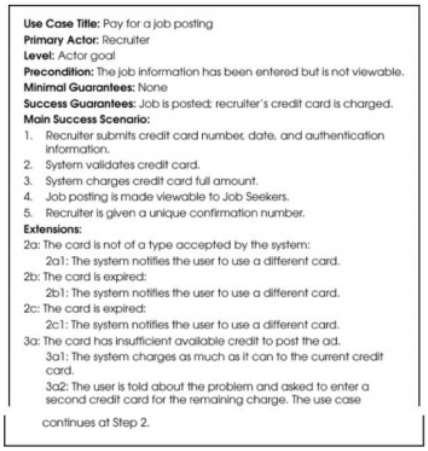


Figure 2. Use case example

Based on Ivar Jacobsen in Mike Cohn's book mentioned that Use Case is a general description of a set of interactions between the system and one or more actors, in which an actor is a user or another system. The Use cases can be written in unstructured text or to adapt to structured templates [18]. Mike Cohn also stated that User Stories is not a Use Case. One of the most obvious differences between User Stories and Use Case is the scope. Both are sized to deliver business value, but Use Stories remain smaller within scope limits. Use Case almost always includes a much larger scope than a User Stories [18].



Use Case Title: Pay for a job posting
Primary Actor: Recruiter
Level: Actor goal
Precondition: The job information has been entered but is not viewable.
Minimal Guarantees: None
Success Guarantees: Job is posted; recruiter's credit card is charged.
Main Success Scenario:
1. Recruiter submits credit card number, date, and authentication information.
2. System validates credit card.
3. System charges credit card full amount.
4. Job posting is made viewable to Job Seekers.
5. Recruiter is given a unique confirmation number.

Extensions:
2a: The card is not of a type accepted by the system:
2a1: The system notifies the user to use a different card.
2b: The card is expired:
2b1: The system notifies the user to use a different card.
2c: The card is expired:
2c1: The system notifies the user to use a different card.
3a: The card has insufficient available credit to post the ad.
3a1: The system charges as much as it can to the current credit card.
3a2: The user is told about the problem and asked to enter a second credit card for the remaining charge. The use case continues at Step 2.

Figure 3. Example of writing Use Case with Alistair Cockburn template [18]

"A Recruiter can pay for a job posting with a credit card."

Figure 4. Example case for writing User Story

Based on the statement, in this article proposes a new method that can be used. While it still complies that Use Case is a larger scope of User Stories and still ensures that Use Case can explain system functionality. Use Case can be used as list of statements in Product Backlog and Sprint Backlog in Scrum framework and able to adapt to Use Case Point especially on Unadjusted Use Case Point (UUCP).

The proposal to be offered is to modify the way of writing Product Backlog and Sprint Backlog by making 3 factors that must be written in this writing. These three factors are Product Backlog and Sprint Backlog must include:

- Actor Name that has been set by actor type based on Weighted actors,
- The Functionality statement that will represent Use Case and will be the goal to be achieved and written on the Product Backlog in the Scrum framework,
- Transaction Statement which will represent a more detailed scope of Functionality. This transaction will also be used as a statement for Sprint Backlog and will be a parameter to calculate the rating of Weighted use cases.

For further details of this proposal will be made a simulation in the form of web applications. This application aims to provide a comprehensive overview of the effect of implementation of the proposed offer.

3.1 Web-based application project management system

The implementation of web-based project management system simulation proposed are:

- Figure 5 is the dashboard page used to view existing projects and to create new projects by selecting the "Project" menu,
- Figure 6 is the page to create a new project. There are several fields in the form of project name, filler position, working hours of the day, project start date and description,
- Figure 7 is the Product Backlog stuffing page. This field is a proposal in this research to integrate UCP and Scrum framework,
- Figure 8 is an entry page for the Technical Complexity Factor. This page related to Table 3,
- Figure 9 is the contents page for Environmental Factor. This page related to Table 4,
- Figure 10 is the calculated page of UUCP, TCF and EF. On this page, user must be filled is the value of the Effort Rate (ER) which is used as a reference in determining the estimated value of UCP,
- Figure 11 is the result of UCP calculation. This value is the final value of the estimated estimate,
- Figure 12 is an estimation graph depicted in the form of a Burndown Chart. This graph is obtained from the calculation of estimation by UCP which has been converted to day unit based on UCP value and project hours.

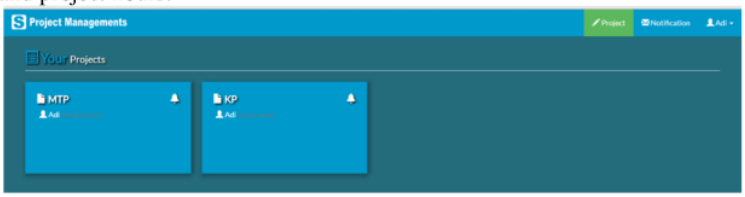


Figure 5. Dashboard page web-based application scrum project management system

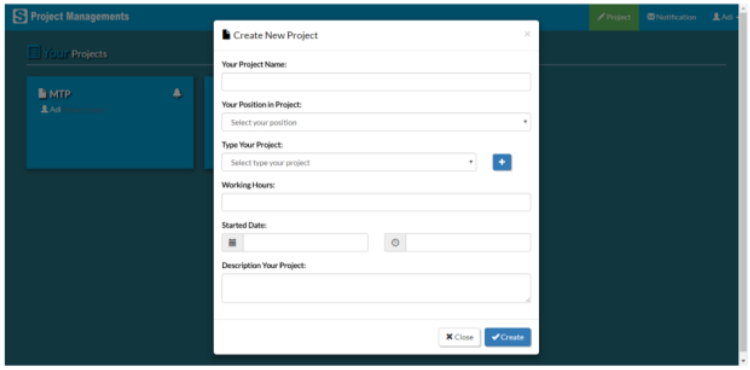


Figure 6. The new project form

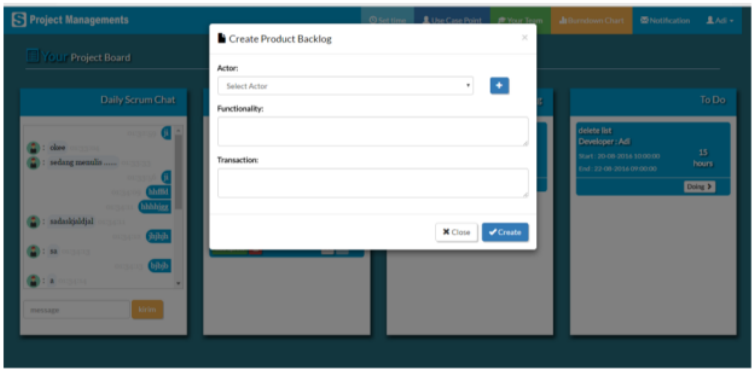


Figure 7. Product backlog form

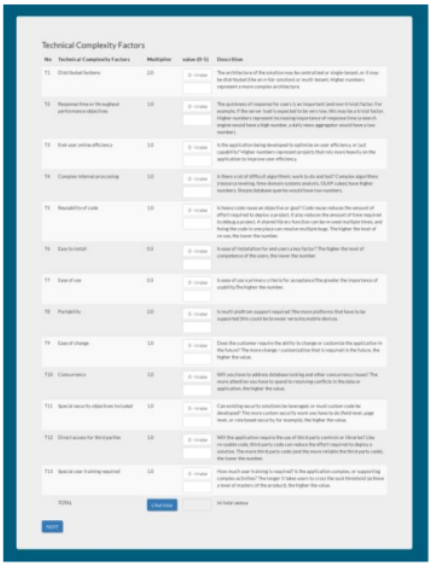


Figure 8. Technical complexity factor form

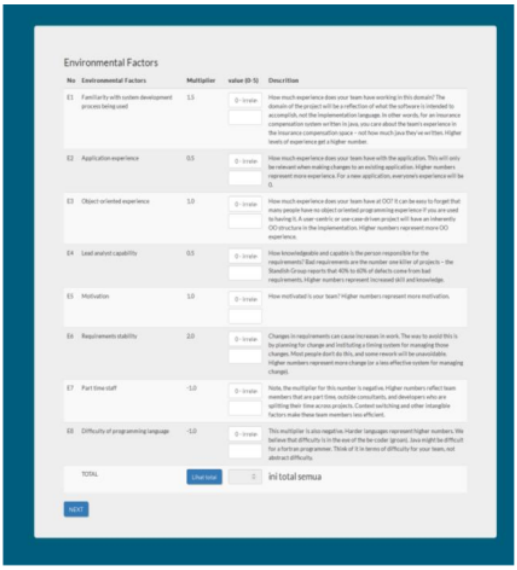


Figure 9. Enviromental factor form

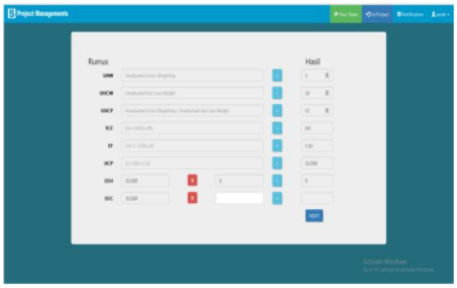


Figure 10. Effort rate form

Figure 11. UCP calculation result

Figure 12. Estimation graph with burndown chart

4. Conclusion

This research proposes the integration of UCP and Scrum framework by associating User Stories on Product Backlog with Use Case on UCP by modifying the User Story attribute into Functionality and adding transaction attribute when defining Product Backlog. Simulations of web-based applications that appear capable of manipulating Product Backlog statements that have been filled to be the components required in calculating UUCP is the assessment of Weighted Use Cases and assessment of Weighted actors.

For future works, this proposal needs to be tested by an expert who has a field of competence in the Scrum framework. This needs to be done because to test that these proposed things do not conflict with the implementation of Scrum framework. The estimates undertaken in this article are global estimates, which means estimates of overall software development work. A further study of the utilization of these estimates can be used to determine the estimates needed to accomplish the tasks in the Sprint Backlog statement. From these results, it is necessary to review the values of the Effort Rate (ER) that can be established and related to the topics of the field of software work to be built.

5. Acknowledgment

Universitas Ahmad Dahlan has supported this research in the scheme Fundamental Research with grants number No: PF-081/SP3/LPP-UAD/IV/2017.

6. References

[1] M. K. Chemuturi, *Software Estimation Best Practices, Tools & Techniques*. USA: J. Ross Publishing, 2009.
[2] M. M. Kirmani and A. Wahid, "Use Case Point Method of Software Effort Estimation : A Review," *Int. J. Comput. Appl.*, vol. 116, no. 15, pp. 43–48, 2015.
[3] M. Ochodek, J. Nawrocki, and K. Kwarciak, "Simplifying effort estimation based on Use Case Points," *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 200–213, 2011.
[4] R. K. Clemmons, "Project Estimation With Use Case Points," *J. Def. Softw. Eng.*, pp. 18–22, 2006.
[5] P. P. Jena, P. Jha, P. P. Jena, and R. K. Malu, "Estimating Software Development Effort using UML Use Case Point (UCP) Method with a Modified set of Environmental Factors Estimating Software Development Effort using UML Use Case Point (UCP) Method with a Modified set of Environmental Factors," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 2742–2744, 2014.
[6] G. Karner, "Resource estimation for objectory projects," *Object. Syst. SF AB*, pp. 1–9, 1993.
[7] G. Schneider and J. P. Winters, *Applying use cases: a practical guide*. Pearson Education, 1998.
[8] A. P. Subriadi, Sholiq, and P. A. Ningrum, "Critical review of the effort rate value in use case point method for estimating software development effort," *J. Theroretical Appl. Inf. Technol.*,

vol. 59, no. 3, pp. 735–744, 2014.

[9] Sholiq, T. Sutanto, A. P. Widodo, and W. Kurniawan, "Effort Rate on Use Case Point Method for Effort Estimation of Website Development," *J. Theroretical Appl. Inf. Technol.*, vol. 63, no. 1, pp. 209–218, 2014.

[10] K. Schwaber and M. Beedle, *Agile software development with Scrum*, vol. 1. Prentice Hall Upper Saddle River, 2002.

[11] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study," *Inf. Softw. Technol.*, vol. 85, pp. 1339–1351, 2017.

[12] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort Estimation in Agile Software Development: A Systematic Literature Review," in *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, 2014, pp. 82–91.

[13] J. Grenning, "Planning poker or how to avoid analysis paralysis while release planning," *Hawthorn Woods Renaiss. Softw. Consult.*, pp. 1–3, 2002.

[14] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," *J. Syst. Softw.*, vol. 85, no. 9, pp. 2086–2095, 2012.

[15] E. Coelho and A. Basu, "Effort Estimation in Agile Software Development using Story Points," *Int. J. Appl. Inf. Syst.*, vol. 3, no. 7, pp. 7–10, 2012.

[16] S. Porru, A. Murgia, S. Demeyer, M. Marchesi, and R. Tonelli, "Estimating Story Points from Issue Reports," *Proc. 12th Int. Conf. Predict. Model. Data Anal. Softw. Eng. - PROMISE 2016*, pp. 1–10, 2016.

[17] V. Singh, "Agile Estimation Techniques," 2016. [Online]. Available: https://www.scrumalliance.org/community/articles/2016/january/agile-estimation-techniques. [Accessed: 27-Aug-2017].

[18] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[19] K. Moløkken-Østvold, N. C. Haugen, and H. C. Benestad, "Using planning poker for combining expert estimates in software projects," *J. Syst. Softw.*, vol. 81, no. 12, pp. 2106–2117, 2008.

[20] E. Ungan, N. Cizmeli, and O. Demirors, "Comparison of functional size based estimation and story points, based on effort estimation effectiveness in SCRUM projects," *Proc. - 40th Euromicro Conf. Ser. Softw. Eng. Adv. Appl. SEAA 2014*, pp. 77–80, 2014.

[21] P. Rola, D. Kuchta, and D. Kopczyk, "Conceptual model of working space for Agile (Scrum) project team," *J. Syst. Softw.*, vol. 118, pp. 49–63, 2016.

[22] H. S. Sverrisdottir, H. T. Ingason, and H. I. Jonasson, "The Role of the Product Owner in Scrum-comparison between Theory and Practices," *Procedia - Soc. Behav. Sci.*, vol. 119, pp. 257–267, 2014.

[23] A. Dennis, B. H. Wixom, and D. Tegarden, *Systems analysis and design: An object-oriented approach with UML*. John Wiley & Sons, 2015.

Hasil.7. Implementation

ORIGINALITY REPORT

35%	30%	18%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Academic Library Consortium Student Paper	8%
2	www.researchgate.net Internet Source	5%
3	ebin.pub Internet Source	4%
4	www.intechopen.com Internet Source	2%
5	repository.uin-suska.ac.id Internet Source	1%
6	iopscience.iop.org Internet Source	1%
7	repository.its.ac.id Internet Source	1%
8	Submitted to Higher Education Commission Pakistan Student Paper	1%
9	Submitted to Bellevue University Student Paper	1%

10	T.L. Rhodes, C. Holland, S.P. Smith, A.E. White et al. "L-mode validation studies of gyrokinetic turbulence simulations via multiscale and multifield turbulence measurements on the DIII-D tokamak", Nuclear Fusion, 2011 Publication	1 %
11	kipdf.com Internet Source	1 %
12	Fahad H.. "Linear Prediction Model for Effort in Programming based on User Acceptance and Revised use Case Point Method", International Journal of Advanced Computer Science and Applications, 2017 Publication	1 %
13	sites.google.com Internet Source	1 %
14	repository.nusamandiri.ac.id Internet Source	1 %
15	Sajjad A. Salih, Baseem A. Aljashaami, Naseer T. Alwan, Mohammed A. Qasim et al. "Numerical investigation of alpha Stirling engine performance based on ideal and actual adiabatic analysis", Journal of Physics: Conference Series, 2023 Publication	1 %
16	repository.uhamka.ac.id Internet Source	1 %

17	Submitted to Far Eastern University Student Paper	<1%
18	eprints.binus.ac.id Internet Source	<1%
19	Meenakshi Saroha, Shashank Sahu. "Tools & methods for software effort estimation using use case points model — A review", International Conference on Computing, Communication & Automation, 2015 Publication	<1%
20	D Kurniadi, A Mulyani, Y Septiana, H Aulawi. "Estimated software measurement base on use case for online admission system", IOP Conference Series: Materials Science and Engineering, 2018 Publication	<1%
21	Renny Sari Dewi, Grandys Frieska Prassida, Sholiq, Apol Pribadi Subriadi. "UCPabc as an integration model for software cost estimation", 2016 2nd International Conference on Science in Information Technology (ICSITech), 2016 Publication	<1%
22	ejournal.st3telkom.ac.id Internet Source	<1%
23	Sarwosri, Muhammad Jabir Al Haiyan, Mujahid Husein, Aditya Putra Ferza. "The development	<1%

	of method of the enhancement of Technical Factor (TF) and Environmental Factor (EF) to the Use Case Point (UCP) to calculate the estimation of software's effort", 2016 International Conference on Information & Communication Technology and Systems (ICTS), 2016 Publication	
24	Suyash Shukla, Sandeep Kumar. "Towards non-linear regression-based prediction of use case point (UCP) metric", Applied Intelligence, 2022 Publication	<1 %
25	aaltodoc.aalto.fi Internet Source	<1 %
26	eejournal.ktu.lt Internet Source	<1 %
27	neverletdown.net Internet Source	<1 %
28	pt.scribd.com Internet Source	<1 %
29	upcommons.upc.edu Internet Source	<1 %
30	www.diva-portal.org Internet Source	<1 %

31	"Sustainable Business Management and Digital Transformation: Challenges and Opportunities in the Post-COVID Era", Springer Science and Business Media LLC, 2023 Publication	<1 %
32	docplayer.net Internet Source	<1 %
33	vdocument.in Internet Source	<1 %
34	www.computerscijournal.org Internet Source	<1 %
35	Safrizal, Harco Leslie Hendric Spits Warnars, Ford Lumban Gaol, Edi Abdurachman. "Use case point as software size measurement with study case of Academic Information System", 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), 2017 Publication	<1 %
36	Mudasir Manzoor Kirmani, Abdul Wahid. "Impact of Modification Made in Re-UCP on Software Effort Estimation", Journal of Software Engineering and Applications, 2015 Publication	<1 %
37	Mudasir Manzoor, Abdul Wahid. "Revised Use Case Point (Re-UCP) Model for Software Effort	<1 %

Estimation", International Journal of Advanced Computer Science and Applications, 2015

Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		