INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 3 ISSUE 1 –JANUARY 2015 - ISSN: 2349 - 9303

A Greedy Heuristic Approach for Sprint Planning in Agile Software Development

S.Jansi[1]
[1]Sona College of Technology,Department of CSE
sumithaselvaraj00@gmail.com

Mrs.K.C.Rajeswari[2]
[2]Sona College of Technology, Department of CSE
rajeswari.chinnasamy@gmail.com

Abstract—The sprint planning phase in agile software development has a major impact on the project success. The optimality of a sprint plan depends on several factors of the user stories. The factors are estimated complexity, risk values and business value of the user stories. So these factors must be included in each sprint. To achieve this, planning problem is first converted into a generalized problem. Then the problem is solved using the linear program in IBM ILOG CPLEX optimizer. The feasible value for the large sized problem takes time to compute and cannot be used for operational use. So, Greedy Heuristic approach is given as a proposal to integrate with the CPLEX optimizer to obtain the optimal solution. The computational result of Greedy heuristic is an effective and takes less time than the time taken by CPLEX optimizer.

Keywords—Agile methods, greedy heuristic, optimization model, scrum framework,sprint planning.

1 INTRODUCTION

Agile software development is mainly focused on the completion of customer’s requirement within in the time. For that it uses several methods such as Extreme programming (XP), scrum and lean software development. Here the consideration is about sprint planning which comes under scrum. Scrum is a framework for the large scale new product development. During the incremental and iterative design implementation in scrum, the software is described in terms of user functionalities (user stories) and at each iteration (sprint) the team should deliver the set of user stories that maximizes the utility of the users. Sprint planning is that within the estimated duration of the sprint, assigned tasks have to be completed. Durations can be limited by assigning the related tasks within the sprints. One user story should be delivered before another one is realized.

In sprint planning, user stories are estimated by the sprint team’s experience that they have faced before. Estimation plays a vital role in the sprint plan, as it decides the successful project’s sprint plan. Based on the estimation, user stories are prioritized for the sprint plan. It includes complexity, risk, and correlation among the user stories. Sprint planning effectiveness is mainly depends on taking all the variables and constraints into the consideration. So that optimal solution will be obtained within less time.

Some tools are available to support agile project management. Scrum works [5] and Mingle [4], provides a set of parameters to deal with user story risk, complexity and utility. But these tools lack in providing the correctoptimization solution for the operational use.

2 LITERATURE SURVEY

Agile software development practices accomplish improved software quality and increased development team productivity. Flexibility was acquired by systematically responding to customer requirements and changes. Most important features of agile are (1) achievement of customer satisfaction through continuous delivery of software (2) co-operative working of product owners and scrum team, and (3) by the face-to-face communication in the team [2].

Agile methodology is specifically developed to facilitate communication and coordination within a dynamic team environment [11]. This includes practices such as shared team rooms to encourage recurrent informal communication. Then the informative workplace environment provides ubiquitous information dissemination and spontaneous feedbacks throughout the development of the product. So that customer representative can have close partnership with the product development by providing the feedback. Rather than adhering to traditional methods of requirements gathering and design before software production, agile teams deal with the complexity of software development by practicing rapid iterative development from project inception.

Agile software development method includes: Dynamic Systems Development Methods (DSDM) [2], Feature Driven Design [3], Crystal [4], Scrum [5], Extreme Programming (XP) [6] and more recently Lean Software Development [7].

InterMod methodology [1] was proposed, its aim is to help the development of high quality software with accuracy. This methodology allows gathering and validation of the requirement as an incremental process. Approach concentrates on demonstrator diagram through which flexibility on planning and facing unexpected changes was analyzed in earlier stage. Then the prior validation of the model allows to progress and guide subsequent activities of the user functionalities. It improves the lack of consistency by means of continuous integration of models in the agile process.

Time-boxing is a common practice used to assist simple design and scheduling. User functionalities Development are split into separate time periods, with a set number of hours allowed for each task. For each time period, deadlines and resources are fixed, while deliverables are more flexible. Thus the scope of the development effort is adjusted to meet scheduling constraints. The functionalities which are not fit into the current time period are dropped or reconsidered based on the customer satisfaction. At the end of the time-box each task should be completed [12].

HDP (Hybrid Dynamic Programming) heuristics is to attain processing time similar to the one of dynamic programming algorithm applied to a classical Knapsack Problem (KP) to have a good performance in terms of gap. Cooperation of BB (Branch and Bound) algorithm with HDP is to obtain an exact solution. This combination gives processing time similar to the one of a classical branch and bound method. Though the cooperative method improves the optimal value, solution is an alternative to limit the processing time [13].

In a previous work [8], the sprint planning problem for agile projects proposed an optimization model that is also based on the team estimates and a set of development constraints. This model produces the optimal sprint plan by maximizing the business value perceived by the user. ILOG CPLEX optimizer (IBM, 2011) was used with optimization model to acquire the feasible solution for medium-sized problem. But it is a time consuming process for large-sized operational use problem.

The sprint planning problem for agile Data Warehouse design was formalized and proposed a multi-knapsack model to solve it. Model was tested on both the synthetic and real projects, and then found that the exact solution is determined for the medium sized problem in a time that is fully compatible with the development process. But for the large sized problems, a heuristic solution that is just a few percentage points far from the exact one can be returned in a couple of seconds. To present the plan in a more effective way, optimization module was coupled with the existing software's for agile project management.

Here, Greedy heuristic approach is proposed that will provide less computing time for large-sized sprint planning problem. Optimal solution is also acquired through this approach. Therefore, approach is integrated to the optimization model that could be used interactively and coupled with existing software for the agile project management.

2.2Outline

The paper is organized as follows section 2 summarizes the agile practices with the goal of sprint planning problem. Section 3 summarizes the mathematical formulation with notation and section 4 ends up with conclusion.

3 Proposed System

Sprint planning phase is within an iterative and incremental approach which ensures the criticality of project success. New requirements may arise during the project and the plan should be flexible enough to accommodate them. In sprint planning, tasks are estimated and completed according to their schedule. Track of the project is recognized through these assessments in sprint planning. Scrum lifecycle embraces the sprint planning phase to acquire the success of the project.

In fig1, initially user stories are the user functionalities which are chunked into tasks. Individual tasks are estimated by the expert's past experience and knowledge. Utility and complexity factors are estimated by the story points of the user story. Planning poker method is used to estimate the complexity of user stories by assigning the story points. Then the user stories are prioritized based on their estimated values. Critical risk and uncertainty risk are the two typical risks on the user stories.

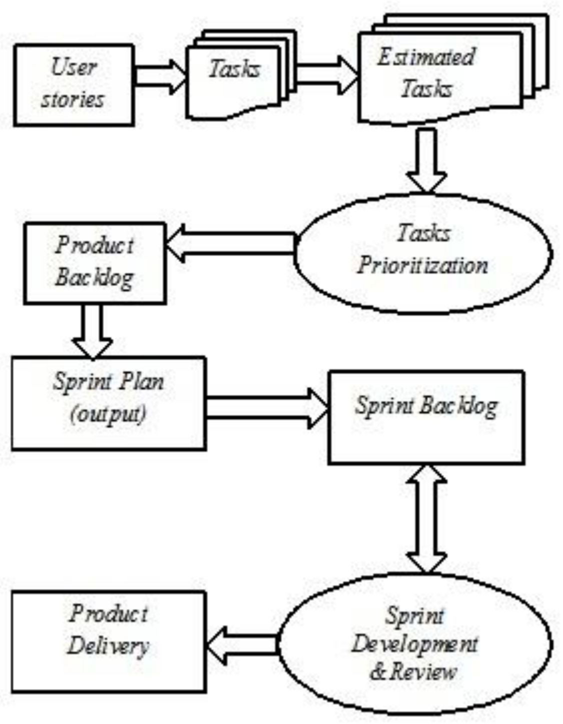

Fig.1Scrum Lifecycle

Mainly, critical risk has a main impact on the user stories. For example, if the user story 1 has a risk and most probably it affect the story 2. The uncertainty risk is about the unexpected problem occurring during the project progress. Both types of risks are estimated within the following values, risk 1(no risk found), 1.3(low risk found), 1.7(medium risk found) and 2(high risk found). The user stories are correlated by the means of OR and AND dependency.

Based on the prioritization, the product backlog is composed and then partitioned into sprints. User stories are allocated to the sprints through the estimation of complexity, development velocity and affinity of the user story. Sprint backlog carries out the details of the user stories from the product backlog. Sprint backlog have the information about the user stories which are in progress and not yet started. It will also have status of unsatisfied user stories which are not delivered. If the user stories are satisfied, then it will be delivered. Here the planning problem is converted into a generalized problem and solved using linear programming model by accepting the constraints in it. Objective function is to maximize the cumulative utility of the project.

Sprint planning problem has a major impact on the project success. To achieve the project success, following goals have to be satisfied in sprint planning:

1. **Customer satisfaction** – It is achieved by,(i) Delivering the user stories with high utility based on the customer expectations.(ii) To maximize the user value, include the affine stories in the same sprint.
2. **Risk Management** –It is achieved by advancing a critical and unpredictable user stories to avoid a late side effects. Then by distributing the risky stories uniformly.

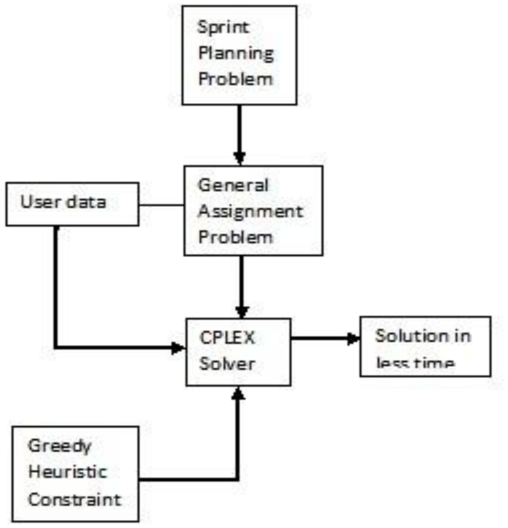

Fig.2 System Architecture

In fig.2, Sprint planning problem is assigned to be a large scale problem which means allocation of user stories in sprint plan should be higher. Then the sprint planning problem is framed into a general assignment problem. The framed problem should be understandable to CPLEX optimizer. In user data, number of user stories and sprint availability details have to be given as input. The MILP model constraints and greedy heuristic constraints are given as the enhancement to the CPLEX optimizer. The solution obtained is with less time when compared to the execution before enhancement.

4 MATHEMATICAL FORMULATION

In this section, greedy heuristic approach is proposed to minimize the time taken for computing the large-sized sprint planning problem in the IBM ILOG CPLEX optimizer. Let U = {1…n} be the set of n user stories to be allotted to the sprints. Here the procedure starts by optimizing sprint i=1 and then optimizes the remaining sprints in them, one at a time. Therefore at each iteration, greedy heuristic procedure consider a sprint i∈S and assigns to the stories that maximize the utility by solving the sub-problems.

Given a set of m sprint S and a set of n user stories U. let:

- w_{ij} =1, iff story j is included in sprint i, otherwise 0.
- U_j , be the utility of the story j.
- q_j , be the number of story points of story j.
- Q_i, be the capacity of sprint i, measured in story points.
- R_{CRj} , be the criticality risk of story j.
- R_{UNj} , be the uncertainty risk of story j.
- C_j , be the affinity between the story in j.
- X_j , be the set of similar stories to the story j.
- x_{ij} , be an accessory variable related to the number of stories in X_j included in sprint i.

To maximize the utility by solving the sub-problems by the following mixed integer linear programming model:

$$(S_i) Z_{S_i} = max \sum_{j=1}^{n} (m-i+1)\, U_j \, (R_{CRj}\, w_{ij} + C_j\, x_{ij}) \quad (1)$$

$$\text{s.t } \sum_{j=1}^{n} q_j\, R_{UNj}\, w_{ij} \leq Q_i, \text{ i} \in S \quad (2)$$

$$x_{ij} \leq \sum_{K \in X_j}^{n} w_{ik}\, , \quad \text{j} \in \text{U} \quad (3)$$

$$w_{ij} \in \{0, 1\}, \quad \text{j} \in \text{U} \setminus B_i \quad (4)$$

$$w_{ij} = 0, \quad \text{j} \in B_i \quad (5)$$

$$x_{ij} \geq 0, \quad \text{i} \in \text{S, j} \in \text{U} \quad (6)$$

Where B_i represent the stories already assigned to the sprints considered in the previous iteration.

The mainly objective function is to (1) maximize the cumulative utility of the sub-problems. The utility U_j for each story is increased by increment of criticality riskR_{CRj}. So that critical stories are considered in the earlier stage of the project and then by increment of affinity C_j of the story, similar stories are included in the same sprint. Constraint (2) ensures the overall complexity of all the stories assigned to each sprint does not exceed the estimated sprint capacity value. To strengthen constraint (2), sprint capacity or weight of the user stories are modified to reduce the computing time. Constraint (3) ensures correct evaluation of x_{ij} which does not require any integrality constraint. This mathematical formulation can be solved by a MIP solver.

Sub-problem should follow the forthcoming strategies:

(i) If there is no user story j in sprint I, then fix w_{ij} =0 and re-optimize the knapsack problem(S_i).

(ii) To increase the profit in every knapsack problem S_i by multiplying the profit with K_j (ie, coefficient K_j is defined for user story j. By having w_{ij}=0 in the current solution, increase the coefficient(K_j). Then restart the greedy heuristic from the first sprint i=1. This strategy requires too much iteration to reach a feasible solution.

Then the unchanged user stories should be distributed among the sprints:

$$\sum_{(i,j)\in w=}^{n} w_{ij} \qquad \geq L' \qquad (7)$$

Where w represent the subset of variables$\{w_{ij}\}$set to one in the previous sprint planning and L is the minimum number of variables of subset w that do not have to change. Then L' is the difference between L and the story that already got changed in the previous sprints.

4 CONCLUSION

In this paper, an approach for the sprint planning in agile methods based on an integer linear programming model is provided. The mathematical formulations are solved by a general purpose MIP solver, IBM ILOG CPLEX optimizer. However, the computing time to solve to optimality of the medium sized problem is very large. In an attempt to reduce the computing time, reduction and sub-problem are solved. Sprint planning problem is framed by gathering the risk, complexity, utility and affinity values of the user stories. Constraints and these factors values are given as input to the sprint plan. Optimal solution for medium-sized instances is solved easily within less time. But for the large instance, the problem prevails in solving the sprint plan. Since solving the model to optimality in MIP solver, is not suitable for an operational use and takes some time to solve this problem. So that Greedy heuristic approach is given as a proposal. In this approach, planning problem are chunked into sub-problem and solved separately in iterations. From the iteration of sub-problem, best feasible solution is retrieved at each step. The computational result using greedy heuristic approach will be able to attain a feasible and optimal solution with very quick time constraints. This approach can be improved by identifying the risk and designing a plan in a real time approach. Then by identifying the skill of each expert, sprint plan can be designed to obtain a further better feasible value as a result.

REFERENCES

[1] BegonaLosada, MaiteUrretavizcaya, Isabel Fernandez- Castro, A guide to agile development of interactive software with a ‘‘User Objectives’’, 78 (2013) 2268–2281.

[2]J. M. Bass, ‘Agile Method Tailoring in Distributed Enterprises: Product Owner Teams’, in Proc. IEEE 8th Int. Conf. on Global Software Engineering, Bari, Italy, 2013, pp. 154–63.

[3] P.Coad, E. LeFebvre, and J.D. Luca, Java Modeling in Color. Englewood Cliffs, NJ: Prentice Hall, 1999.

[4] A.Cockburn, Agile Software Development. Reading, MA: Addison Wesley, 2001.

[5] K.Schwaber and M. Beedle, Agile Software Development with Scrum, 2001.

[6] K. Beck and C. Andres, Extreme Programming Explained, 2nd ed. Addison Wesley, 2004.

[7] M.Poppendieck and T.Poppendieck, Lean Software Development: An Agile Toolkit. Addison-Wesley Longman Publishing Co., Inc., 2003.

[8] Matteo Golfarelli, Stefano Rizzi, and Elisa Turricchia, ”Sprint Planning Optimization in Agile Data Warehouse Design“, 40136 Bologna, Italy.

[9] Kolisch. R. Padman. R, An integrated survey of deterministic project scheduling. Omega 29, 249–272, 2001.

[10] Herroelen, W., Leus, R., Demeul -emeester, E., “Critical chain project scheduling: do not over-simplify”, Project Management Journal 33, 48–60., 2002.

[11] Stavros Stavru, “A critical examination of recent industrial surveys on agile method usage” P.B. 48, 1164 Sofia,

[12] Golfarelli M, Rizzi S, Turricchia E. Multi-sprint planning and smooth re-planning: an optimization model. Journal of Systems and Software 2013; 86(9):2357–70.

[13]. V. Boyer, Didier EL BAZ, MoussaElkihel LAAS-CNRS, “Solution of Multi-dimensional Knapsack Problem via Cooperation Of Dynamic Programming And Branch And Bound”, 31077.

[14]. Boschetti M, Hadjinconstantinou E, Mingozzi A,” New upper bounds for the finite two-dimensional orthogonal non-guillotine cutting stock problem”,2002;13:95–119.

[15]. Boschetti M, Maniezzo V, Roffilli M, BolufeRohlerA, “Matheuristics :optimization, simulation and control, vol. 5818. Springer; 2009.p.171–7.

[16].Boschetti M, Mingozzi A. The two-dimensional finite bin packing problem, 2003; 1:27–42.

[17]. Boschetti M A, Montaletti L. An exact algorithm for the two- dimensional strip- packing problem.OperationsResearch2010; 58 (November (6)):1774–91.