Cost Efficient Scrum Process Methodology to Improve Agile Software Development

Ehab E. Hassanein1
dept. Information systems
Faculty of Computers and Artificial intelligence
Cairo, Egypt
ehabez1@gmail.com

Salma A. Hassanien2
dept. Information systems
Faculty of Computers and Artificial intelligence
Cairo, Egypt
Salma.hassanien@ejust.edu.eg

Abstract— *Current Agile methodology doesn't put into consideration the effect of the variability of skills required to perform the tasks of the project. Furthermore, it does not account. In advance, for the availability of those skills among the agile team of the developers at the proper time they are needed. Narrow view of current sprint requirements and their tasks hide the future need of those skills, hence wasting valuable resources that might have been used in earlier stages of the development. Consequently, a waste of resources usually exist that needed to be accounted for in every Agile project using the current methodologies. The proposed CESP process will have global project view that will help to eliminate waste of resources in those earlier stages by rearranging the tasks that will required in earlier stages. Hence, reduces the overall projects time and cost. All that while maintaining the agility and flexibility desired by adopting the Agile methodology. The main objective of our paper is reduces the overall projects time and cost. All that while maintaining the agility and flexibility desired by adopting the agile methodology.*

KEYWORDS- Agile;Scrum; Kanban;Scrumban; Scrumbanfall ; L-Scrumban ,Cost,Efficiency

I. INTRODUCTION

In the past decade, there has been a shift away from traditional software methodologies. As many people found that the overheads imposed by traditional methods such as the waterfall model, the standard process, etc., slowed down the development process and did not achieve the required quality [4]. Before 2001[1]-[3], the software industry used traditional software development processes (i.e., classical waterfall model, iterative waterfall model, spiral model, RAD model). The most common method used was the Waterfall. It is a sequential in nature and it dominated the world for long time. This model was first cited by Winston W. Royce in 1970[7]. While these traditional models are known to be cost saving for bigger, off-shore projects, there is criticism that exists [3]-[7]. Adopting Agile methodologies or any of its framework emphasize the project correct requirement execution which leads to less efficient development. In this research we devised a frame work that combine the quality, flexibility and precision of the Agile process with the efficiency of developing projects with less waste of resources.

The paper is designed in the following sections, section 2 explains the literature of agile model including scrum, section 3 explains the limitations of agile model while using in large software development in large organizations and challenges that faced Scrum, section 4 explains the research methodology of paper, section 5 explains the result discussion and suggestions and section 6 explains conclusion.

II. EASE OF USE

A. Agile Methodology

The term "Agile software development" dates back to 2001 when a team consist of seventeen person, they gathered in a ski resort in Snowbird, Utah each with his own way of practicing software development to find out in common among their method of software development. After this meeting, they came out with the Manifesto for Agile Software development, a collection of asset of four values and twelve principles. Since the introduction of the "AgileManifesto" in 2001, agile methodologies have gained much popularity and success. The software industry had a huge shift from practicing traditional Software development to now widely adopting agile methodologies.

The use of the agile framework was first suggested for nearly twenty years. During that time, there has been significant adoption of agile principles and methods in organizations and teams. Agile methods are reported to bring business value to users but only little research existed about whether this is actually happening at the expense of the well-being of the personnel in agile projects.

In general, the shift in Agile methodologies focuses more on individuals and interactions over processes and

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

tools, working process over detailed documentation, customer collaboration over contact negotiation and responding to change rather than following a plan [5], [6]. It was also seen that Agile software development could handle changing requirements flexibly [4]. The Software is developed in iterations following agile models [8]. The completion time of an iteration is from two weeks to one month.

Agile methodologies invite the developers to get involved in testing, rather than a separate quality assurance team. Agile methodologies are popular because of their ability to work effectively and efficiently in changing environments.This is due to the modern practices and principles enabling development teams to complete software as per the schedule.

The main goal of agile methodologies is to increase the ability to react and respond to changing customer, business and technological needs at all organizational levels [17]. Several companies are moving to agile software development to improve quality and productivity, and to reduce delivery times. In contrast to traditional software development processes, where work is typically broken down into a series of sequential steps, agile methods rely on short, iterative cycles and close collaboration between the customers and the development team [18].

This active participation of the customer or user throughout the development lifecycle can lead to major weaknesses [19]. Sometimes customer does not have the time or the needed knowledge to interact with the development team. Other observed weaknesses of Agile methodologies as presented by Tarwani et al. [20] are miscommunication, resource increase, overall cost increase, in appropriateness for large projects and lack of coordination.

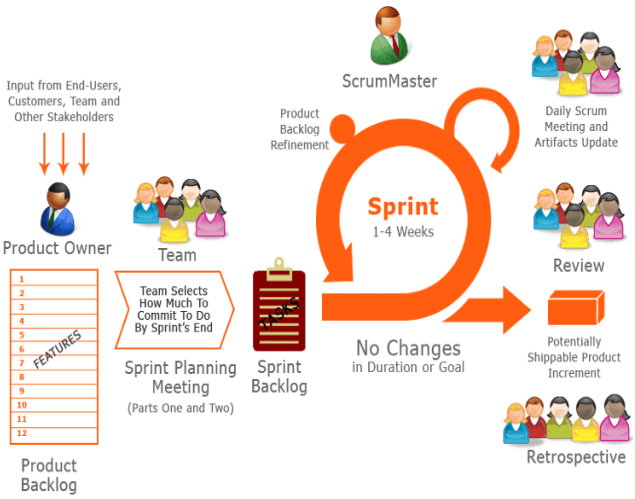
B. The XP methodology

XP methodology is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. Many case studies are available comprising successful stories of XP model for small projects. Although XP has enormous strengths but still significant number of software, companies are hesitant to transfer from plan driven methodologies to XP [9]. In recent years, the lean approach to software development and its concepts have become increasingly popular. **Define the lean approach from a source***The lean approach was first applied in the manufacturing industry. It was devised at Toyota and was originally called the Toyota Production System (TPS). The aim of the lean approach is to deliver value to customers more effectively and efficiently through the process of finding and eliminating waste, which is a huge impediment to the productivity and quality offered by an organization (Liker and Hoseus 2008; Magee 2008). Numerous methods and frameworks have been developed to provide a framework for the agile approach to software development, the two most common being scrum and extreme programming.

C. Scrum

Scrum is the most used agile framework in the industry [10] it was originally applied in 1990's by Ken Schwaber and Mike Beedle According to Schwaber and Sutherland (2017) Scrum defined as "A framework within which people can address complex adaptive problems while productively and creativity delivering products of the highest possible value. " Scrum is based on an empirical process control theory [10], describes an event, namely, Sprint Retrospective Meeting, and a role, namely, Scrum Master, to support the project teams' continuous improvement.

The scrum team consists of the product owner (PO), the scrum master (SM) and the developers. The scrum events, sprint planning, the daily scrum, the sprint review (of the product), and the sprint retrospective (of the process) are used to promote inspection and adaptation. The scrum artifacts are the product backlog (overall product requirements), the sprint backlog (requirements for each sprint) and the increment (the specific deliverable for each sprint). A small stream of research has examined modifications that are commonly made to scrum to better respond to requirements [23], to



https://sites.google.com/site/ijcsis/
ISSN 1947-5500

adapt to distributed teams or to adapt in larger projects [23].

It is common to combine it with other agile frameworks such as extreme Programming [22] and Kanban[23] (a process called ScrumBan).

The Major Challenges of Scrum

Adopting the Scrum process lead to a number of major challenges that hindered the efficient use of the framework. Although Scrum is simple to understand, it is hard to master Nevertheless, it is relatively difficult to follow Scrum in large projects as it is more suitable for small projects.[15][16] It requires self-organizing, cross-functional teams and collaboration between business and development teams to develop products incrementally. Especially in companies with former waterfall approach, these factors might lead agile to fail.

The following challenges are typical to Scrum based projects:

a) Sprint Workload

A challenge was observed when we apply Scrum, represented in the sprint workload as that Teams have a lot of work to do in every sprint. It can be said that the challenge does not happen every Sprint, but this represents a challenge to the department often. According to (Popli & Chauhan, 2013) the main cause of this problem is an inaccurate estimate of the length of the task by the developer team.

b) Testing in the Next Sprint

There was another challenge that mentioned by Test Team Manager is that the testing is not always done in the same sprint as development. According to Scrum theory that presented by (Schwaber &Sutherland.2017): every sprint should produce a "Done" which implies that testing has to be done within the same sprint as development. West, Gilpin ,Grant And Anderson(2011) mention this problem by states that testing is often posted to another separate team which contradicts with agile principles. One reason for why testing and development is not done in the same Sprint is that there might not be enough time in the Sprint to do both.

D. Kanban

One of the most popular principles of Lean approach is Kanban [2]. Kanban is a visual method that helps in managing the production of a product [3]. This methodology can not only be used for development but also has its strengths in teaching, According to Ghobadi, S.,&Mathiassen, L. In contrast to Scrum, it is less descriptive and focuses on visualizing workflows using Kanban boards, limiting the work in progress and ensuring that work flow as fast as possible through the system by removing bottlenecks [35]. Kanban is considered, relatively speaking, less prescriptive and more adaptive than Scrum, meaning there are fewer rules to follow [35]. Kanban has become popular because of its ease of implementation, use of visual controls, work in progress management, and relentless focus on the continuous process improvement.

One of the most popular principles of the lean approach is kanban, which is a tool for controlling the logistical chain from a production point of view and is a –method by which just in- time (JIT) is achieved (Ohno 1988). Kanban is one of the two pillars in the lean house that was developed by Toyota. Since 2003, David Anderson (2010) has attempted to tailor the Kanban system to software development, formulating the Kanban method, which applies incremental, evolutionary process and systems changes in organizations [3].

Scrum, which is based on an empirical process control theory [10], describes an event, namely, Sprint Retrospective Meeting, and a role, namely, Scrum Master, to support the project teams' continuous improvement.

E. . Scrumban

Scrumban is a hybrid Agile method of Scrum and Kanban[30]-[31]. Scrum and Kanban hybridization is needed Promote the scrum method by improper deletion Practices and adopt appropriate practices from Kanban method [30]. The appropriate practices of kanban and scrum have been adopt by the team members based on different situations to satisfy the needs [32],[33]. The major advantage of scrumban is make Agile team members to be creative in developing new methods to meet their requirements [31].on the other hand there are no specific practices for Scrumban, but the Agile team members have to understand, which practices of Scrum and Kanban deliver value and choose the appropriate practices accordingly [32]. In other words, Scrumban ensures a slow transition from Scrum to Kanban. ScrumBan framework similar to Kanban in pulling the work according to the needs, continuous flow, measuring

FIGURE 1. SCRUM + KNABAN = SCRUMBAN

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

lead time, has Kanban board, has ready and open queues instead of product backlog as in scrum, limits the work in progress, prioritized the work based on demands and the class, and has diagram demonstrates the continuous flow, and like scrum in: having ceremonies suchas daily meeting and retrospective meeting, the roles of the team which having product owner, scrum master, the workersand the team should have members with different specialties and expertise (cross-functional team). According to the latest studies regarding the impact of agile methodologies on software development market [34] [35], Scrumban, as hybrid methodology, has gained one percent (from 6% in 2015 to 7% in 2016), but remains behind other two hybrid methodologies that keep their position in surveys: Scrum-XP Hybrid (10%) and Custom Hybrid (multiple methodologies –8%).

F. L-Scrumban

Lean principles only provide the "behavioral approach" to guarantee the success of the development process. Lean does not cover the technical and managerial issues, and all its concern is about minimizing the wastes and improving the quality. Lean, therefore, is not considered as a complete model to be implemented in software development field [35].

L-ScrumBan methodology is an agile framework for managing software development process. The term L-ScrumBan is derived from Lean thinking, Scrum methodology, Kanban tool. Moreover, L refers to Lean thinking, Scrum refers to Scrum methodology, and Ban refers to Kanban tool.

L-ScrumBan methodology has five tools: Customer Demand List, Product Backlog, Sprint backlog, General board, and L- ScrumBan board. Three roles: Product Owner, Team Master, and The Developers. Six meetings: Eliminate-waste meeting, Sprint Planning, Daily meeting, Quality-Test meeting, Sprint Review, and Sprint Retrospective.

G. Scrumbanfall

Because each method has its own advantages and disadvantages, Scrum, Kanban, or Independent Waterfall cannot provide complete solutions to all of the challenges of Software Engineering Management (SEM) process. Scrumbanfall is an agile integration of Scrum and Kanban with Waterfall model using the mixture of traditional SDLC protocols with the empiricism, agility and workflow management.

Scrumbanfall = Scrum + Kanban + Waterfall

Figure 3 represent the combination of scrum and kanban with waterfall for the formation of scrumbanfall. Scrum base of Scrumbanfall, by keeping Kanban in the center of the Scrum and wrapping Waterfall prior to Scrum Sprints. Following are core elements selected from each of Scrum, Kanban and Waterfall models in the formation of Scrumbanfall.

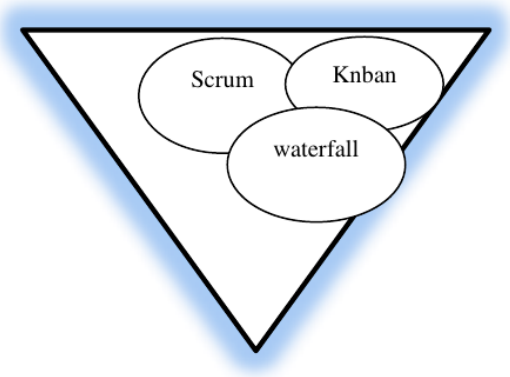


FIGURE 2 SCRUMBANFALL

The integration of Scrum and Kanban with Waterfall provides a great strength to Software Engineering Management (SEM) practices in the form Scrumbanfall. The phases, which are derived from the Scrumban [42] (Scrum + Kanban) and Waterfall as Scrumbanfall is a combination of all of them.

- Requirement Analysis
- Project Planning
- Sprint
  - Sprint Planning
  - Daily Scrum
- Work Item Management
  - To – Do
  - In Progress
  - Done
- Continuous Integration and Continuous Delivery
- Sprint Review
- Sprint Retrospective

Finally, the combination of Scrum and Kanban with Waterfall gives a great strength to Scrumbanfall but still it can't cover all.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Summery

As we presented none of the frameworks address the issue of waste of recourse due to the different requirements assigned to each sprints, instead they all emphasis performing the scrums process or a variation of it in clear and direct manner as well as solve the integration problems resulting from combining different development frameworks.

III. *The CESP methodology*

Lean principles only provide the "behavioral approach" to guarantee the success of the development process. Lean does not cover the technical and managerial issues, and all its concern is about minimizing the wastes and improving the quality. Lean, therefore, is not considered as a complete model to be implemented in software development field [33]. Scrum illustrates the details of development process using the scrum methods including all the phases and practices that can be done with emphasis on one sprint at a time. That guaranteed the agility and the independence of each sprint such that the customer my change add or drop requirements and the model respond accordingly. But that lack of a holistic approach leads to a lot of waste of recourses and lack of good project planning.

The CESP will maintain the agility of the scrum process in addition to having a global plan for the project that can change many times through the software development cycle. Each change may affect the cost and time of the development. If the changes reduce the time or the cost the CESP will recompute the project plan such that are plan for the whole project sprints is presented. For those changes that increase the cost and/or time. A new project plan is presented immediately. This plan may need renegotiation with the customer specially if it will eventually result in changes of delivery dates that were agreed u[on initially. It the changes was a result of the customer requests or adding new features or requirements there will be no problem in negotiation. But if the changes due to a problem in thedevelopment, the project manager needs to know what the cost of this change are and how much it will cost to delay the submission of the tasks. And if they will affect the delivery date an initial remedy may be to add some resources or postpone some other tasks for later sprints.

The CESP is built on the Scrum process. Not only it results in decrease for time and cost for project developments but also it handles the basic challenges of the scrum process. Sprint Workload and Testing in the Next Sprint. To overcome the Sprint Workload problem, our system, from the beginning of the project, addresses it, as it estimates the size of the sprint, and if it is longer than necessary, it migrates some tasks to another sprint based on the pre-determined conditions. By simulating the whole project daily scrum meetings an accurate assumption are given hence more accurate project plans. Estimating maintenance time is a challenge that is met by giving a probability of failure and account to it in the original sprint plan, when the load is not needed it can be used for tasks from later sprints. If more than expected maintenance time some of these tasks not on the sprint critical path can be halted and direct the resources to those urgent maintenance tasks. The CESP we recomputed the later sprints and revise the new project plan. To overcome the Testing in the Next Sprint problem, the CESP estimates the time needed for development and testing from the beginning. In CESP testing is a independent task that need a recourse like any other development activity. Nevertheless, internally a scrum master or project policy can insist that the developer must run his own testing task. And can also states that no sprint can ends and a development task was not tested. These rules may be relaxed to be applied on sprints that represent a delivery sprint.

H. *The CESP workflow*

The CESP workflow consists of three main phases: The project simulation phase, optimization phase, and the activation phase.

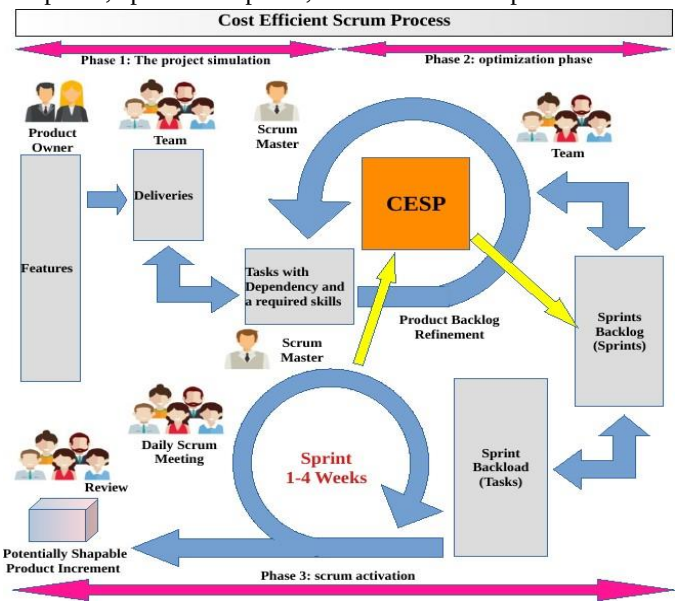


FIGURE 3COST EFFICIENT SCRUM PROCESS FRAMEWORK

A. *The project simulation phase*

In this phase the project team will engage in simulating a modified scrum process for the project development until the end of the project. The development team will

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

negotiate with the product owner a set of deliveries. A precise set of requirements will be attached for each delivery. The CESP tool will produce a set of sprints for each delivery.

Then, the development team will decompose each delivery requirement into tasks. For each task, the team will determine the skill needed, the estimated effort and the tasks it depends on. Record all these data in the CESP tool, which in turn will construct the lattice of all project tasks dependencies.

Starting from a single sprint for the delivery, the tool will add tasks to the latest delivery sprint, the chosen task will be from the lattice such that all other tasks it depend upon (if any) are already in this sprint or earlier sprints. When a task caused that sprint to exceed its capacity in any skill type, a new sprint is created and repeated the process until no task exist in the delivery Backlog and repeated for all deliveries.

Finally, the CESP tool will produce a set of deliveries each delivery is assigned a number of sprints, each sprint is assigned a number of tasks, and each task is assigned a skill and the estimated m/h needed for the task.

B. The optimization phase

This phase will involve the scrum team and the scrum master using the CESP tool. The objective of this phase is to move those tasks that can be moved to earlier sprints provided that there was some free resources available that was not fully utilized in that earlier sprints.

The CESP tool will rearrange those tasks and present the suggested change pending approval from the development team. Compute the saving per project, if any, resulting from the new change and iterate until there is no more optimization possible.

After that we fill up the tasks with sprints that do not depend on other tasks. Tasks that depend on tasks don in existence sprint or earlier sprints.

1- For each skill capacity in earlier sprint that exist, Search for task in later sprints that can use the available capacity if and only if its meet the dependencies required.
2- If there is a task that serves more than one delivery, we assign it to the nearest earlier eligible delivery.
3- For each change made adjust it's the sprint critical path and adjust the sprint end date and all later sprints started and finished date.
4- Repeated until there is no more optimization can be done.
5- Computer the overall cost in term of M/H of the whole project and report the change of the whole cost compared with the original cost.
6- If there is a human resource that is no longer needed for the development, it will be removed from the project hence decrease the cost of the development.

By the end of this phase the CESP will produce: sprints backlog composed of series of sprints, A sprint backlog, which is composed of a set of tasks that is contain of the current sprint, and an initial project plan that will show all tasks tell the end of the project hence allow the project manager to have a clear budget and monitor the execution of the whole project it term of cost and time.

A. *The* Scrum Activation phase

In the phase the normal scrum process will be followed with only minor changes. There might be some changes to the project plan such as adding new requirements, changes to current requirements, bug fixes or delay in any task execution for any reason may backlog refinement. Any such change will be done using the CESP tool to compute the effect of the change on the current sprint and latter sprints and the deliveries date and consequently on the overall cost of the project.

III. Experimentation and Results

We produced 2000 projects data randomly according to different agile benchmarks. Applied the normal scrum methodology of assigning requirements tasks to each sprint. The applied CESP methodology with the holistic project approach using its 3 phases and collected the results for comparison.

The experiment is conducted on a set of 2000 projects with data generated according to the benchmark. The average overall performance enhancement due to applying the use of the CESP is 17.42% which mean that the average cost per project has decreased by 17.42% for all 2000 projects tested and optimized.

Then analyzed the effect of changes in each parameter and its effect on the cost of the development of the project.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

1) The effect of the number of tasks per project

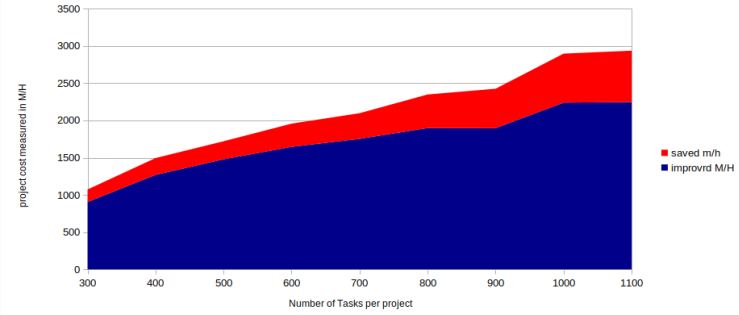


FIGURE 4THE EFFECT OF THE NUMBER OF TASKS PER PROJECT

The figure shows the optimized cost and the saved cost for projects with different number of tasks. And it is apparent that the decrease of cost is consistent for different project sizes in terms of the number of the overall tasks. The red area represents the saved M/H the blue are represent the cost of the project after applying the CESP optimization. The sum of the blue and the red area represent the original cost of the projects of different sizes (number of tasks per project.)

*1) . The effect of the change of the number **developers** per project*

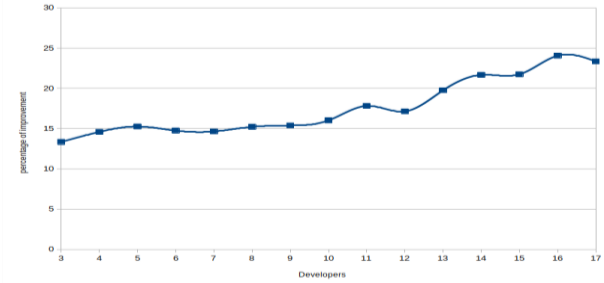


FIGURE 5 THE EFFECT OF THE CHANGE OF THE NUMBER OF DEVELOPERS PER PROJECT

This figure shows the relationship between the number of developers per project and the enhancement due to the use of CESP. Clearly it shows that the more developer available the more enhancement we get. That is due to the fact that the more developers exist in a project the chances the waste of resources occur. Consequently, the more waste of resources exist in normal scrum process the more chances that the CESP will be able to use this waste of resources hence fill up the earlier sprints with tasks moving many tasks earlier which will reduce the project time hence reduce time and cost.

2) The effect of increasing the direct dependent tasks

The direct dependent task is defined as the maximum number of tasks each task of the project will be allowed to be dependent upon. As a result, the bigger the number of direct dependent tasks the wider and shallower the dependencies lattice. And the less the number of direct dependencies the narrower and deeper the dependencies lattice.

This figure shows the decrease of cost related to the maximum number of dependent tasks for every task. It is clear that the performance enhanced when the number of dependent tasks is decreased. After careful examination we have found that the reason is that the more dependent tasks per every task the more likely that the chain of dependent tasks is longer. Hence it is harder or may be impossible to fit in earlier sprints that will reduce the chances for real improvement in time and cost. When the maximum number of direct sub-tasks decreased the string of dependent tasks becomes shorter that ease packing them into earlier sprints hence improving the usage of resources and decreases time and cost.

3) The effect of change of the number of Sprints per project

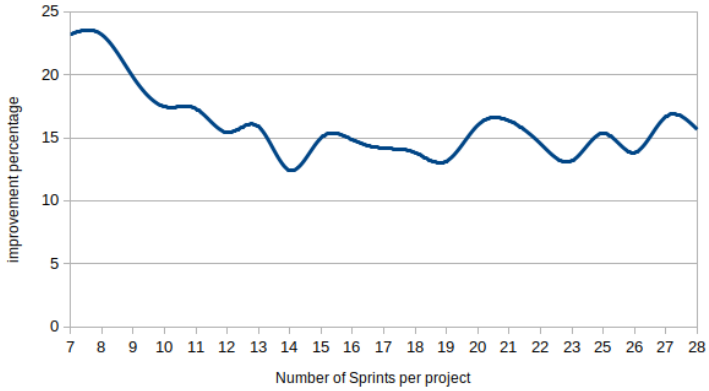


FIGURE 6THE RELATIONSHIP BETWEEN THE NUMBER OF SPRINTS AND IMPROVEMENT

In this figure we see the effect of the number of sprints on the improvement. On Average, the more sprints for a project the less enhancement the CESP can provide. The line is not straight because of the variability of the other factors. This general trend is due to that more sprints is required in many cases due to the long strands of dependent tasks. Which makes it very difficult to achieve big improvement due to the reallocation of tasks since many sprints end up with resources that is not used efficiently.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

4) The effect of the changes of the number of skills needed per project

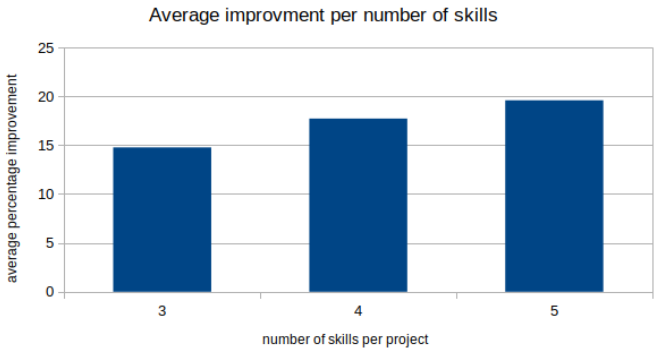


FIGURE 7RELATIONSHIPE BETWEEN THE SKILLS AND THE PROJECT IMPROVEMENT

It has been observed that the number of skills needed for the project will determine the improvement of using the CESP. The justification for that was that project that has more skills more likely to hit the maximum skill for a sprint hence leaving wasted resources that can be utilize better for later sprints and deliveries.

IV. ANALYSIS OF THE RESULTS

The bigger and more sophisticated the project the more reduction of cost we achieve using the CESP methodology. If we have a small project with few developers and few requirements using the CESP will not enhance the project efficiency over the normal Scrum process in fact it will be exactly the normal Scrum process if, for example we have one skill needed for the whole project, or if we have no dependencies among tasks.
It is strongly advised to use the VESP process if the project have many developers having a large variety of skills. Such projects probably have a large number of tasks and Sprints.

V. CONCLUSION

By emphasizing the planning for the whole project prior to any development and simulate the scrum process for the whole project putting into consideration the availability of the resources throughout the development phases, it was possible to devise a methodology that keep the agility and flexibility of the scrum processes while having a precise view of the development of the project. The methodology improve the efficiency of agile software development and it help to overcome the weaknesses of the previous methodologies and comprises all their strengths together. The validation of the proposed methodology has shown that an expected 20% reduction of cost of projects on average.

Another result of the study is the feasibility of using the given tool as a back office for lean methods such as Kanban..

REFERENCES

[1] Hohl, P., Klünder, J., Van,B.A., Lockard, R., Gifford, J., Münch, J., Stupperich, M.,&Schneider, K. (2018). Back to the future: Origins and directions of the "Agile Manifesto" – Views of the originators. Journal of Software Engineering Research and Development.

[2] Anwer, F., Aftab, S., Waheed, U.,&Muhammad, S.S. (2017). Agilesoftware development models TDD.FDD, DSDM, and Crystal Methods: A Survey.

[3] Kaushik, S., Bharadwaj, A., Awasthi, V.,&Sharma, R. (2017). Applicability and issues in traditional model of ERP implementations: an industry perspective. International Journal of Advanced Computer Research.

[4] Kulkarni, R.H., Padmanabham, P., Harshe, M., Baseer, K.K.,&Patil, P. (2017). Investigating agile adaptation for project development. International Journal of Electrical and Computer Engineering; Yogyakarta.

[5] Poth, A., Sasabe, S., Mas, A.,&Mesquida, A.L. (2019). Lean and Agilesoftware process improvement in traditional and Agileenvironments. Journal of Software: Evolution and Process.

[6] Qureshi, M.R.J. (2017). Evaluating the quality of proposed agile XScrum model. International Journal of ModernEducation and Computer Science.

[7] Walker W.Royce, "Managing the Development of Large software system" in Proc. IEEE WESTCON, Los Angeles, IEEE Computer Society Press, 1970, pp. 328-338.

[8] R. Green, T. Mazzuchi, S. Sarkani, "Communication and Quality in Distributed Agile Development: An Empirical Case Study," WASET, pp. 322-328, 2010.

[9] D. Boland, B. Fitzgerald, "Transitioning from a Col-located to a globally-distributed software development team: a case study at analog devices inc.," Proc. ICSE Workshop on Global Software Development, Scotland, pp. 4-7, 2004.

[10] Sutherland J, Schwaber K. The Scrum Guide. http://www.scrumguides. org/docs/scrumguide/v1/Scrum-Guide-US.pdf Accessed March 2, 2016; 2013.

[11] VersionOne. 9th annual state of agile development survey results. http://www.versionone.com/pdf/state-of-agile-development-surveyninth. pdf Accessed May 5, 2015; 2015.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

[12] Jinzenji K, Hoshino T, Williams L, TakahashiK.An experience report for software quality evaluation in highly iterative development methodology using traditional matrics. 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE);Novmeber 2013:310-319

[13] Cohn M. Succeeding With Agile: Software Development Using Scrum. 1st ed. Boston, MA, USA: Addison-Wesley Professional; 2009.

[14] Anderson DJ. Successful Evolutionary Change for Your Technology Business. 1st ed. Seattle,WA, USA: Blue Hole Press; 2010.

[15] Sutherland J, Schwaber K. The Scrum Guide. http://www.scrumguides. org/docs/scrumguide/v1/Scrum-Guide-US.pdf Accessed March 2, 2016; 2013.

[16] Schwaber K. Scrum is hard and disruptive. http://www. verheulconsultants.nl/ScrumIsHardandDisruptive.pdf Accessed.

[17] Abrahamsson P., Warsta J., Siponen M.T. And Ronkainen J.: New Directions on AgileMethods: A Comparative Analysis. In: Proceedings of the International Conference on Software Engineering, Portland, Oregon, USA, 2003 An empirical study of software metrics for assessing the phases of an agile project.

[18] A.H. Mohammad et al.," Agile Software Methodologies: Strength and Weakness,", Int. J. of Engineering Science and Technology, Vol. 5, No. 03, March 2003, pp. 455- 459.

[19] Tarwani, S.,&Chug, A. (2016). Agilemethodologies in software maintenance: A systematic review. Informatica; Ljubljana.

[20] Ashraf, S. and S. Aftab, Latest Transformations in Scrum: A State of the Art Review. International Journal of Modern Education and Computer Science, 2017. 9(7): p. 12.

[21] Diebold, P. and M. Dahlem. Agile practices in practice: a mapping study. in Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. 2014. ACM.

[22] Hron, M. and N. Obwegeser. Scrum in practice: an overview of Scrum adoptations.in Proceedings of the 51st Hawaii International Conference on systems scince 2018.

[23] Shenhar, A.J., et al., Project success: a multidimensional strategic concept. Long range planning, 2001. 34(6): p. 699-725.

[24] Pollack, J., J. Helm, and D. Adler, What is the Iron Triangle, and how has it changed? International Journal of Managing Projects in Business, 2018. 11(2): p. 527-547.

[25] Davis, K., Different stakeholder groups and their perceptions of project success. International journal of project management, 2014. 32(2): p. 189-201.

[26] Davis, K., Reconciling the Views of Project Success: A Multiple Stakeholder Model. Project Management Journal, 2018. 49(5): p. 38-47.

[27] Belassi, W. and O.I. Tukel, A new framework for determining critical success/failure factors in projects.

[28] Eloranta, V.-P., K. Koskimies, and T. Mikkonen, Exploring ScrumBut—An empirical study of Scrum anti- patterns. Information and Software Technology, 2016. 74: p. 194-203.

[29] C. Ladas, "Scrumban," Lean Softw. Eng. Contin. Deliv. High Qual.Inf.Syst.,2008.

[30] A. Reddy, The Scrumban [r] evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban. Addison-Wesley Professional, 2015.

[31] H. Dunsky, A case for Scrumban. 2014, pp. 1–10.

[32] J. Similä, M. Oivo, and K. Liukkunen, "Empirical Investigation of Scrumban in Global Software Development," in Model-Driven Engineering and Software Development: 4th International Conference, MODELSWARD 2016, Rome, Italy, February 19-21, 2016, Revised Selected Papers, 2017, vol. 692, p. 229.

[33] The 10th annual STATE of AGILE Report, VersionOneInc. 2016, http://stateofagile.versionone.com/

[34] The 9th annual STATE of AGILE Report, VersionOneInc. 2015, http://stateofagile.versionone.com/

[35] Pilar, "Combining Lean Thinking and Agile Software Development", University of Oulu, Finland, 2013

[36] Williams, "A survey of agile development methodologies,"2007.

[37] K. Bhavsar, V. Shah and S. Gopalan, "Scrumban: An Agile Transformation of Scrum and Kanban in Software Engineering", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, issue 4, pp.1626-1634, 2020. ISSN: 2278-3075, DOI: 10.35940/ijitee.D1566.029420.