Performing Reengineering using Scrum Agile Framework

Jaswinder Singh[1,*]
[1]*Chitkara University Institute of Engineering and Technology*
Chitkara University
Punjab, India
[*]*Department of Computer Application, IKGPTU*
Kapurthala, Punjab, India
jaswinder.singh@chitkara.edu.in

Kanwalvir Singh Dhindsa[2]
[2]*Baba Banda Singh Bahadur Engineering College*
Fatehgarh Sahib, Punjab, India
kdhindsa@gmail.com

Jaiteg Singh[3]
[3]*Chitkara University Institute of Engineering and Technology*
Chitkara University
Punjab, India
jaiteg.singh@chitkara.edu.in

***Abstract* - Software reengineering is an approach of improving the maintainability of the existing software. Reengineering enhances the quality of the software and keeps the software alive for a more extended period. For the last four decades, researchers used a reengineering approach to transform the legacy system, to adopt new requirements as well as to reduce the faults of the system. The proposed work reduces the complexity of the software by applying the reengineering process. Scrum agile approach has been used for implementing the process of reengineering. Work also validates the improvement in the software after performing reengineering. The internal design complexity of the software is measured and reengineering is performed to reduce software complexity.**

Keywords—reengineering, agile framework, Scrum, planning poker

I. INTRODUCTION

Software reengineering is a critical approach to reduce the complexity of the software. For the last four decades, researches contribution to the work of software reengineering is significant. Reengineering Includes three main tasks

- Reverse Engineering
- Alterations
- Forward Engineering

As shown in Fig.1, Reengineering goes through interpretation and analysis of the existing code, making required alterations and performing integration of updated modules as well as their testing. Input to the reengineering process is the existing software and output is the reengineered software with enhanced quality.

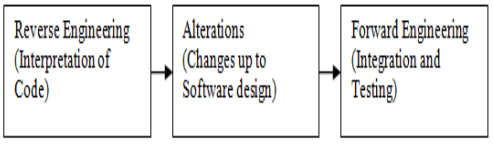

Fig.1.Reengineering process.

The reengineering process itself can be improved by utilizing the advantages of agility. The agile framework may enhance the reengineering process by using the feature of collaborative problem-solving. Proposed work used the Scrum approach of the agile framework for performing the reengineering process. The construction of the product in Scrum is iterative, and iteration is called Sprint. One sprint is 2-4 weeks long. A complete software construction can have multiple sprints. After each sprint, the integration of the system is performed. Fig. 2 below shows three main activities of the Scrum process. Requirements are received as user stories. The requirement is implemented in Sprint. For more requirements or changes, sprint is performed again, and finally, after each sprint integration is performed.

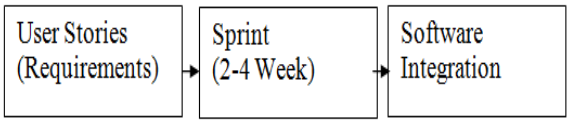

Fig.2. Scrum process.

II. LITERATURE

Researchers justified the role of reengineering in software enhancement and maintenance improvement[1]–[5]. Reengineering improves the maintainability of the software, and it is the best approach to convert a legacy system to a new system[6].

An agile framework is one of the most popular and highly used Software development approaches among software professionals [7]. The name Agile comes from the word 'agility' which means developing the software quickly and with ease. Two decades back, a group of seventeen people agreed upon Agile manifesto, which introduced a new approach to software development[8]. The manifesto stated twelve principles that aimed to uncover better ways of software development. Agile Framework gives many advantages to its stakeholders. The most prominent is the involvement of customers during the software development process. Agile promotes customer collaboration and thus enhances customer satisfaction[9]. The agile methodology includes various software development approaches like Extreme Programming, Feature Driven Development Scrum,[10][11]. Among various approaches of the agile, the most successful approach is Scrum[12].Scrum emphasizes all stakeholders' involvement while developing the software. The focus is to deliver the working software as soon as possible. The development iteration process in the scrum is called Sprint. An important estimation and planning technique in an agile framework is planning poker. According to Usman*et al.*[13], the most useful technique of

effort estimation in agile is planning poker. Planning poker is also used for empirical story point estimation in the work of Haugen [14], Molokken-ostvold*et al.*[15], and Mahnic*et al.*[16]. As stated by Cohn [17], "Planning poker is a proper mix of expert opinion, analogy, and disaggregation techniques which can give quick and reliable estimates successfully." Forreengineering, it is essential to estimate the requirements to reduce the internal design complexity (CK metric values). The estimation is performed using Planning poker. Requirements are estimated in terms of user stories. Story points are assigned to each user story and this assignment is consensus-based. For assigning story points, numeric numbers are used.

The main objective of reengineering using Scrum is to reduce the complexity of the software and ultimately to improve the maintainability of the software. To access the improvement in the maintainability of the software, internal design Complexity of software in terms of Cohesion, coupling, Inheritance, the number of methods has been analyzed.

III. Case Study

The case study includes Java-based Object-Oriented software, which is reengineered using an agile scrum software development approach. The software classes chosen for reengineering tasks are given in Table I. Software design complexity is determined using Chidamber and Kemerer (CK) object-oriented metric [18]. CK java Metric tool (CKJM) version 1.9 [19] is used for calculation of CK metrics suit.

CK metric suit is measured for each class using the CKJM tool. Various numeric values have been assigned against each metric for every class. As stated by Shatnawi[20], the larger the value of the CK metric, the more challenging to maintain and test the software system. So the proposed works CK metric values using the agile methodology.

TABLEI. CK Metric Analysis Of Three Classes Before Reengineering

Classes	WMC Metric Value	DIT Metric value	NOC Metric Value	CBO Metric Value	RFC Metric Value	LCOM Metric Value
Login class	12	6	0	9	78	60
IDE class	17	6	0	17	121	70
User Detail Class	23	5	0	12	109	183

As given in Table I, Classes have been assigned with various values. These six metrics are important to analyze the design complexity of the software. More the values of these metrics, the greater will be the complexity of the software. In Table I,the Value of LCOM is 183 that shows that UserDetails Class highly lacks the cohesiveness in their module.

IV. Estimation Measure Before Reengineering

In agile, requirements or features required in the software are documented as user stories. Each story is estimated by giving numeric values called Story points. In planning poker, estimators assign story points that are usually picked up from the Fibonacci series that is 0,1,2,3,5,8,13, and so on. Assigning 0 means the least efforts required, and as the assigned value increases, efforts required to solve the user story also increases. Estimated story points for software classes are shown in Table II. For three classes, story points are estimated to reduce the CK metric value that is to reduce the internal design complexity of the software.

TABLE II. Story Point Estimations

Sr No	User Stories (Requirements)	Estimated Story Point Values
1	For "UserDetail Class," the requirement is to reduce complexity and making it more maintainable.	8
2	For "IDE Class," the requirement is to reduce complexity and to make it more maintainable.	5
3	For "Login Class," the requirement is to reduce complexity and making it more maintainable.	2

Estimated story points assigned in Table II shows that the UserDetail class is more complicated as compared to IDE as well as Login class.

A. Efforts Estimation Before Reengineering

Researchers [21] have estimated the efforts of software reengineering but using conventional methods. For efforts calculation, an updated agile effort estimation model [22] is proposed.The basic estimation Model method is shown in equation (1)

$$Effort = 6.8 * REQ0.4071 \quad * Staff0.440 \qquad (1)$$

Here are given in Person-Month (PM), and REQ represents Requirement size.

For the java classes, size is measured using story points. So REQ in research work is proposed to represent the size of story points assigned to classes. The total story points assigned to classes are fifteen. So assigning REQ= 15 in equation (1). The value of variable Staff represents the number of people involved throughout the process. Considering the value as 3(Scrum Master, Developer, and tester), the effort measured is around 33 PM.

By applying the Scrum approach [21,23], various reengineering tasks performed for three classes are shown in Table III below. Reengineering is performed in 3-Week Sprint.

TABLE III. Reengineering Tasks Using Scrum

Sr No	Sprint	Tasks	Sub Tasks
1	Sprint 1.0-First Week	Reverse Engineering	Existing Structure Understanding
			Design Specifications Analysis
			Requirements Analysis
			Retrospective
2	Sprint 1.0-Second Week	Alterations	Class Design Analysis
			Class Behavior Reconstruction
			Testing (Unit and regression)
			Retrospective
3	Sprint 1.0-Third Week	Forward reengineering	Integration
			Integration testing
			Retrospective

After performing the reengineering tasks, the CKJM tool is applied again for the three classes. Table IV shows a reduction in CK metric values.

TABLE IV. CK METRIC ANALYSIS OF THREE CLASSES AFTER REENGINEERING

Sr. NO	Metric/ Classes	Reengineered Login	Reengineered IDE	Reengineered User Detail
1	WMC	4	4	6
2	DIT	6	6	6
3	NOC	0	0	0
4	CBO	5	12	4
5	RFC	48	102	67
6	LOCM	2	0	0
	Total	65	124	83

V. ESTIMATIONS MEASURE AFTER REENGINEERING

Story points are measured again for the three classes, and the estimated values are given in Table V. The reason for lower estimations is clearly because of the reduction in the complexity of Software classes.

TABLE V.STORY POINT ASSIGNMENT AFTER REENGINEERING

Sr No	Requirements	Estimated Story Points
1	For "UserDetail Class," the requirement is to reduce complexity and making it more maintainable.	2
2	For "IDE Class," the requirement is to reduce complexity and to make it more maintainable.	2
3	For "Login Class," the requirement is to reduce complexity and making it more maintainable.	1

A. Efforts After Reengineering

For cost calculation, the total story points assigned to classes are 5. So assigning REQ= 5 in equation (1), the effort measured is around 21 Person-Month. The main cause of effort reduction is due to a decrease in internal design complexity metrics. As shown in Table 4, CK metric values have been reduced after reengineering.It is easy to estimate and assign the story points to those classes which already undergo modifications by the scrum team.Thus after performing reengineering, the maintenance efforts required to maintain the software system have been reduced.

VI. CONCLUSION

It is vital to improving the maintainability of software. Story points are assigned to the requirements before and after reengineering. This assignment is based on a consensus-based approach. Results validate the reduction in complexity and maintenance efforts. Size estimations are also reduced after performing reengineering as there is a reduction in the software design complexity. Agile maintenance effort for the software before reengineering is 33 Person-Month, and after reengineering, the agile maintenance effort estimated is 21 Person-Month. Thus proposed paper shows improvement in software maintenance after performing reengineering and improves the quality of software. Further, the proposed work can be validated by considering various other software quality factors. To generalize the results, it is required to consider more software classes.

REFERENCES

[1] H. M. Sneed, "Planning the reengineering of legacy systems,"IEEE Softw., vol. 12, no. 1, pp. 24–34, 1995.
[2] H. M. Sneed and A. Kaposi, "A study on the effect of reengineering upon software maintainability," in Conference on Software Maintenance , pp. 91–99, 1990.
[3] I. Jacobson and F. Lindström, "Reengineering of old systems to an object-oriented architecture,"ACM SIGPLAN Notices., vol. 26, no. 11, pp. 340–350, 2005.
[4] J. J. Mulcahy and S. Huang, "Reengineering autonomic components in legacy software systems: A case study,"in 11th Annual IEEE International Systems Conference, SysCon ., Canada, pp.1-7, 2017.
[5] A. Cimitile, "Towards reuse reengineering of old software," inFourth International Conference on Software Engineering and Knowledge Engineering, Italy,pp140-149, 1992.
[6] M. Majthoub, M. H. Qutqut, and Y. Odeh, "Software re-engineering : An overview,"in 8th International Conference on Computer Science and Information Technology (CSIT), Amman, pp. 266–270, 2018.
[7] M. Hirsch, "Moving from a plan-driven culture to agile development," in 27th international conference on Software Engineering, NY, United States, pp. 38, 2005.
[8] M. Fowler and J. Highsmith, "The agile manifesto (article),"Softw. Development Magazine: The Lifecycle Starts Here, pp. 1–7, 2001.
[9] T. Dingsøyr *et al.*"Agile software development," Springer, vol. 5, no. 1, pp. 31-59 2010.
[10] A. Scott, W. and H. Matthew, Agile For Dummies, IBM Limited Edition, 2nd ed. John Wiley & Sons, 2002.
[11] L. Williams, Agile Software Development Methodologies and Practices, 1st ed., vol. 80, Elsevier Inc., 2010.
[12] The Standish Group, "CHAOS REPORT: 21stAnniversary Edition Chaos Resolution for All Projects," 2014.
[13] M. Usman, E. Mendes, and J. Börstler, "Effort estimation in agile software development,"in 19th International Conference on Evaluation and Assessment in Software Engineering, China,pp. 1–10, 2015.
[14] N. C. Haugen, "An empirical study of using planning poker for user story estimation," in Agile Conference, pp. 23–31, 2006.
[15] 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.
[16] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories,"J. Syst. Softw., vol. 85, no. 9, pp. 2086–2095, 2012.
[17] M. Cohn, Agile Estimating and Planning, Pearson Education, 2006.
[18] S. R. Chidamber and C. F. Kemerer, "A Metrics suite for object-oriented design,"IEEE Transactions on Software Engineering, vol. 20, no. 6, pp. 476–493, 1994.
[19] M Jureczko and D Spinellis. "Using object-oriented design metrics to predict software defects, Fifth International Conference on Dependability of Computer Systems DepCoS, Monographs of System Dependability, Poland, pp 69–81, 2010.
[20] R. Shatnawi, "A quantitative investigation of the acceptable risk levels of object-oriented metrics in open-source systems,"IEEE Transactionson Software Engineering, vol. 36, no. 2, pp. 216–225, 2010.
[21] P. Kumawat and N.Sharma, "Design and development of cost measurement mechanism for re-engineering project using function point analysis," International Conference on Advanced Computing Networking and Informatics. Advances in Intelligent Systems and Computing, vol. 870. Springer, Singapore,2019.
[22] W. Rosa, R. Madachy, B. Clark and B. Boehm, "Early phase cost models for agile software processes in the US DoD," 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Toronto, pp. 30- 37, 2017.
[23] J. Singh, K. Singh, and J. Singh, "Reengineering cost estimation using scrum agile methodology," IJCISIM, vol. 11, pp. 208–218,2019.