

Optimal Resource Allocation for Software Development under Agile Framework

Adarsh Anand¹, Jasmine Kaur^{1*}, Ompal Singh¹, Mangey Ram²

¹Department of Operational Research, University of Delhi, Delhi-110007, India.

²Department of Mathematics, Computer Science & Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand-248002, India.

adarsh.anand86@gmail.com; jasminekaur.du.aor@gmail.com (*Corresponding Author);
drompalsingh1@gmail.com; drmrswami@yahoo.com

Abstract

Software development techniques have always been in-focus while creating high-quality software. Off late, most software firms have adopted the latest approach of the market- "Agile Software Development" to create a superior product. Agile software development allows an incremental and iterative development of the product with close interaction of all the teams on deck. This allows for the efficient utilization of the available resources to satisfy the demands of the end consumer. In the current proposal, a mathematical model has been created which allows optimal allocation of the limited developmental resources among the various sprint cycles and modules. Optimization tools have been used to determine this optimal distribution of resources. The proposed model has been validated on the resource allocation data with promising results.

Keywords: Agile Software; Budgeting; Optimization Method; Resource Allocation; Software Development.

I. Introduction

The rapidly progressing world of technology forces us, individuals, to keep moving ahead with the flow and one is left with no other option than evolve and keep up with the times. The same has happened with the field of software. Till a decade or so ago, there were a handful of software development approaches available. But with time not only has the number of approaches increased, but so many different models have evolved out of the basic few which cater to the diverse environments. Agile based software development is no different. It incorporates the basics of the waterfall model by using the various phases like the requirements analysis, design, coding, testing, and implementation. The add-on in agile based methodologies is that each of these phases can be re-visited repeatedly to incorporate the changing requirements. The core idea of building the basic prototype and enhancing it in various developmental cycles is the main principle in the prototype model and in the iterative and incremental model. The iteration or cyclical development approach is achieved in agile based methodologies, specifically in scrum, with the help of sprints. Each sprint is a time-bound (alternatively task-bound event) wherein a pre-decided number of tasks are performed in a given time frame. The incremental approach is implemented by developing the most essential and most characteristic features of the software in the first sprint. Each subsequent sprint

adds-on to this basic version by developing more items from the requirements list. The Verification and Validation model also called the V-model, designs test cases for each development phase thus making testing an essential step of each phase. The same principle is implemented in scrum wherein various types of testing take place throughout a particular sprint to ensure that the right product is being developed by building the product right. Continuous testing is achieved through Unit testing, Component Testing, Functional Testing, Exploratory Testing, User Acceptance Testing (Alpha, Beta, and Gamma Testing), Performance Testing, Load Testing, Security Testing, etc. performed throughout the project. What was missing in other approaches was the flexibility to change and deal with unforeseen circumstances and ever-changing requirements. This indispensable feature was inculcated in the agile principles which made it a go-to option for software development. Thus, it can be safely concluded that agile manifesto creators [1] were visionaries who not only identified the strengths and weaknesses of the existing models and inculcated them in their proposed methodology but also created a dynamic product that is still relevant two decades later. The principles proposed by them can be used to make any model, project, or sector agile. Frameworks like Scrum, Kanban, Lean, Test Driven Development, Feature Driven Development, Crystal, RAD, etc. are the popular ones based on the agile principles. The agile frameworks have gained popularity in recent times with them being utilized in numerous sectors besides software development like product development, distribution, healthcare, education, governance, construction, management, research, etc. [2-4]

The scrum methodology is the most used among the various agile based methodologies. The name itself draws focus to one of the key aspects of agile i.e. the highly interactive teams. The team consists of three classes of people: the Product Owner, the Scrum Master, and the Development Team. The product owner is the person who represents and conveys the wishes of the various stakeholders to the development team. The development team consists of the various members like the coders, the testers, the researchers, the analysts, etc. who build the product. The scrum master acts as a link between the product owner and the development team and provides them with the logistics to perform their job [5]. The team size is kept small and each member is multi-talented, cross-functional, and works in very close association with others. Daily scrum meetings, face-to-face interactions, open channels of communication, lack of hierarchy system within a team, retrospective meetings, etc. all promote a healthy work environment and an easy flow of information. All the team members are valuable resources within an organization. Hence, providing them with the right inputs and helping them learn and grow within an organization is essential.

Efficient project management helps in optimally allocating the tasks, assigning resources, planning the process, deciding the timeline, effort estimation, cost estimation, and resource estimation. Three main areas which require skillful planning are product backlog planning, sprint planning, and release planning. All of these activities require resources such as man-power, data, hardware components, software, office supplies, infrastructure, work amenities, etc. Most of these factors like manpower, technical aids, etc. are the direct contributors towards the success of the project while others like infrastructure and work amenities contribute to a comfortable and safe work environment which are essential for higher productivity. Software development process specifically focusing on software testing and software maintenance has been widely studied [6-9]. In the current proposal, the focus is on studying how the project money or budget can be used to achieve the best possible output. All the varied resources have been normalized and studied with a common denominator of money spent in acquiring, using, and maintaining them. Utilizing these resources optimally is essential to garnering profits without compromising on the product.

The resource allocation problem in the software testing and operational phases has been studied widely in the literature [10]. Huang and Lyu [11] have discussed optimal resource allocation throughout the software development cycle. Huang and Lo [12] focused on assigning testing resources in modular software. Inoue and Yamada [13] discussed an optimal testing effort allocation

problem. Anand et al. [14] allocated testing budget to various versions of the software. Bhatt et al. [15] assigned optimal resources to detect software vulnerabilities of different severity levels. Anand et al. [16] designed an optimization problem to optimally allocate resources to various software updates to remove software vulnerabilities.

Most resource allocation problems were defined keeping in mind the waterfall approach wherein the resources were assigned to the various phases of the development cycle. Considering the prevalence of agile based models, an alternate resource allocation method is required which considers the characteristic features of sprints in agile based models to make an optimal allocation. In the context of agile software development projects, not much work has been done with respect to resource allocation. The topic has been dealt indirectly through effort and budget estimation. Mahnic [17] discussed a case study on the learning patterns observed in effort estimation made by teams implementing scrum for the first time. Morris et al. [18] patented a method of resource allocation in an agile environment. Colomo-Palacios et al. [19] suggested a recommendation system to assign the human resource to the team using fuzzy and rough sets. Cao et al. [20] discussed the project funding and budgeting in an agile based project. Mansor et al. [21] discussed the budgeting challenges faced in agile based projects. Thom-Manuel et al. [22] discussed a budget model to incorporate risk management in agile software development. Ploder et al. [23] gave a Continuous Forecasting Framework to handle project budgeting in an agile project.

The approach to determine the effort consumed in an agile based project is to first breakdown the project into user stories and then determine the time required to successfully complete each user story. The story points, ideal days, etc. are outputs used to measure them [24]. Techniques such as analogous estimation, parametric estimation, Delphi method, Poker method, 3 Point Estimate, Expert Judgment, Published Data Estimates, Vendor Bid Analysis, Reserve Analysis, Bottom-Up Analysis, and Simulation are used to determine the same. Litoriya and Kothari [25] discussed cost and effort estimation for web-based projects through their proposed AgileMOW. Owais and Ramakishore [26] discussed cost, effort, and duration in agile software development. A systemic review of cost estimation in agile software development has been given by Bilgaiyan [27]. Usman and Petersen [28] compiled the available knowledge to create a taxonomy for effort estimation in agile based software development. Gultekin and Kalipsiz [29] used Machine Learning Algorithms to conduct story point-based effort estimation. Fernández-Diego et al. [30] provided extended literature on the work done in effort estimation in agile software development.

In the proposed work, a model has been discussed which allocates the available resources to the different sprints depending on the number of tasks that need to be performed in it. The current work has been organized in the following way: Sec. II discusses the notations used in model development. Section III proposes a model to allocate the resources in the development project. Section IV demonstrates the working of the model through an illustration and discusses the findings. Section V concludes the work and is followed by a list of the references used herein.

II. Notations

The notations used in the development of the model are as follows:

- n : Total number of sprints required to complete the project
- i : A counter variable representing a particular sprint, $i = 1, 2, \dots, n$
- $k_i(R_i)$: Cumulative number of tasks dealt using R_i resources in the i^{th} sprint of software
- $g_i(t)$: Probability Distribution function
- $G_i(t)$: Cumulative Distribution function, $F(t) = \int f(t) dt$
- a_i : Number of tasks in the i^{th} sprint of the software

- b_i, β_i : Model parameters
 R_i : Resources allocated to sprint i
 q_i : Aspiration level for sprint i
 c_B : Project development budget

III. Model Development

The resource allocation has been achieved through an optimization problem. The problem considers that there are a certain number of user stories defined in the product backlog which are to be developed in the various sprint cycles. Before the start of each sprint, the sprint backlog is decided in the sprint meeting which defines the sprint's task list.

To describe the task performance/completion phenomena in an agile environment, an analogy from the fault debugging phenomenon observed in software reliability literature has been drawn [10, 31-32]. The number of tasks being performed in a sprint can be treated as a counting process. A counting process is used to describe the occurrence of an event of time (here the completion of a task).

Let $N(T)$ represent the number of occurrences of an event in the interval $[0, t]$ with a mean value function $k(t)$. Let $v(t) = \frac{g(t)}{1-G(t)}$ denote the instantaneous rate at which tasks are performed, also referred to as the intensity function. Then, $N(T)$ is considered to be a Non-Homogenous Poisson Process (NHPP) if it satisfies the following properties:

- At the start of the process no tasks have been completed, i.e. at $t=0$, $N(0) = 0$.
 - $\{N(t), t > 0\}$ has independent increments.
 - $P\{N(t+\Delta t) - N(\Delta t) \geq 2\} = o(\Delta t)$, i.e. the probability of completing two or more tasks simultaneously is negligible.
 - $P\{N(t+\Delta t) - N(\Delta t) = 1\} = v(t) + o(\Delta t)$ denotes the probability of completing a single task.
- where $o(\Delta t)$ denotes a quantity that tends to zero for a small value of t .

Let $k(t)$ represent the mean number of tasks completed by time t , i.e. $k(t) = \int_0^t v(x) dx$, $t > 0$ then it can be shown that:

$$\Pr[N(t) = j] = \frac{(k(t))^j e^{-k(t)}}{j!}, \quad j = 0, 1, 2, \dots \quad (1)$$

i.e. $N(t)$ has a Poisson distribution with a mean value function $k(t)$. Consider the situation where the time scale is very large as compared to the total number of tasks to be completed and the mean number of tasks are completed at random time points. Hence, the cumulative number of tasks that are completed by a given time t can be defined using the following differential equation:

$$\frac{dk(t)}{dt} = v(t)(A - k(t)) \quad (2)$$

Equation (2) defines a direct relationship such that the number of tasks that will be performed by time t is dependent on the number of tasks that have not yet been performed and the intensity function. On solving equation (2) with the initial condition $t = 0, k(t) = 0$ i.e. at the start of the sprint, no tasks have been completed, the following expression is obtained:

$$k(t) = AG(t) \quad (3)$$

Here, the cumulative distribution function $G(t)$ can take different functional forms depending on the nature of the problem. In the agile framework, it is observed that with passing time, the velocity

or the efficiency of the team increases due to factors like improved project knowledge, better communication, commitment towards the project, etc. Hence, a logistic rate has been considered that denotes a learning pattern or an increased task completion rate with passing time. It is represented as:

$$v(t) = \left(\frac{b}{1 + \beta e^{-bt}} \right) \quad (4)$$

Using eq. (4) in eq. (3), the following expression is obtained:

$$k(t) = \frac{A(1 - e^{-bt})}{(1 + \beta e^{-bt})} \quad (5)$$

The above expression can be generalized for i sprints.

Using the work of Kapur et al. [10], the number of tasks completed in the sprint i by using resources R_i can be obtained as:

$$k_i(R_i) = \frac{A_i(1 - e^{-b_i R_i})}{1 + \beta_i e^{-b_i R_i}}, \quad i = 1, 2, \dots, n \quad (6)$$

I. Basic resource allocation assigning resources to various sprints

The simple resource allocation problem is designed which assigns the resources to each sprint while trying to maximize the number of tasks performed within a sprint. The problem is constrained such that the total budget is not exceeded and the non-negativity condition holds. Thus, problem P1 is defined as:

$$\begin{aligned} \text{Maximize } \sum_{i=1}^n k_i(R_i) &= \sum_{i=1}^n \frac{A_i(1 - e^{-b_i R_i})}{1 + \beta_i e^{-b_i R_i}} \\ \text{Subject to } \sum_{i=1}^n R_i &\leq C_B \\ R_i &\geq 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (P1)$$

Since the objective function is a fraction hence P1 is a fractional programming problem and to solve it, the following procedure is used:

$$\text{Let } g_i(R_i) = A_i(1 - e^{-b_i R_i}) \text{ and } h_i(R_i) = 1 + \beta_i e^{-b_i R_i} \quad (7)$$

$$\text{Let } H_i(R_i) = g_i(R_i) / h_i(R_i) \quad (8)$$

Using the work of Dur et al. [33] and Geoffrion's scalarization [34] formulation with fixed weights λ_i for the objective function, the problem can be reframed as:

$$\begin{aligned} \text{Maximize } \sum_{i=1}^n \lambda_i (g_i(R_i) - h_i(R_i)) \\ \text{Subject to } \sum_{i=1}^n R_i &\leq C_B \\ R_i &\geq 0, \quad i = 1, 2, \dots, n \\ \lambda &\in \Omega = (\lambda \in R^N / \sum \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, \dots, n) \end{aligned} \quad (P2)$$

II. Modified problem to include aspiration constraint in each sprint

The number of tasks that can be performed using the assigned resources may vary from sprint to sprint. Hence, an aspiration is assigned to each sprint such that at least a certain proportion of the task list is completed. The proportion can be varied from sprint to sprint depending on the sprint goal. Hence, the new optimization problem P3 is:

$$\begin{aligned} \text{Maximize } \sum_{i=1}^n k_i(R_i) &= \sum_{i=1}^n \frac{A_i(1 - e^{-b_i R_i})}{1 + \beta_i e^{-b_i R_i}} \\ \text{Subject to } k_i &\geq A_{i_0} \\ \sum_{i=1}^n R_i &\leq C_B \\ R_i &\geq 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (P3)$$

where $A_{i_0} = q_i A_i$ denotes the number of tasks that need to be performed to meet the sprint aspiration.

Since $k_i \geq A_{i_0}$

$$\Rightarrow \frac{A_i(1 - e^{-b_i R_i})}{1 + \beta_i e^{-b_i R_i}} \geq A_{i_0} \quad (9)$$

$$\Rightarrow R_i \geq -\frac{1}{b_i} \log \left[\frac{1 - \frac{A_{i_0}}{A_i}}{1 + \frac{A_{i_0}}{A_i} \beta_i} \right] = Z_i \quad (10)$$

where Z_i denotes the minimum amount of resources required to perform A_{i_0} tasks. Using the approach discussed in subsection I, the fractional programming problem (P3) is transformed to obtain (P4) as:

$$\begin{aligned} \text{Maximize } \sum_{i=1}^n [(A_i - 1) - (A_i + \beta_i) e^{-b_i R_i}] \\ \text{Subject to } R_i &\geq Z_i \\ \sum_{i=1}^n R_i &\leq C_B \\ R_i &\geq 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (P4)$$

The number of tasks that could be completed by satisfying the sprint aspiration is obtained by solving the problem (P4) as:

$$k_i(Z_i) = \frac{A_i(1 - e^{-b_i Z_i})}{1 + \beta_i e^{-b_i Z_i}}, \quad i = 1, 2, \dots, n \quad (11)$$

III. Resource allocation to meet the overall project aspiration constraint

The sprint aspiration will help achieve the sprint goal but to meet the project goal another aspiration is required. Thus, the resources left over after meeting the sprint aspiration will be reallocated to meet the project aspiration. For this, the problem can be modified in the following way:

$$\text{Let } S_i = R_i - Z_i; \quad \bar{Z} = C_B - \sum_{i=1}^n Z_i \quad \text{and} \quad \bar{A}_i = A_i - A_{i_0} \quad i = 1, 2, \dots, n \quad (12)$$

Then,

$$\begin{aligned} & \text{Maximize } \sum_{i=1}^n [(\bar{A}_i - 1) - (\bar{A}_i + \beta_i) e^{-b_i S_i}] \\ & \text{Subject to } \sum_{i=1}^n S_i \leq \bar{Z} \\ & S_i \geq 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (P5)$$

The optimal number of tasks that could be completed in meeting the project goal is obtained by solving the problem (P5) and is given as:

$$k_i(S_i^*) = \frac{\bar{A}_i(1 - e^{-b_i S_i^*})}{1 + \beta_i e^{-b_i S_i^*}}, \quad i = 1, 2, \dots, n-1 \quad (13)$$

Hence, the total optimal number of tasks that could be completed in a particular sprint i of the project is given as:

$$k_i(R_i) = k_i(S_i^* + Z_i), \quad i = 1, 2, \dots, n \quad (14)$$

IV. Model Illustration

For numerical illustration, a simulated agile based software development dataset has been considered. Let the project contain 175 tasks to be performed in 8 sprints of 1 month each. The project budget to be allocated is assumed to be \$200,000. The minimum aspiration of task completion from each sprint is 65% i.e. $q_i = 0.65$. The minimum aspiration of task completion from the whole project is 85%. SAS software [35] has been used to estimate the model parameters for each sprint (using eq. 5) and has been demonstrated in Table 1. LINGO software [36] has been used to solve optimization problems (P1-5).

The aim was to utilize the minimum possible resources, hence in the initial round of assignment, 80% of the available budget was allocated i.e. $C_b = \$160,000$. On solving the optimization problem P2, the basic allocation was obtained and the number of tasks completed in each sprint and the overall project was determined. As can be seen in Table 1, a sequential distribution of resources was occurring which results in lesser resources being available for the last few sprints and a low task completion rate. In sprint 1, almost all the tasks could be completed but in the last sprint, only 32% (approx.) tasks could be performed. Despite this, the overall project task completion was quite high at 84% (approx.).

Table 1: Parameter Estimates and initial resource allocation for sprints

| Sprint No. | A_i | b_i | β_i | R_i | $k_i(R_i)$ | % of Tasks Performed | % of Tasks Remaining |
|-----------------|-------|-------------|-------------|----------|------------|----------------------|----------------------|
| Sprint 1 | 45 | 0.000412932 | 0.170795673 | 10921.64 | 44.42 | 98.71 | 1.29 |
| Sprint 2 | 13 | 0.000319987 | 0.129642265 | 9435.735 | 12.29 | 94.52 | 5.48 |
| Sprint 3 | 16 | 0.000264216 | 0.124115883 | 11480.16 | 15.14 | 94.62 | 5.38 |
| Sprint 4 | 35 | 0.000150112 | 0.26584682 | 21653.54 | 33.30 | 95.14 | 4.86 |
| Sprint 5 | 14 | 8.95707E-05 | 0.256627137 | 20413.03 | 11.29 | 80.61 | 19.39 |

| | | | | | | | |
|-----------------|-----|-------------|-------------|-------------------|--------|--------------|-------|
| Sprint 6 | 21 | 5.87323E-05 | 0.324640162 | 30800.98 | 16.67 | 79.40 | 20.60 |
| Sprint 7 | 20 | 3.20165E-05 | 0.69718776 | 36618.9 | 11.36 | 56.78 | 43.22 |
| Sprint 8 | 11 | 3.15115E-05 | 0.728208159 | 18676.01 | 3.48 | 31.68 | 68.32 |
| | 175 | | | 159999.995 | 147.95 | 84.54 | 15.46 |

Thus, the need for evaluating problem P4 arises which satisfies the sprint aspiration. As can be seen in Table 2, out of the \$160,000 budget, \$156,880 was spent in meeting the sprint aspiration of 65%. The least task completion percentage was observed in sprint 7. The leftover budget of \$3,120 was allocated by solving problem P5. But since the leftover budget is quite low, the project aspiration of 85% product backlog task completion could not be met and the rate fell to 74% (approx.).

Table 2: Minimum aspiration of 65% task completion in each sprint

| Sprint No. | A_i | A_{i0} | Z_i | $k_i(Z_i)$ | \bar{A}_i | S_i | $k_i(S_i)$ | R_i | $k_i(R_i)$ | % of Tasks Performed | % of Tasks Remaining |
|-----------------|-------|----------|--------------------|------------|-------------|------------------|------------|--------------------|------------|----------------------|----------------------|
| Sprint 1 | 45 | 30 | 2921.6606 | 30 | 15 | 3006.7590 | 10.1645 | 5928.4196 | 40.5109 | 90.02 | 9.98 |
| Sprint 2 | 13 | 9 | 3952.0526 | 9 | 4 | 0.0000 | 0.0000 | 3952.0526 | 9.0000 | 69.23 | 30.77 |
| Sprint 3 | 16 | 11 | 4712.1847 | 11 | 5 | 0.0000 | 0.0000 | 4712.1847 | 11.0000 | 68.75 | 31.25 |
| Sprint 4 | 35 | 23 | 8203.5656 | 23 | 12 | 114.0340 | 0.1615 | 8317.5996 | 23.1892 | 66.26 | 33.75 |
| Sprint 5 | 14 | 10 | 15865.3937 | 10 | 4 | 0.0000 | 0.0000 | 15865.3937 | 10.0000 | 71.43 | 28.57 |
| Sprint 6 | 21 | 14 | 22041.1933 | 14 | 7 | 0.0000 | 0.0000 | 22041.1933 | 14.0000 | 66.67 | 33.33 |
| Sprint 7 | 20 | 13 | 44463.6643 | 13 | 7 | 0.0000 | 0.0000 | 44463.6643 | 13.0000 | 65.00 | 35.00 |
| Sprint 8 | 11 | 8 | 54719.4874 | 8 | 3 | 0.0000 | 0.0000 | 54719.4874 | 8.0000 | 72.73 | 27.27 |
| | 175 | 118 | 156879.2022 | 118 | 57 | 3120.7930 | 10.3260 | 159999.9952 | 128.7001 | 73.54 | 26.46 |

After running simulations it was realized that an additional \$25,000 was required to meet the 85% project completion rate while maintaining the 65% aspiration level from each sprint. Table 3 shows the results obtained by evaluating problem P5 with this increased budget. It can be seen that here both the aspirations were fulfilled.

Table 3: Minimum aspiration of 65% task completion in each sprint and 85% in the whole project

| Sprint No. | A_i | A_{i0} | Z_i | $k_i(Z_i)$ | \bar{A}_i | S_i | $k_i(S_i)$ | R_i | $k_i(R_i)$ | % of Tasks Performed | % of Tasks Remaining |
|-----------------|-------|----------|--------------------|------------|-------------|-----------------------|------------|------------|------------|----------------------|----------------------|
| Sprint 1 | 45 | 30 | 2921.6606 | 30 | 15 | 6785.121 | 13.9449 | 9706.7816 | 44 | 97.78 | 2.22 |
| Sprint 2 | 13 | 9 | 3952.0526 | 9 | 4 | 3892.634 | 2.7465 | 7844.6866 | 12 | 92.31 | 7.69 |
| Sprint 3 | 16 | 11 | 4712.1847 | 11 | 5 | 4806.122 | 3.4745 | 9518.3067 | 15 | 93.75 | 6.25 |
| Sprint 4 | 35 | 23 | 8203.5656 | 23 | 12 | 10507.74 | 9.0261 | 18711.3056 | 32 | 91.43 | 8.57 |
| Sprint 5 | 14 | 10 | 15865.3937 | 10 | 4 | 29.3288 | 0.0084 | 15894.7225 | 10 | 71.43 | 28.57 |
| Sprint 6 | 21 | 14 | 22041.1933 | 14 | 7 | 2099.849 | 0.6311 | 24141.0423 | 15 | 71.43 | 28.57 |
| Sprint 7 | 20 | 13 | 44463.6643 | 13 | 7 | 0 | 0.0000 | 44463.6643 | 13 | 65.00 | 35.00 |
| Sprint 8 | 11 | 8 | 54719.4874 | 8 | 3 | 0 | 0.0000 | 54719.4874 | 8 | 72.73 | 27.27 |
| | 175 | 118 | 156879.2022 | 118 | 57 | 28120.794 8 | 29.8315 | 184999.997 | 149 | 85.1429 | 14.8571 |

Thus, the two goals of project and sprint task completion aspiration could be achieved with \$185,000 and net savings of 7.5% of the allocated budget of \$200,000. It will be a managerial decision to either invest the leftover resources in achieving an even higher output or let it be the project cost saving. Hence, it is a trade-off between how much one wants to invest to get a higher completion rate. A rational choice needs to be made here and the agile principle of developing only the most essential features should be applied in determining the level of output desired. In this model illustration, the aspiration level was considered to be the same for each sprint. But this can be varied for different sprints. By running different simulations of the model and varying the input and the desired level of output, the best fit for a particular project/firm can be obtained.

V. Conclusion

Agile based methodologies have gained prominence and are the most used software development approach in the current time. Since all the developmental resources are usually limited hence there optimal utilization is necessary. In the current proposal, a resource allocation problem in agile based software has been discussed through optimization problems. Three different problems were discussed wherein the best fit can be chosen as per the managerial decisions. A basic problem determined how much of the available resources need to be spent in each sprint. The second problem was able to determine how many resources would be required in each sprint to complete a minimum proportion of sprint backlog. The third problem further extended the second problem and was able to meet the product backlog aspiration along with the sprint aspiration. Hence, the current proposal provides a simple yet effective mathematical tool to determine the optimal allocation of resources for maximum yield. The management can decide on how much to invest in the project by correlating

with the gains achieved in terms of the tasks completed.

Acknowledgement: The second author would like to thank the Ministry of Minority Affairs (UGC) for financial assistance provided to carry out this research work under the Maulana Azad National Fellowship scheme (UGC-Ref. No.: 190520033186, No. F. 82-27/2019 (SA-III)).

References

- [1] Beck, K., Beedle, M., Bennekum, A. Van, Cockburn, A., Cunningham, W., Fowler, M.,... and Kern, J. (2001). The agile manifesto.
- [2] Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time, Random House Business Books.
- [3] Pasic, L., Mladjenovic, M., García, A. C. and Aggrawal, D. (2017). Significant factors of the successful lean six-sigma implementation, *International Journal of Mathematical, Engineering and Management Sciences*, 2(2): 85-109.
- [4] Kumar, M. Garg, D. and Agarwal, A. (2019). An analysis of inventory attributes in leagile supply chain: cause and effect analysis. *International Journal of Mathematical, Engineering and Management Sciences*, 4(4): 870-881.
- [5] Sutherland, J. and Schwaber, K. (2013). The scrum guide. The definitive guide to scrum: The rules of the game, Scrum.org, 268.
- [6] Bhatt, N., Anand, A., Yadavalli, V. S. S. and Kumar, V. (2017). Modeling and characterizing software vulnerabilities, *International Journal of Mathematical, Engineering and Management Sciences*, 2(4): 288–299.
- [7] Anand, A., Bhatt, N. and Aggrawal, D. (2020). Modeling Software Patch Management Based on Vulnerabilities Discovered, *International Journal of Reliability, Quality and Safety Engineering*, 27(02): 2040003.
- [8] Das, S., Anand, A., Agarwal, M. and Ram, M. (2020). Release time problem incorporating the effect of imperfect debugging and fault generation: an analysis for multi-upgraded software system, *International Journal of Reliability, Quality and Safety Engineering*, 27(02): 2040004.
- [9] Kaur, J., Anand, A. and Singh, O. (2019). Modeling Software Vulnerability Correction/Fixation Process Incorporating Time Lag, *Recent Advancements in Software Reliability Assurance*, eds. Anand, A. and Ram, M., CRC Press, 39-58.
- [10] Kapur, P. K., Pham, H., Gupta, A. and Jha, P.C. (2011). *Software reliability assessment with OR applications*, London: Springer.
- [11] Huang, C. Y. and Lyu, M. R. (2005). Optimal testing resource allocation, and sensitivity analysis in software development, *IEEE Transactions on Reliability*, 54(4): 592-603.
- [12] Huang, C. Y. and Lo, J. H. (2006). Optimal resource allocation for cost and reliability of modular software systems in the testing phase. *Journal of Systems and Software*, 79(5): 653-664.
- [13] Inoue, S. and Yamada, S. (2018). Optimal Software Testing Effort Expending Problems. *System Reliability Management: Solutions and Technologies*, eds. Anand, A. and Ram, M., Boca Raton, CRC Press, 51-64.
- [14] Anand, A., Das, S., Singh, O. and Kumar, V. (2019). Resource allocation problem for multi versions of software system. In *Proc. of Amity International Conference on Artificial Intelligence, AICAI, IEEE*, 571-576.
- [15] Bhatt, N., Anand, A. and Aggrawal, D. (2019). Improving system reliability by optimal allocation of resources for discovering software vulnerabilities, *International Journal of Quality and Reliability Management*, 37(6/7): 1113-1124. <https://doi.org/10.1108/IJQRM-07-2019-0246>.
- [16] Anand, A., Kaur, J., Gokhale, A. A. and Ram, M. (2020). Impact of available resources on software patch management. *Systems Performance Modeling*, eds. Anand A. and Ram M., Walter de Gruyter GmbH & Co KG., 4: 1-11.

- [17] Mahnic, V. (2011). A case study on agile estimating and planning using scrum. *Elektronika ir Elektrotechnika*, 111(5): 123-128.
- [18] Morris, S., Kroeger, K., Mcguire, T. and Nikolaev, I. (2012). U.S. Patent No. 8,332,251. Washington, DC: U.S. Patent and Trademark Office.
- [19] Colomo-Palacios, R., González-Carrasco, I., López-Cuadrado, J. L. and García-Crespo, Á. (2012). ReSySTER: A hybrid recommender system for Scrum team roles based on fuzzy and rough sets. *International Journal of Applied Mathematics and Computer Science*, 22(4): 801-816.
- [20] Cao, L., Mohan, K., Ramesh, B. and Sarkar, S. (2013). Adapting funding processes for agile IT projects: an empirical investigation. *European Journal of Information Systems*, 22(2): 191-205.
- [21] Mansor, Z., Razali, R., Yahaya, J., Yahya, S. and Arshad, N. H. (2016). Issues and challenges of cost management in agile software development projects. *Advanced Science Letters*, 22(8): 1981-1984.
- [22] Thom-Manuel, O., Ugwu, C. and Onyejebu, L. (2017). An Extended Agile Software Development Project Budget Model, *International Journal of Computer Science and Software Engineering*, 6(12): 306-314.
- [23] Ploder, C., Dilger, T. and Bernsteiner, R. (2020). A Framework to Combine Corporate Budgeting with Agile Project Management. In *Software Engineering (Workshops)*.
- [24] Cohn, M. (2005). *Agile estimating and planning*, Pearson Education.
- [25] Litoriya, R. and Kothari, A. (2013). An efficient approach for agile web based project estimation: AgileMOW, *Journal of Software Engineering and Applications*, 06(06): 297-303.
- [26] Owais, M. and Ramakishore, R. (2016). Effort, duration and cost estimation in agile software development. In *Proc. of Ninth International Conference on Contemporary Computing, IC3, IEEE, 2016*, 1-5.
- [27] Bilgaiyan, S., Sagnika, S., Mishra, S. and Das, M. (2017). A Systematic Review on Software Cost Estimation in Agile Software Development. *Journal of Engineering Science & Technology Review*, 10(4): 51-64.
- [28] Usman, M., Börstler, J. and Petersen, K. (2017). An effort estimation taxonomy for agile software development. *International Journal of Software Engineering and Knowledge Engineering*, 27(04): 641-674.
- [29] Gultekin, M. and Kalipsiz, O. (2020). Story Point-Based Effort Estimation Model with Machine Learning Techniques. *International Journal of Software Engineering and Knowledge Engineering*, 30(01): 43-66.
- [30] Fernández-Diego, M., Méndez, E. R., González-Ladrón-De-Guevara, F., Abrahão, S. and Insfran, E. (2020). An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review. *IEEE Access*, 8: 166768-166800.
- [31] Rawat, S., Goyal, N. and Ram, M. (2017). Software reliability growth modeling for agile software development. *International Journal of Applied Mathematics and Computer Science*, 27(4): 777-783.
- [32] Anand, A., Kaur, J., Singh O. and Alhazmi, O. H. (2021). Optimal sprint length determination for agile-based software development. *Computers, Materials & Continua*, 68(3): 3693–3712.
- [33] Dur, M., Horst, R. and Thoai, N. V. (2001). Solving sum-of-ratios fractional programs using efficient points. *Optimization*, 41: 447–466.
- [34] Geoffrion, M. (1968). Proper efficiency and theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22: 613–630.
- [35] SAS Institute Inc., *SAS/ETS user's guide version 9.1*. Cary, NC: SAS Institute Inc. (2004).
- [36] Lindo System, Inc. (2020).