

An Optimized Agile Estimation Plan Using Harmony Search Algorithm

C. Sathish Kumar^{#1}, A. Anitha Kumari^{*2}, R. Srinivasa Perumal^{#3}

School of Information Technology & Engineering,
VIT University, Vellore

¹csathishkumarmse@yahoo.in

²anithakumari.a@vit.ac.in

³r.srinivasaperumal@vit.ac.in

Abstract—Agile Software development methodology is most emerging in software development, there has been a significant contribution from many researchers towards estimating the size, cost, schedule, effort and duration. However, the approaches reported in the literature lack in considering the evolution of biological systems for better optimization in agile. Here Agile Estimation Plan method determines the story size but the velocity is not properly defined and Harmony Search algorithm is adopted and modified in improvisation process towards agile to calculate the better velocity for each end of the iteration plan. This method help to improve the optimization while compare with normal agile process. The experimental results show how the harmony agile process is better than the normal agile process and it shows the efficiency of the method.

Keyword- Swarm Intelligence, Harmony Search, Optimization, Agile Software Development.

I. INTRODUCTION

Agile is one of the upcoming trends in software development methods. Many organizations have already implemented the Agile process in development and others are willing to switch to agile based projects because [1]

- It improves the process of current development process;
- It guides to perform proper planning and estimation;
- It guides to handle the risk management;
- It helps to develop the project at reduced cost;
- It delivers the quality product to the customers within the specified deadline of the project;
- It improves the standard of your organization;
- It gives the return – on – investment for the project and also to the organization.

Estimation and planning are the two major activities which drive the agile projects [2]. No agile project is successful without proper estimation and planning. However, it is difficult to carry out these as agile development is an iterative process and the requirements planned before the start of a project, may change as the project progresses. Planning and estimation become more difficult to perform, as the requirements may change at any stage of the project cycle. Tracking the features planned for implementation in iteration and features actually implemented are necessary to get accurate estimates. An iteration plan is developed based on the work accomplished in the last finished iteration.

In Agile Software Development (ASD), the size of the application can be measured based on the features in a particular iteration through function points or story points, but the overall size of an application cannot be determined at the start of a project as the requirements change or are continuously added and the number of iterations cannot be decided *a priori*. So the optimization is needed at all the iterations throughout the project development cycle to screw up the dynamic process. The optimization is focused based on bio inspired algorithm.

A. Motivation of Swarm Intelligence in Agile Software Development

Swarm intelligence (SI) is a collection of bio inspired nature of process and it is a subset of artificial intelligence. The nature of implementing SI is based on characteristics of insects of interacting agents. The example of swarm or insect is a group of ants and its agents are ants. Swarms are based on adaptive nature of patterns and their result will be predicted randomly. There are many literature shows swarm intelligence is applied to get an optimization result in various domains which are adaptive in nature [6, 7, 8]. Harmony Search is one of the SI algorithm, the principles of SI and agile are similar in nature. Some of the SI principles are adaptive, iterative, population based algorithms, quality of outcomes, parameter tuning, monitoring, controlling and proximity.

ASD is an effective process of implementing successful delivery of the projects. It has been widely used in different product delivery of an organization. The agile principles were wrote by agile manifesto authors and their value based on Individuals and interactions at all stages of the project, customer collaboration at each end of the iteration and responding to change based on the customer requirements has to be reflected at any stage. Agile team has to concentrate on: analyzing, developing and testing a product as a team, the functionality has to be developed and delivered in a short iteration based on the customer priorities and also at any stage the system should be in a position to regress and accept the dynamic changes given by the customer [2].

In this paper, the proposed method shows how the optimization can be achieved by the adaptation of harmony memory search algorithm and predict size and effort of an application. In Section 2 explores the process of agile estimation method. In Section 3 deals about the optimization of agile estimation plan through harmony search algorithm. In Section 4 contains the experimental results of normal agile process and harmony agile process. Finally Section 5 gives the conclusion and related future work.

II. AGILE ESTIMATION METHOD

Estimation of agile project ensures the successful product or project delivery. In agile development we are estimating the parameters like size, effort, cost, duration etc. This paper concentrates on two parameters that is size and effort. The size is calculated by using story points and the effort is calculated based on the number of story points completed per iteration and number of employees in particular user story. Capability of the team taken to implementation of a user stories within a specified time is called iteration. As we know the agile methodology is iterative and incremental approach, so the initial plan as to be modified at every end of iterations based on number of story points finished in current iteration (velocity). The total number of story points and velocity determines the total number of iterations for any agile development and the current iteration velocity is updated to all the remaining iterations. So the velocity is the only factor that determines the proper optimization at final stage of development.

A. Story Points

A Story Point is a high-level definition of requirements or features for the software to be developed given by the user. A software application has many features and each feature is defined as one user story. The story point contains the needs of customer, but it doesn't have the structure. Story Point metric is used to measure the size, which is needed for software to implement all user stories. The weightage or points assigned for each user stories are predicted by using Fibonacci series called story points. A story point can have many iterations depending upon the application requirements [2]. The overall size needed to develop a particular project cannot be estimated in one attempt. The example of estimating user story with story size explained below,

User story: As a project manager I need to develop user authentication for ATM system.

Story point = 5 (predicted by Fibonacci sequence)

The drawback of the existing system, story point is used for determining the size of the user story, the method doesn't able to predict the effort, because story point value is not sufficient to count the proper value. To overcome the above problem, the agile estimation plan predicts the size of a user stories by considering three parameters including story points.

- Story Point (Fibonacci series value) (SP)
- Implementation Level Factor (ILF)
- Complexity (Com).

In agile estimation plan each iteration velocity will not be same because of the incremental approach of agile process. So it may gradually increase or decrease at any iteration. But we will not be in a position to get an idea of the resource utilization required for estimating the size and effort in the development process to match the initial plan with the final plan. To solve the above challenging issues, one may have to analyze and predict the velocity value at beginning stage of an iteration that would take the optimized value at all other iterations. The proposed model, adopted HS algorithm in agile process to solve the above mentioned optimization problem.

III. OPTIMIZED AGILE ESTIMATION METHOD

This method is used to estimate the story size and effort from the user story. In this paper, three parameters (SP, ILF, and Complexity) are used to calculate the size of a story.

A. Story Points

The user story size is determined by based on the nature of the stories either it's small, medium or large. Based on the categorization of story size the weights are assigned to estimating the size of user stories. Table I represents the weight for each description of story point.

TABLE I
Weight of Story Point

Parameter	Description	Weight
Story Point (SP)	Very small story	1
	Small story	2
	Medium story	3
	Large story	5
	Very large story	8

B. Implementation Level Factor

The level of understanding of each user story is different from one person to another is categorized by some factors like off the shelf and components. These factors are called as ILF (Implementation Level Factor). In Table II represents the weight for each description of ILF.

TABLE II
Weight of Implementation Level Factor

Parameter	Description	Weight
Implementation Level Factor (ILF)	Off the Shelf	1
	Full experience components	2
	Partial experience components	3
	New Components	4

C. Complexity

Complexity has been determined based on the user stories in ILF. For example, if the user story is very low and ILF is off the shelf then complexity may be very low. In Table III represents the weight for each description of Complexity.

TABLE III
Weight of Complexity

Parameter	Description	Weight
Complexity (Com)	Very low	1
	Low	2
	Medium	3
	High	4
	Very high	5

Story size can be estimated using below equation.

$$\text{Story size} = \text{SP} * \sum_{i=1}^4 \text{ILF} * \sum_{i=1}^5 \text{Com}$$

For Example,

- User story : As a project manager I need to develop user authentication for ATM system.
- SP : 5 (Large story)
- ILF : 2 (Full experience components)
- Com : 3 (Medium)

$$\text{Story Size} = 5 * 2 * 3 = 30.$$

The above method determines the story size with the help of Story Point, Implementation Level Factor and Complexity. The story size is optimized but velocity is not properly predicted; So Harmony Search algorithm principles are applied in agile process for adjusting the velocity to determine the better optimization.

D. Harmony Search Algorithm (HS)

HS is a meta-heuristic algorithm and this new population based algorithm is used to simulate the musical creativeness process. HS algorithm is mainly used for solving optimization problems in all the domains. The steps involved in the optimization procedure of the HS meta-heuristic algorithm as follows,

- Start the optimization problem and initialize the parameters involved in algorithm.
- Initialize and set up the harmony memory (HM).
- Improve a new harmony from the HM.
- Update the Harmony Memory based on the new HM.
- Repeat step 3 and step 4 up to the termination condition is satisfied [9].

Some of the applications of HS algorithms are,

- The HS algorithm is adopted in Nurse Scheduling problem [10]
- Adopted in construction projects resource constrained scheduling problem [11]
- Applied in university time tabling problem [12].

Table IV represents the comparison between agile and HS algorithm process.

TABLE IV
Comparison between Harmony Search algorithm and Agile process.

Harmony Search Algorithm		Agile process	
Terms	Description	Terms	Description
HMS (Harmony Memory Size)	A number of solution vectors in Harmony Memory	MF (Matrix Formation)	It indicates the iteration length of an application in the solution vector.
HMCR (Harmony Memory Consideration Rate)	parameter used to improve the solution vectors	SPCR (Story Point Consideration Rate)	Identifying new velocity rate from the current iteration for the next iteration.
PAR(Pitch Adjustment Rate)	parameter used to improve the solution vectors	VAR(Velocity Adjustment Rate)	Velocity value at each latest finished iteration identified by SPCR can be considered as a Pitch adjustment rate for the next remaining all iterations.
NI(Number of Improvisation)	Number of improvisation in HMS	TNIN(Total Number of Iterations Needed)	Iteration required for an entire application.
RAN (Randomization)	Denotes the Bandwidth	HV (Historical Velocity)	It is needed at the initial stage of the process for identifying / picking the first iteration value and the same value will be used for all remaining iteration.
		LV (Latest Velocity)	It is the new velocity generated from the latest / current iteration. Now this latest velocity will be replaced to all other remaining iteration for Historical value.

E. Estimation and Tracking Size, Effort In Agile Using HS Algorithm

The following steps used in agile estimation with the use of HS algorithm for attain a better optimization,

- The user stories are collected from users, then the initial plan is prepared for agile development, first user stories are assigned with story size by using following formula

$$\text{Story size} = SP * \sum_{i=1}^4 ILF * \sum_{i=1}^5 Com$$

- Initialize the team size based on the number of stories given by users. Find the total number of story size by using following formula,

$$\text{Total number of story size} = \text{sum of all story size}$$

- By using randomize function generate the initial velocity value for an application to decide the first iteration value.

$$\text{Velocity} = \text{no of story size delivered in the current iteration}$$

- Find the total number of iterations required for an entire application based on the initial iteration velocity value and generate the initial plan.

$$\text{Total no of Iterations} = (\text{Total no of story size} + \text{initial velocity} - 1) / \text{initial velocity}$$

- The every finishing of iteration, get the previous iteration velocity and adjust the velocity using HS algorithm. The adjustment is done by taking an average on previous iteration velocity and current iteration velocity, the average value is considered as a new velocity.

$$\text{New velocity} = \text{previous iteration velocity} + \text{current iteration velocity} / 2$$

- Calculate the remaining story size at each end of the iteration by using the following formula,(Initially the remaining story size value is total no of story size)

$$\text{Remaining story size} = \text{Remaining story size} - \text{new velocity}.$$

- If new story is needed in between iterations we can add and calculate the new story size by using the formula defined in step 1, or else just skip this step. (If no story needed new story size value is “0”).
- Find the total number of story size, remaining story size and new Total no of iterations by using following formulas,

$$\begin{aligned}
 \text{Total no of story size} &= \text{new story size} + \text{total no of story size} \\
 \text{Remaining story size} &= \text{Remaining story size} + \text{new story size} \\
 \text{Total no of Iterations} &= \text{completed iteration} + [(\text{Remaining story size} + \text{new velocity} - 1) / \text{new velocity}].
 \end{aligned}$$

Where completed iteration is no of iteration are previously completed (For first iteration the completed iteration value is “0”).

- Now replace the new velocity to remaining iterations and check the final condition as “if number of story size completed is equal to total number of story size”. If the condition is true terminate the process else go to step 5.
- Compare the initial plan with final plan.

IV. EXPERIMENTAL RESULT

The result analysis is done by considering simple application called STUDENTS LOGIN for college students. To develop this application some of the user stories are collected from the users. The stories are discussed in detail among the team members to estimate the plan for calculating the size and effort required for developing this application.

The story size estimation for each user stories are calculated based on three parameters story point, ILF and Complexity. The total story size for Students Login application is 132. Table V shows the sample user story and determines the story size.

TABLE V
Sample user story and story size Estimation

Story Number	Story Name	Story Point	ILF	Complexity	Story Size
1	As a student, I want to register five subjects for my fall semester through online.	5	2	1	10
2	As a student, I want to check my last semester’s grades through online.	3	2	2	12
3	As a student, I want to check my current semester timetable through online.	1	1	3	3
4	As a student, I want to check my current semester registered subject’s attendance details through online.	3	3	3	27
5	As a student, I want to download the study materials posted by the faculty through online.	5	4	4	80
Total story size					132

A. Output for normal agile process

Based on the estimated story size, the effort can be predicated by considering the number of story size (velocity) delivered at each end of the iteration. First, the initial velocity will be generated randomly. Table VI shows the initial iteration plan for the application. The random value generated for this application is 20. So the total iterations needed for this application is 7 iterations considering 20 story sizes will be delivered at each of the iteration. The iteration length should not exceed more than four weeks.

TABLE VI
Initial iteration plan for the sample application

Iteration number	1	2	3	4	5	6	7
Story Size Completed	20	20	20	20	20	20	12
Total story size completed = 132 and remaining story size= 0.							

Later, by considering the first iteration velocity, the remaining iterations are predicated and the procedure will be continued until the story size is equal to the value zero. In the process the first iteration velocity is 25. So the number of iterations will be decreased 6 iterations. Table VII represents the iteration plan other than initial plan of sample application.

In normal agile process the velocity prediction is a very challenging one because of velocity delivered at the end of the iterations will not be same and the previous iteration velocity value is given as input for the next iteration velocity prediction. So the velocity produced at the current iteration may increase or decrease. This will lead to a worst optimization.

TABLE VII
Result of Normal agile Process for sample application

Iteration number	IT 1	IT 2	IT 3	IT 4	IT 5	IT 6	IT 7	IT 8	IT 9	Total SS	Remaining SS
1	25	25	25	25	25	7	-	-	-	132	107
2	25	34	34	34	5	-	-	-	-	132	73
3	25	34	12	12	12	12	12	12	1	132	61
4	25	34	12	30	30	1	-	-	-	132	31
5	25	34	12	30	25	6	-	-	-	132	6
6	25	34	12	30	25	6	-	-	-	132	0

B. Output for Agile with HS Algorithm

The initial iteration plan of harmony agile process for sample application is represents in Table VIII.

TABLE VIII
Initial iteration plan for sample Application

Iteration Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Story Size Completed	10	10	10	10	10	10	10	10	10	10	10	10	10	2
Total story size completed = 132 and remaining story size= 0.														

In Harmony agile process, the velocity adjustment in iteration is done by calculating the average of previous two iterations velocity value. The result of the value is given to current iteration. This process will be done till the story size value becomes null. The result of the harmony agile process is represented in Table IX.

TABLE IX
Result of Harmony agile Process for sample application

Iteration number	IT 1	IT 2	IT 3	IT 4	IT 5	IT 6	IT 7	IT 8	IT 9	Total SS	Remaining SS
1	15	15	15	15	15	15	15	15	12	132	117
2	15	24	24	24	24	21	-	-	-	132	93
3	15	24	18	18	18	18	18	3	-	132	75
4	15	24	18	24	24	24	3	-	-	132	51
5	15	24	18	24	24	24	3	-	-	132	27
6	15	24	18	24	24	21	6	-	-	132	6
7	15	24	18	24	24	21	6	-	-	132	0

The result is generated for both normal agile process and harmony agile process, the graph is generated for each process. The Figure 4.2 shows the optimization is achieved in Harmony agile process where the velocity value is predicted effectively and it is stabilized after third iteration. In Figure 4.1 shows the normal agile process, till the end of the process the velocity is not optimized (more variation at each of the iteration).

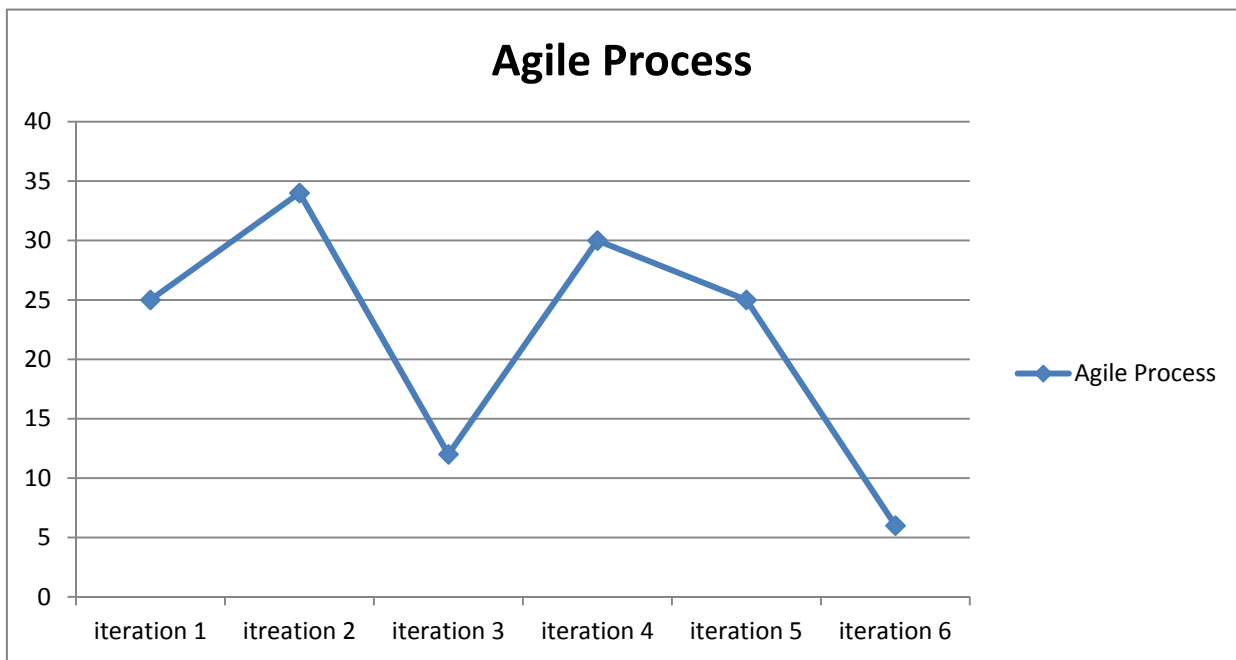


Fig. 1. Normal Agile Process

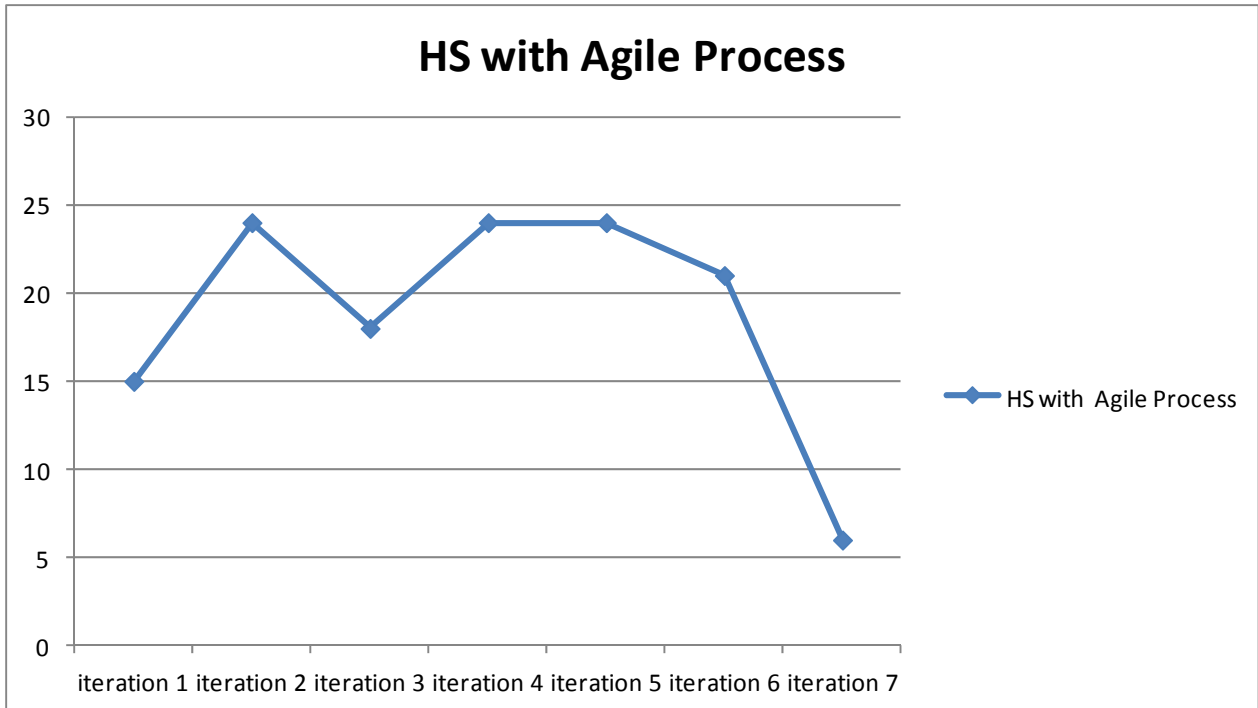


Fig. 2. Harmony Agile Process

V. Conclusion

The HS algorithm is well-defined and the principles are matches with agile process. The well-defined formulas are applied for finding the required parameters to attain good optimization in result. This method is analyzed by considering sample application and the result is proven enough to show after some iteration it is possible to attain an optimized value. This method is suitable and effective for large application and shows the accurate result in the final stage. Like HS algorithm all other algorithm has similar nature with agile process, so in future we can implement the agile estimation plan by using these algorithms. Similarly the same concept can be verified with other languages and other methodologies with its appropriate procedure.

REFERENCES

- [1] Greg Smith, Ahmed Sidky, "Becoming Agile: --in an Imperfect World", Manning Publications Company, 2009.
- [2] Mike chon, "Agile Estimation and planning", Prentice Hall of India, 2006.
- [3] P.V.G.D.P. Reddy, C.H.V.M.K. Hari," Fuzzy based pso for software effort Estimation", Communications in Computer and Information Science, 147 (2011), pp. 227–232
- [4] T. Gonsalves, A. Ito, R. Kanwabata and K. Itoh, "Swarm Intelligence in the Optimization of Software Development Project Schedule", Annual IEEE International Computer Software and Applications Conference, IEEE, 2008.
- [5] S.S. Dahiya, J.K. Chhabra, S. Kumar, "Application of Artificial Bee Colony Algorithm to Software Testing", 21st Australian Software Engineering Conference (ASWEC), April (2010), pp. 149–154
- [6] Frederick Ducatelle, Gianni A. Di Caro, Luca Maria Gambardella," Principles and applications of swarm intelligence for adaptive routing in telecommunications networks, Swarm Intelligence, September 2010, Volume 4, Issue 3, pp 173-198.
- [7] James Kennedy, James F. Kennedy, Russell C. Eberhart, "Swarm Intelligence", Morgan Kaufmann, 2001.
- [8] Binitha S, S Siva Sathya , "A Survey of Bio inspired Optimization Algorithms ", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012, PP 137-151.
- [9] Kang Seok Lee, Zong Woo Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice", Computer Methods in Applied Mechanics and Engineering , Volume 194, Issues 36–38, 23 September 2005, PP 3902–3933.
- [10] Mohammed A. Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar; Asaju La'aro Bolaji, "Nurse Scheduling Using Harmony Search", Sixth IEEE International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011, PP 58 – 63.
- [11] Yaghob Gholipour and Mohammad Mehdi Shahbazi , "Resource-Constrained Scheduling of Construction Projects Using the Harmony Search Algorithm", Journal of Industrial Engineering, University of Tehran, Special Issue, 2011, PP. 51-60.
- [12] Mohammed Azmi Al-Betar and Ahamad Tajudin Khader, "A harmony search algorithm for university course timetabling", Annals of Operations Research, April 2012, Volume 194, Issue 1, pp 3-31.