Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, v. 6, n. 1, 2018.

Trabalho apresentado no XXXVII CNMAC, S.J. dos Campos - SP, 2017.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Optimization model for assigning tasks in Scrum Agile Development (work in progress)

Ellen L. Mendez[1]
Diego P. Pinto-Roa[2]
Universidad Nacional de Asunción, Facultad Politécnica, San Lorenzo, Paraguay

Currently software development companies are developed in a context where changes are a constant in all projects, and the response to these changes must be made with the least time and impact. Because of this the tendency is to use agile methodologies. Scrum is an agile software development methodology that meets the needs of responses to change. Scrum defines an incremental process, so, it allows to constantly deliver to the client software requirements in short time lapses [1].

The development of tasks is divided into work cycles called Sprint, where each Sprint can last two to four weeks. Before starting a Sprint, a planning meeting is held where the customer, also called Product Owner, prioritizes the tasks and the development team selects the tasks to be developed within the Sprint. This selection of tasks is called Sprint Backlog, see Figure 1.

Figure 1: Scrum overview

Defining the tasks of a Sprint and the assignment of tasks are activities that are done through team experience and based on the technique of Expert Judgment that are empirical. On the other hand, the mathematical optimization provides several tools to solve the problem of assigning tasks to resources, this is known task schedulling [2]. This paper proposes to apply the exact techniques of scheduling to the task assignment problem in agile software development based on the Scrum model.

The problem of task planning or task scheduling presents its beginnings in industrial planning [2]. This type of problem addressed to the problem in question can be defined

[1]emendez@pol.una.py
[2]dpinto@pol.una.py

© 2018 SBMAC

as follows: Given M resources (developers) and N tasks (codes to be developed) with T_{ij} time duration units of each task i executed in resource j-thest. It is desired to program the N tasks in the M resources, searching for the most appropriate execution order, fulfilling certain conditions of precedence between the tasks defined in P_{ik}. As defined below, the input parameters to be used are presented:

$i = 1, 2, ...N$, Tasks corresponding to Product Owner requirements.

$j = 1, 2, ...M$, Resources or developers with different capabilities. In this work we consider junior, middle and senior developers.

$P_{i,i'}$, Matrix of precedence between tasks. If $P_{i,i'} = 1$ indicates that task i must be completed before starting task i', otherwise $P_{i,i'} = 0$.

T_{ij}, The integer matrix of task execution time i in a resource j. This table can be formed considering the three types of developers and time based to complete a task.

U, Constant value large enough, which can be considered $U = \sum_{i,j} T_{ij}$.

The variables of problem are: $X_{i,j}$, If $X_{i,j} = 1$ the task i is assigned to the resource j, otherwise $X_{ij} = 0$. $Y_{i,i'}$, If $Y_{i,i'} = 1$ indicates that the task i is executed before the task i', otherwise $Y_{i,i'} = 0$. a_i, Start time of task i Z, Total time from start to finish all tasks (makespan)

The model ILP is described in the following equations:

$$\text{Min } Z \tag{1}$$

$$s.a.\ Z \geq \sum_i T_{i,j} \cdot X_{i,j}, \qquad \forall j \tag{2}$$

$$Z \geq \sum_i X_{i,j} = 1, \qquad \forall i \tag{3}$$

$$a_{i'} \geq (a_i + T_{i,j} \cdot X_{i,j}) \cdot P_{i,i'}, \qquad \forall k \neq i, \forall j \tag{4}$$

$$Y_{ki} + Y_{ik} = 1, \qquad \forall i \neq i' \tag{5}$$

$$a_{i'} \geq (a_i + T_{i,j} \cdot X_{i,j}) - U \cdot (3 - Y_{i,i'} - X_{i,j} - X_{i',j}), \qquad \forall i \neq i', \forall j \tag{6}$$

The restriction 2 defines the maximum time among all developers. The constraint 3 indicates that each task can only be assigned to a developer. Inequality 4 guarantees the non-overlapping time between two tasks with precedence. The constraints 5 and 6 ensure non-overlapping time between tasks assigned to the same developer. With the problem formulation and the mathematical modeling we have formalized a task planning process for the software development process oriented towards agile methodologies (Scrum, for this case). Among the following activities are the validation of the model in real development cases and the corresponding tests in comparison to other heuristic models already available in the state-of-the-art in what corresponds to the Scrum methodology.

References

[1] K. Schwaber. Agile project management with Scrum. Microsoft press, 2004.

[2] Miller G., Galanter E. Plans and the Structure of Behavior, Editorial Holt, New York-USA (1960).

© 2018 SBMAC