DOI: 10.24996/ijs.2023.64.12.34

ISSN: 0067-2904

Development of the Cost and Time Estimation Factors of the Project Dimension in the Agile Software

Nikhil Govil*, Ashish Sharma
Department of CEA. IET, GLA University, Mathura, India

Received: 14/2/2022 Accepted: 1/12/2022 Published: 30/12/2023

Abstract

Agile methodologies are adopted extensively by many of the software industries as it is flexible in nature as well as can address the required changes in any phase of development. Authentic estimation of the software products is not an easy task as it requires continuous attention of the product owner. Effort and cost can be estimated in a proper manner to ensure the success of the project. In this article, we considered the Scrum-based Agile projects that are developed into several Sprints. We proposed an extension to an existing algorithm, based on a total of 36 success factors; that estimate the development cost and effort required to complete the project. For estimation and computations, we have taken a dataset of 12 projects that are validated through experienced professionals. We also compared our results with the existing approach and it is found that our results are cost-effective even after considering more success factors.

Keywords: Agile Methodologies, Effort Estimation, Cost Estimation, Scrum, Success Factors.

1. Introduction

Agile software development is one of the most favoured and trendy methodologies that is being applied in software industries [1]; in which estimation is the decisive factor. Timely and authentic estimation of any project is essential to the process of software development. Through effective estimation, we can increase the success possibilities of software that is being developed [2] [3]. For a successful project, it is mandatory to deliver the project within the stipulated time frame as well as meet the requirements of the customer. These conditions can be ensured enough when a product owner follows the rigorous estimation exercises. However, in the case of Agile methodologies, it is not so easy to estimate perfectly because as per Agile Alliance [4], customer expectations and requirements must be on the highest priority and addressed at any of the development stages.

There are several frameworks available in Agile methodologies including Scrum, XP, Crystal, Kanban, etc. In this article, we are considering projects that are being developed under the Scrum framework. In the Scrum framework, project development is divided into several iterations or Sprints. Each Sprint follows the requirement elicitation, analysis, development, testing, and delivery of something executable to the end-users. It is obvious that after delivery of some part of the product that is executable in nature, it will be in the maintenance phase. To ensure the proper maintenance, regression testing usually took place to ensure no new error or fault has been raised after adding or modifying any existing features. During regression testing,

* Email: nikhil.govil@gla.ac.in

there are several activities that need to follow. Time consumed in these activities is dependent on the number of user stories.

The main objective of this research is to improve the existing method for estimation by including all possible success factors. The challenge that was faced during this research work was identifying and incorporating all the critical factors that leverage the progress of any Agile-based project. The motivation behind this research was to compute success factors oriented accurate estimation for time, effort, and cost required to complete a project in Agile methodologies as it is the deterministic activity through which any project moves towards success. For this, we proposed an extension in an existing algorithm for success factors-based Agile software estimation technique.

In this article, we considered a data set of 12 individual Agile-Scrum-based projects validated by different industries professionals who had sound working experience on Agile projects. We further applied our algorithm to this data set to compute the estimations. A comparative analysis is also provided as per obtained computations. From the comparisons of the extended algorithm and experimented existing algorithm, it is obtained that our extended algorithm gives less-effort and cost-effective results as compared to the existing considered approach. However, it is also notable that in our extended algorithm we incorporated 36 success factors having different dimensions including Organizational, People, Process, Technical and Project while the existing approach incorporated just 14 success factors having Project and People dimensions. Thus, our results are quite acceptable as there are more success factors included with the reduced effort and cost.

2. Related Work

In this section, we provide a literature survey for some published research works in the domain of Agile software development. In the paper [6], the authors advocate the effective use of machine learning for effort estimation in Agile software development. They also presented the estimation efficiency as 37% using machine learning, 26% using expert judgment, and 21% using algorithmic methods. The most commonly used and implemented attributes in their research were complexity, experience, size, and time.

Wilson Rosa et al. [7] presented a study related to the effort and schedule estimation model for 36 United States Department of Defence projects. These projects were based on agile methodologies. The authors provided effort and schedule estimation mechanisms through their detailed study. The authors identified that the sum of modules and external interfaces are the deciding factors for size measurement for early estimation of projects based on agile methodologies. Further, after including groups of the application domain and peak staff as inputs, the presented model provided improved results in terms of accuracy. In the paper [8], authors discussed estimation models that are associated with Agile–Scrum such as, Planning Poker, T-Shirt Size, Dot Voting, Bucket System, Large/Uncertain/Small method, ordering method, and divide until maximum size or less method. The authors also identified the Agile–Scrum estimation challenges.

In the article [9], authors proposed a multiple linear regression model. This presented model was applied for the identification of the best Agile development model. The authors identified dependent and independent variables and their correlation. Using correlation values, three multiple linear regression models were presented to estimate effort for Agile-based projects. In the research article authored by M. Fernández-Diego et al. [10], authors extracted data from 73 research papers ranging from the year 2014 to 2020. They applied an arithmetic approach for cost factors and presented their results with improvements over the existing approach. For their research, they experimented with six Agile methods including Scrum, XP, and others.

Cláudio Ratke et al. [11], proposed a method for effort estimation based on narrative texts in their research article. In this paper, they applied a unique technique for symbolic analysis of natural language to extract the nouns and verbs from texts. It was further applied to estimate the story points. Rashmi Popli and Naresh Chauhan [12] spotlighted the estimation phase of typical agile software development. Improper estimation in the early phases, is one of the root causes of many other problems including software failure. The authors presented a tool for agile estimation that is based on sprint points. They developed sprint-point based estimation tool in MS-Excel that includes the delay factor in the project's release date. This tool yields effort, cost, and release date estimations.

Authors in article [13] presented a fuzzy method in their research. They took the raw data of facts, figures and produced output as effort estimation. In their research, they used three input variables a user story, team expertise, and complexity to estimate effort. Mohit Arora et al. [14] presented an approach in their research article to estimate the effort required to perform regression testing. The authors proposed a unique mechanism estimating the effort required for regression testing in story-points-based Agile methodologies.

In the research article [15], authors conducted a survey to identify the effort estimation approaches in co-located Agile methodologies as well as how these estimation approaches were distributed in agile environments worldwide. As an outcome of their research, they identified that similar size matrices and effort estimation techniques were being applied globally. However, cost drivers are the crucial factors.

In the present scenario, there is a strong need to consider various other success factors to estimate effort so that we get more optimized and realistic results. Rashmi Popli & Naresh Chauhan [16], proposed a regression testing-based algorithm for effort estimation. They considered 14 project-related and people-related success factors. The authors also examined a case study of a small project of a web-based application and computed its estimations.

3. Extended Algorithm for Agile Software Development Effort Estimation

Tsun Chow and Dac-Buu Cao [5] identified 36 success factors that affect the development process of any Agile-based project to a great extent. There are five dimensions (organizational, people, process, technical, and project) where these 36 success factors are categorized. These dimension-vised factors are listed in Table 1. As per the intermediate COCOMO [17], we are assuming the influencing degree of each factor as nominal having value 1.0 for computation purposes throughout our research. Additionally, regression testing is an essential part of each sprint. Therefore, it was incorporated also while estimating duration, cost, and effort.

Table 1: Dimension wise success factors [5].

Dimension	Factor	Dimension	Factor
People	1. Team with high competence and expertise	Organizational	21. Strong executive support
	2. Team members with great motivation		22. Committed manager
	3. Managers knowledgeable in Agile process		23. Cooperative organizational culture
	4. Adaptive management		24. Face-to-face communication
	5. Coherent, self-organizing teamwork		25. Agile accepted organizations
	6. Good customer relationship		26. Collocation of the whole team
Process	7. Agile-oriented requirement management		27. Facility with proper Agile environment
	8. Agile-oriented project management		28. Reward system appropriate for Agile
	9. Agile-oriented configuration management	Technical	29. Well-defined coding standards up front
	10. Daily face-to-face meetings		30. Pursuing simple design
	11. Honoring regular working schedule		31. Rigorous refactoring activities
	12. Strong customer commitment		32. Right amount of documentation
	13. Customer having full authority		33. Regular delivery of software
Project	14. Project nature being non-life-critical		34. Delivering most important features first
	15. Project scope with emergent requirement		35. Correct integration testing
	16. Projects with dynamic, accelerated schedule		36. Appropriate technical training to team
	17. Projects with small team		
	18. Projects with no multiple independent teams		
	19. Projects with up-front cost evaluation done		
	20. Projects with up-front risk analysis done		

In this article, we presented a data set of 12 different Agile-based projects. For unbiased computation, we have taken three individual sets having four projects in each set for the Low (L), Medium (M), and High (H) level projects. The Unadjusted Value (UV) of the story-point is taken as 1 for low, 2 for medium, and 4 for the high-level project that is on the basis of Geometric Progression (GP) series. We have taken some hypothetical values for our data set as Regression effort per iteration is 10 units, average Velocity Factor (VF) of all 36 factors ranging from 0.95 to 0.97. The data taken in this data set is further validated by different professionals from discrete software industries.

3.1 Extended Algorithm

The presented extension in an existing algorithm consisting of eleven steps to compute Total Estimation and Total Cost estimations. This extended algorithm is shown in Figure 1. This block diagram has new calculations on the method which are extended to the existing approach as it is now considering all 36 success factors along with updated hypothetical values.

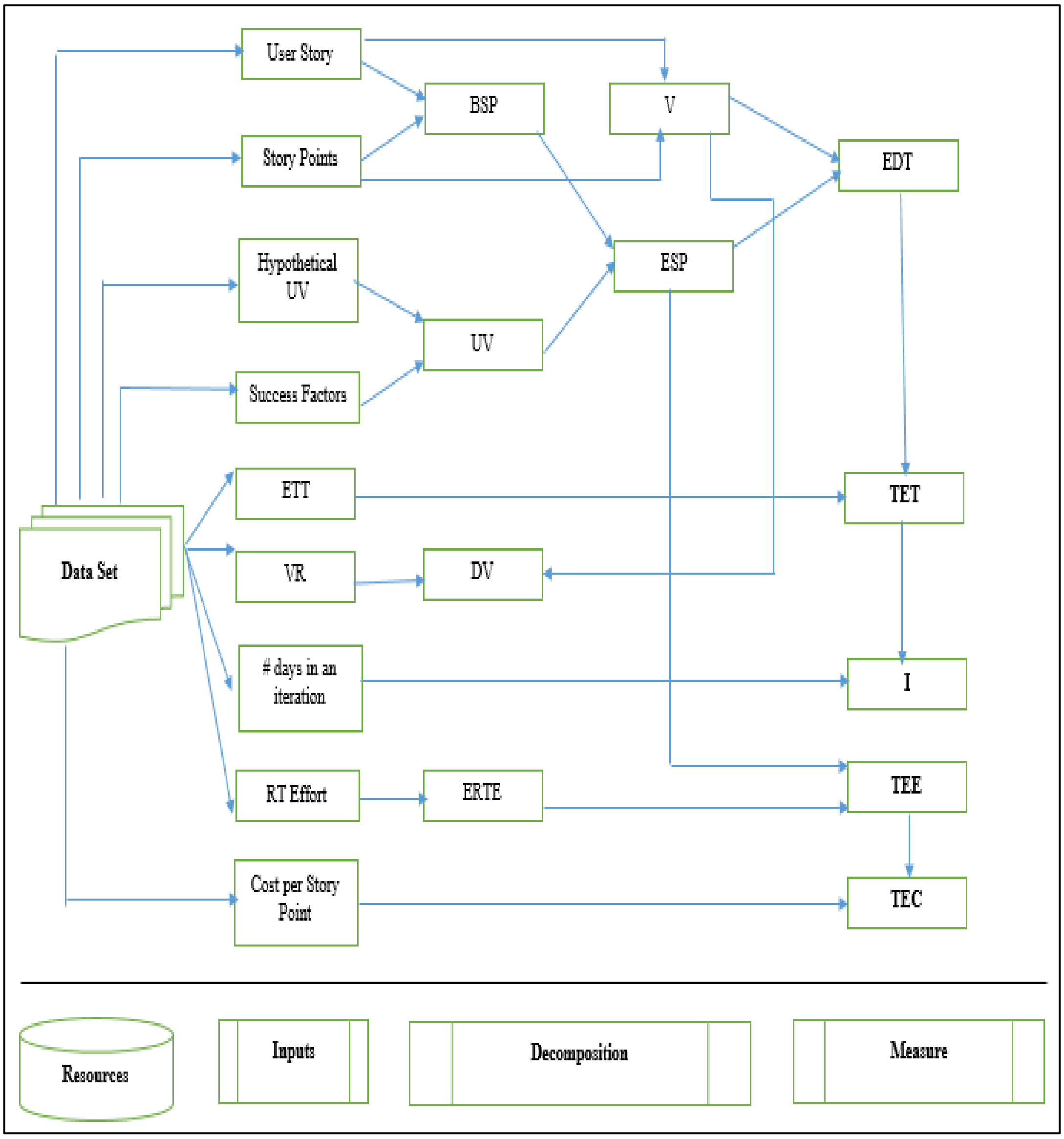

Figure 1: Block diagram of extended algorithm

4. Data Set

Two individual data sets of 12 Agile-based projects ranging from P1 to P12 were used. Table 2 represents a data set whose values are as per the extension in an existing algorithm. In this data set, four projects are from the low, medium, and high factors group each. The various inputs are as follows:

Table 2: Data set inputs for 12 projects as per extended algorithm

Input	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Factors Level as L/M/H	L	L	L	L	M	M	M	M	H	H	H	H
Hypothetical UV as per factor level	1	1	1	1	2	2	2	2	4	4	4	4
No. of User Stories (US)	22	23	21	24	43	45	47	50	86	84	90	87
No. of Story Points in one US	10	11	10	11	8	10	9	9	6	6	7	7

No. of SP completed in an iteration	48	49	51	47	41	41	42	43	33	33	32	31
No. of days in one iteration	13	14	15	14	18	18	20	17	28	28	28	27
No. of working days per month	22	23	22	21	23	23	20	22	22	22	23	21
No. of working hours per day	9	8	8	8	9	9	8	8	8	9	8	9
Regression effort per iteration	10	10	10	10	10	10	10	10	10	10	10	10
Cost per SP	48	51	50	49	62	61	64	65	77	79	76	78
VF: Avg. of VF for all 36 factors	0.96	0.97	0.95	0.96	0.97	0.96	0.96	0.96	0.96	0.97	0.96	0.97
Estimated Testing Time in days	7	8	7	8	12	12	13	13	18	17	19	18

Similarly, Table 3 represents another data set whose values are as per the experimented existing algorithm. In this data set also, four projects are from the low, medium, and high factors group each. Inputs taken in this data set are exactly as per the algorithm proposed by (Rashmi Popli et al. [16] [18] [19]). The various inputs are as follows:

Table 3: Data set inputs for 12 projects as per existing algorithm.

Input	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Factors Level as L/M/H	L	L	L	L	M	M	M	M	H	H	H	H
Hypothetical UV as per factor level	1	1	1	1	6	6	6	6	10	10	10	10
No. of User Stories (US)	22	23	21	24	43	45	47	50	86	84	90	87
No. of Story Points in one US	10	11	10	11	8	10	9	9	6	6	7	7
No. of SP completed in an iteration	48	49	51	47	41	41	42	43	33	33	32	31
No. of days in one iteration	13	14	15	14	18	18	20	17	28	28	28	27
No. of working days per month	22	23	22	21	23	23	20	22	22	22	23	21
No. of working hours per day	9	8	8	8	9	9	8	8	8	9	8	9
Regression effort per iteration	10	10	10	10	10	10	10	10	10	10	10	10
Cost per SP	48	51	50	49	62	61	64	65	77	79	76	78
VF: Avg. of VF for all 36 factors	0.96	0.97	0.95	0.96	0.97	0.96	0.96	0.96	0.96	0.97	0.96	0.97
Estimated Testing Time	15	15	16	16	17	18	18	18	20	20	22	21

5. Illustration of the Extended Algorithm

This section demonstrates the evaluation of Project (P1) as per the extension to an existing algorithm. The values can be computed as follows:

1. BSP=US+SP. So BSP = 22*10 =220
2. UV= hypothetical unadjusted value * 36. So UV= 1*36=36
3. Estimated SP (ESP) as ESP = BSP+0.1*(UV). So ESP=220+0.1(36) =223.6
4. Initial Velocity (V) =SP completed in one Sprint / SP in one US. So V=48/10=4.80
5. DV=V*VR. So, DV=4.80*0.96=4.61 Days

6. EDT=ESP /Velocity (in days). So EDT=223.6/4.61=48.52 Days
7. TET=EDT+ETT. So TET = 48.52+7=55.52 Days
8. No. of iterations (I) =TET/ (iteration time). So No. of iterations=55.52/13=4.27
9. ERTE=Regression effort per iteration*(No. of iterations-1). ERTE=10*(4.27-1) =32.71 SP
10. TEE=ESP+ERT. So TEE= 223.6+32.71=256.31 SP
11. TEC =TEE*Cost per SP. So TEC=256.31*48=$12302.93

Table 4 represents the finding of the various parameters obtained as per the extended algorithm. These findings are as follows:

Table 4: Computation of results as per extended algorithm.

Results	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
BSP	220	253	210	264	344	450	423	450	516	504	630	609
UV	36	36	36	36	72	72	72	72	144	144	144	144
ESP	223.6	256.6	213.6	267.6	351.2	457.2	430.2	457.2	530.4	518.4	644.4	623.4
V	4.8	4.45	5.1	4.27	5.13	4.1	4.67	4.78	5.5	5.5	4.57	4.43
DV	4.61	4.32	4.85	4.1	4.97	3.94	4.48	4.59	5.28	5.34	4.39	4.3
EDT	48.52	59.39	44.09	65.24	70.65	116.16	96.03	99.68	100.45	97.17	146.84	145.12
TET	55.52	67.39	51.09	73.24	82.65	128.16	109.03	112.68	118.45	114.17	165.84	163.12
I	4.27	4.81	3.41	5.23	4.59	7.12	5.45	6.63	4.23	4.08	5.92	6.04
ERTE	32.71	38.13	24.06	42.31	35.91	61.2	44.51	56.28	32.31	30.77	49.23	50.42
TEE	256.31	294.73	237.66	309.91	387.11	518.4	474.71	513.48	562.71	549.17	693.63	673.82
TEC	12302.93	15031.36	11882.89	15185.78	24001.10	31622.35	30381.66	33376.36	43328.30	43384.81	52715.66	52557.60

Similarly, in Table 5, the findings of the same parameters are represented as per values taken in the experimented existing algorithm. These findings are as follows:

Table 5: Computation of results as per existing algorithm.

Results	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**
BSP	220	253	210	264	344	450	423	450	516	504	630	609
UV	14	14	14	14	84	84	84	84	140	140	140	140
ESP	221.4	254.4	211.4	265.4	352.4	458.4	431.4	458.4	530	518	644	623
V	4.8	4.45	5.1	4.27	5.13	4.1	4.67	4.78	5.5	5.5	4.57	4.43
DV	4.61	4.32	4.85	4.1	4.97	3.94	4.48	4.59	5.28	5.34	4.39	4.3
EDT	48.05	58.88	43.63	64.7	70.89	116.46	96.29	99.94	100.38	97.09	146.74	145.03
TET	63.05	73.88	59.63	80.7	87.89	134.46	114.29	117.94	120.38	117.09	168.74	166.03
I	4.85	5.28	3.98	5.76	4.88	7.47	5.71	6.94	4.3	4.18	6.03	6.15
ERTE	38.5	42.77	29.76	47.65	38.83	64.7	47.15	59.38	32.99	31.82	50.27	51.49
TEE	259.9	297.17	241.16	313.05	391.23	523.1	478.55	517.78	562.99	549.82	694.27	674.49

TEC	1247 5.08	1515 5.62	1205 7.75	15339 .21	24256	3190 9.22	30627 .03	33655 .54	43350 .4	4343 5.74	5276 4.22	5261 0.4

6. Result Analysis and Discussion

The results represented in Table 4 and 5 can be further analysed to determine the most cost effective and less effort oriented approach that can be implemented in software industries. In this context, we analysed the findings of Table 4 and 5 respectively on the deterministic parameters.

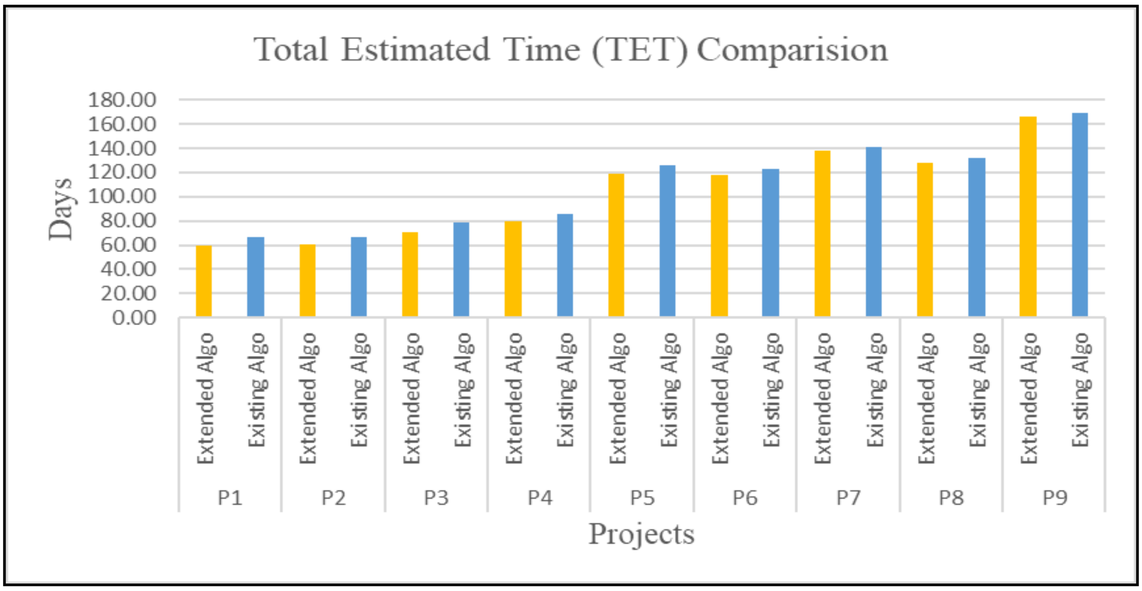

Figure 2: Comparison of Total Estimated Time (TET) among all 12 projects

Figure 2 depicts the comparison of Total Estimated Time (TET) in terms of days among all 12 projects for both the data sets. It is found that the extended algorithm requires a smaller number of days as compared to the existing algorithm. This difference is obtained in each project. The difference in total estimated time ranges from 1.92 Days to 8.55 Days.

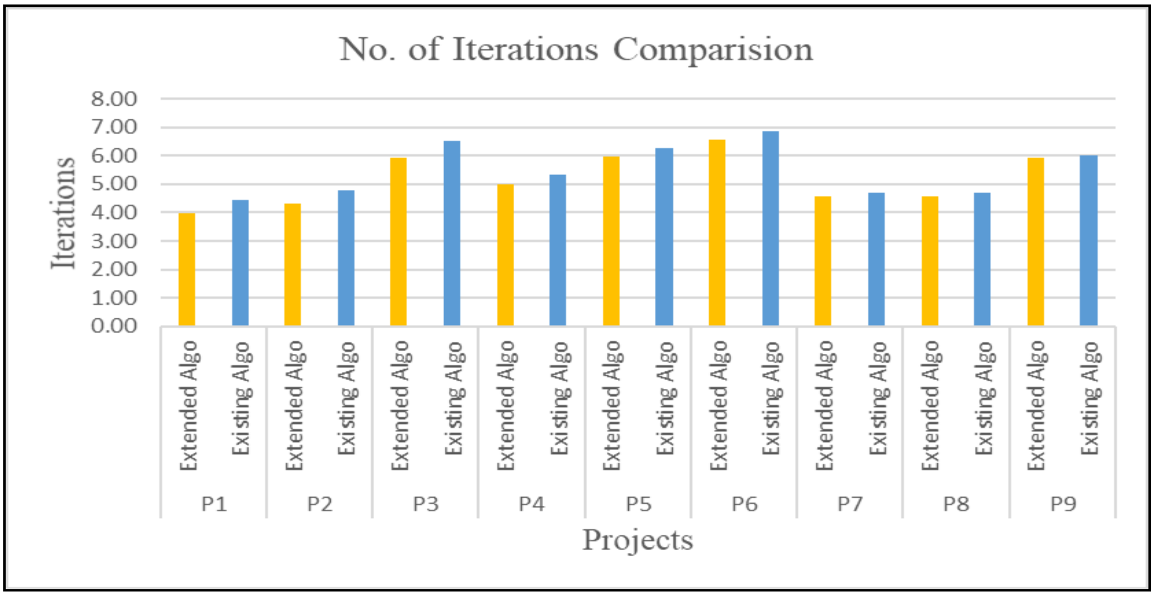

Figure 3: Comparison of Number of Iterations (I) among all 12 projects

Similarly, Figure 3 shows the comparison of the required number of iterations (I) among all 12 projects for both data sets. It is evident that the extended algorithm requires fewer iterations as compared to the existing algorithm. This difference is apparent in every project. The difference in the number of iteration (Sprint) ranges from 0.07 to 0.58.

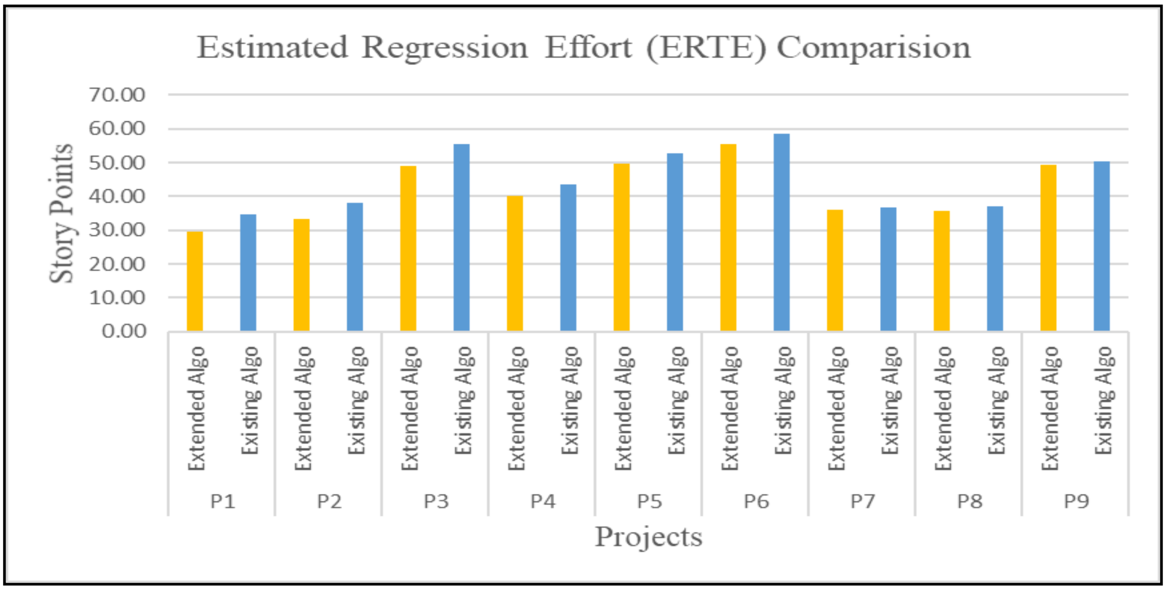

Figure 4: Comparison of Estimated Regression Testing Effort (ERTE) among all 12 projects

Figure 4 depicts the comparison of Estimated Regression Testing Effort (ERTE) in terms of story points among all 12 projects for both the data sets. It can be seen that the extended algorithm requires fewer story points compared with the existing algorithm. This difference is obtained in each project. The difference in estimated regression testing effort ranges from 0.69 to 5.79.

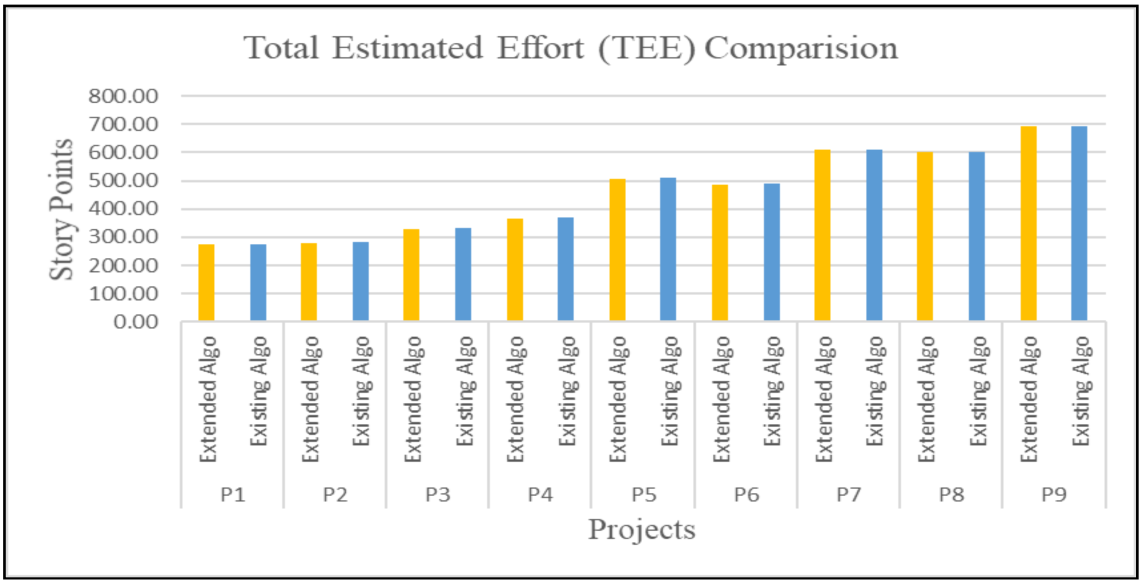

Figure 5: Comparison of Total Estimated Effort (TEE) among all 12 projects

Figure 5 depicts the comparison of Total Estimated Effort (TEE) in terms of story points among all 12 projects for both the data sets. It is obtained that the extended algorithm requires less amount of effort compared with the existing algorithm. This difference is apparent in every project. The difference in the unit of story point ranges from 0.29 SP to 4.70 SP.

Figure 6 shows the comparison of total estimated cost (TEC) in terms of cost in dollar ($) among all 12 projects for both the data sets. It is clear that the extended results in lower cost compared with the existing algorithm. This difference is in every project. The difference in the cost ranges from $22.12 to $286.87.

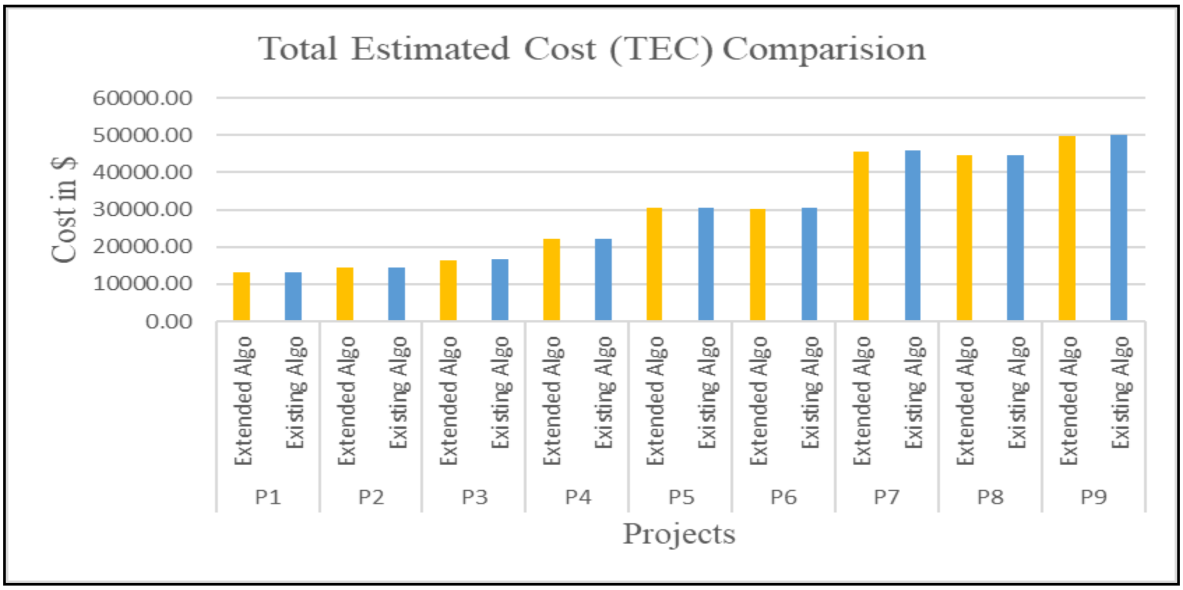

Figure 6: Comparison of Total Estimated Cost (TEC) among all 12 projects

Table 6 provides the detailed data related to the difference between the existing algorithm and the extension in an existing algorithm among 12 projects. It is found that the extended algorithm is advantageous over the existing algorithm in all 3 levels groups as Low, Medium, and High.

Table 6: Difference between existing algorithm and extended algorithm among 12 projects

Differences	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
TET	7.52	6.49	8.55	7.46	5.24	6.3	5.27	5.26	1.92	2.93	2.91	2.91
I	0.58	0.46	0.57	0.53	0.29	0.35	0.26	0.31	0.07	0.1	0.1	0.11
ERTE	5.79	4.64	5.7	5.33	2.91	3.5	2.63	3.1	0.69	1.04	1.04	1.08
TEE	3.59	2.44	3.5	3.13	4.11	4.7	3.83	4.3	0.29	0.64	0.64	0.68
TEC	172.16	124.25	174.86	153.43	254.94	286.87	245.37	279.18	22.12	50.93	48.55	52.78

7. Conclusion and Future Scope

In this article, we proposed an extension to an existing algorithm for an Agile-based software project's estimation based on regression testing. Through this approach, we calculated the estimated time, number of iterations, estimated regression testing effort, total estimated effort, and total estimated cost for all considered projects of our data set. In the result analysis section, we compared our obtained results with findings of the existing approach. It was found that the results obtained from the extended algorithm are cost-effective and less effort-oriented to a great extent.

The difference in total estimated time (TET) ranged from 6.49 to 8.55 days in low factors level group. In medium level group, it is ranging from 5.24 to 6.30 days and from 1.92 to 2.93 days in the high factors level group. In the case of the number of iterations (I), the difference ranged from 0.46 to 0.58 iterations in the low factors level group. The medium level group was from 0.26 to 0.35 iterations and from 0.07 to 0.11 iterations in the high factors level group. The difference in estimated regression testing effort (ERTE) ranged from 4.64 to 5.79 story points in the low factors level group. In the medium level group, it was from 2.63 to 3.50 story points and from 0.69 to 1.08 story points in the high factors level group. The difference in total estimated effort (TEE) ranged from 2.44 to 3.59 story points in the low factors level group. While in the medium level group, it was 3.83 to 4.70 story points and from 0.29 to 0.68 story points in the high factors level group. Similarly, in the case of total estimated cost (TEC), the difference ranged from $124.25 to $174.86 in the low factors level group. In the medium level group, it was from $245.37 to $286.87 and from $22.12 to $52.78 in the high factors level group. The results obtained from the extended algorithm incorporated more success factors compared to the existing algorithm. So, the presented extended algorithm in this article can be accepted and applied in software industries to achieve more success chances of software systems with less effort and reduced cost.

Recent work has tended to focus on an advanced mechanism to estimate effort and cost. There is an increasing interest in the automated and intelligent effort estimation approaches. In this context, there is a scope for developing new machine learning models for more accurate effort estimations and predictions of the level of the factors for Agile-based projects. We hope the future will see a further coming of age with the development of highly precise effort, time, and cost estimation on the basis of training of data sets.

References

[1] Al-Zewairi1, M., Biltawi, M., Etaiwi, W. and Shaout, A. “Agile Software Development Methodologies: Survey of Surveys.” *Journal of Computer and Communications*, vol. 5, pp. 74-97, 2017.

[2] R. Agrawal, D. Singh and A. Sharma, "Prioritizing and optimizing risk factors in agile software development", 2016 *Ninth International Conference on Contemporary Computing (IC3), Noida*, pp. 1-7, 2016.

[3] Srinivasan Selvaraj et al., "A Survey on Effort Estimation Techniques in Agile Software Development", *International Journal of Interdisciplinary Global Studies*, vol. 14, Issue 04, pp. 91-95, Dec 2020.

[4] https://www.agilealliance.org/agile101/the-agile-manifesto/last accessed on 26 January 2022.

[5] Tsun Chow, Dac-Buu Cao, “A survey study of critical success factors in agile software projects”, *Journal of Systems and Software*, vol. 81, issue 6, pp. 961-971, 2008. ISSN 0164-1212.

[6] Pantjawati Sudarmaningtyas and Rozlina Mohamed, “A Review Article on Software Effort Estimation in Agile Methodology”, *Pertanika Journal of Science & Technology*, vol. 29, issue 2, pp. 837-861, April 2021.

[7] W. Rosa, B. K. Clark, R. Madachy and B. Boehm, "Empirical Effort and Schedule Estimation Models for Agile Processes in the US DoD," in *IEEE Transactions on Software Engineering*, doi: 10.1109/TSE.2021.3080666.

[8] Ravi Kiran Mallidi and Manmohan Sharma, “Study on Agile Story Point Estimation Techniques and Challenges”. *International Journal of Computer Applications*, vol. 174, no. 13, pp.9-14, January 2021.

[9] A. Sharma and N. Chaudhary, "Linear Regression Model for Agile Software Development Effort Estimation," 5th IEEE *International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India*, pp. 1-4, 2020.

[10] M. Fernández-Diego, E. R. Méndez, F. González-Ladrón-De-Guevara, S. Abrahão and E. Insfran, "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review," in *IEEE Access*, vol. 8, pp. 166768-166800, 2020.

[11] C. Ratke, H. H. Hoffmann, T. Gaspar and P. E. Floriani, "Effort Estimation using Bayesian Networks for Agile Development", 2nd *International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia*, pp. 1-4, 2019.

[12] Popli R., Chauhan N. (2019) A Sprint Point Based Tool for Agile Estimation. In: Hoda M., Chauhan N., Quadri S., Srivastava P. (eds) Software Engineering. Advances in Intelligent Systems and Computing, vol 731. Springer, Singapore. https://doi.org/10.1007/978-981-10-8848-3_6.

[13] A. Saini, L. Ahuja and S. K. Khatri, "Effort Estimation of Agile Development using Fuzzy Logic," 7th *International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India*, pp. 779-783, 2018.

[14] Arora, Mohit & Chopra, Shivali & Gupta, Pooja. "Estimation of Regression Test Effort in Agile Projects", *Far East Journal of Electronics and Communications*. pp. 741-753, 2016.

[15] M. Usman and R. Britto, "Effort Estimation in Co-located and Globally Distributed Agile Software Development: A Comparative Study", *International Conference on Software Process and Product Measurement*, pp. 219-224. 2016.

[16] Rashmi Popli & Naresh Chauhan, "An Agile Software Estimation Technique based on Regression Testing Efforts", 13th *Annual International Software Testing Conference*, IEEE, Bangalore, India, pp. 1-9, Dec 2013.

[17] Boehm B, W., "*Software Engineering Economics", Prentice-Hall*, 1981.

[18] R. Popli and N. Chauhan, "Agile estimation using people and project related factors," 2014 *International Conference on Computing for Sustainable Global Developmen*t (INDIACom), 2014, pp. 564-569, doi: 10.1109/IndiaCom.2014.6828023.

[19] R. Popli and N. Chauhan, "A sprint-point based estimation technique in Scrum," 2013 *International Conference on Information Systems and Computer Networks*, 2013, pp. 98-103, doi: 10.1109/ICISCON.2013.6524182.