Turkish Online Journal of Qualitative Inquiry (TOJQI)

Volume 12, Issue 5, May 2021:4719 - 4729

Research Article

A Preliminary Review on Resource Allocation within a Software Team

R. Mamatha [1], Dr K. G. Suma [2]

[1]Research Scholar, Department of CSE, KL University Hyderabad,

Assistant Professor, Department of CSE, G. Narayanamma Institute of Technology & Science.

[1]mamatha.racharla@klh.edu.in

[2]Associate Professor, Department of CSE, KL University Hyderabad.

[2]kgsuma@klh.edu.in

Abstract:

Currently, in the IT industry people factor has become more crucial at the time of determining the project quality. Resource allocation in modern software engineering is the art of recognizing the best available resources for the project, then assigning them to a team and monitoring their workload throughout the development process, and re-assigning resources if at all required. There exist no quick and ready-made solutions for the human resource allocation problem. Project managers however can estimate by using tested guide-lines that are based on experience and data. In software projects, one cannot even precisely answer how accurate is the allocation of a resource is because the only way to ascertain the accuracy of allocation is to compare it with the actual available resources. Hence the project manager's goal is to allocate the available resources to maximize employee productivity. There exist many commercial tools for allocating the task to the employee but no intelligent tool is available to automatically delegate the work among the team. This method will make use of existing project-related data to find the useful information of the work that is done by the team member to automatically allocate the new task.

Keywords: Resource allocation, Modern software development.

I. Introduction

Generally speaking, any Software project has two main activity dimensions: Engineering and project management.

The first dimension will deal with building the system and focuses on issues such as how to design, test, code, etc. The project management dimension that deals with properly controlling and planning the engineering activities to meet project goals. To perform various tasks a well-defined process is required so that the output becomes more dependent on the capability of the process. The project manager has to follow the process because:

- A Process represents collective knowledge. Using which will increase the chance of success.
- A Process may have extra steps, but always never know before-hand which ones are not required, and hence will increase the risk by taking shortcuts.
- Without process, we cannot predict much about the outcome of the project.
- Organizations cannot learn effectively without having a defined process.

Principles of resource allocation:

Compartmentalization: The project should be disintegrated into manageable tasks and activities.

• Time allocation: Each task should be allocated some number of time units, also possibly a start and a completion date.

• Interdependence: The relationships between the tasks have to be established because some activities will depend on others, when other activities may occur independently.

• Effort validation: Each project will have a desired number of staff. The project manager should ensure that at any given time no more than the allocated number of people have been scheduled

• Defined responsibilities: Each task that is scheduled must be assigned to a specific team member.

• Defined outcomes: Each task that is scheduled must have a definite result.

• Defined milestones: Group of tasks must be associated with milestones.

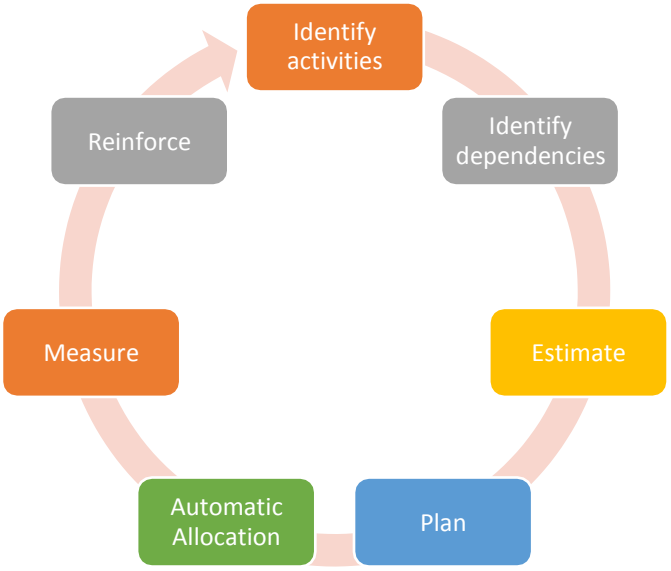


Fig:1.1 Resource Allocation Process

- Identify Activities: Identifying the activities is a process which must be performed to produce the various project deliverable.
- Identify Activity Dependencies: Documenting and identifying interactivity dependencies.
- Estimate Resources: Estimating the resources and calculating the number of periods that will be needed to be complete an individual activity.
- Create a Project plan: To analyze the resource requirements, activity sequences, and activity duration by creating the project plan.
- Allocate People to Activities: The people are allocated to activities based on various activities.
- Measure: Measuring the outcome in terms of productivity.
- Reinforce: If additional resources are required for completing the task, the allocation will be done according to change requirements it is a continuous cyclic process.

II. Literature Review

A lot of work is done on how resource allocation can be done; this paper summarises how resource allocation can be done in different methods.

Otero, L. D.,[1] has given a Best-Fitted Resource (BFT) methodology demonstrated using a simple human resource allocation consisting of six software engineers and a single task. This method contains four main steps and at each step, a table is constructed:

Step1: Task Required Skills (TRS)

The primary goal of the TRS table [1]"is to define the levels of skills that are required for a given task. Each skill level will be specified in terms of its expected use (e_{jt}) and the complexity (c_{jt}). both e_{jt} and c_{jt} are defined subjectively by a decision-maker with discrete values ranging from 0 to 1" [1].

For e_{jt}, the default values for Little use are 0.3, for Significant use is 0.7 and for Extensive use is 1.0

For c_{jt}, default values for a Simple task are 0.2, for Complex task is 0.5 and for a Very challenging task is 1.0

Step 2: Skill Relationships (SR)

The main goal of the SR Table is to show the learning curve relationships (r_{jk}) between known and unknown skills. Managers who are the decision-makers subjectively define learning curve relationships with discrete values ranging from 0 to 1. Default values for r_{jk} values are as follows: No relationship is 0, for a Weak relation it is 0.2, for Intermediate type relation it is 0.5 and for Strong relation, it is 1.0.

Step 3: Resources' Skill Set (RSS)

The main objective of this table is to prepare a tabular representation of knowledge for available resources with discrete values.

Step 4: Best-Fitted Resource (BFR)

To determine available resources with the skills required for a task. The most suitable resource will mostly take the least amount of training time.

It is analyzed that only 1 main skill is considered to assess the suitability of the candidate with software task. The BFR methodology takes into consideration the skill-relationship tables to describe how previous knowledge on various related skills have contributed to learning required skills.

Carlos E. Otero et.al [2] have proposed a "multi-criteria decision-making approach which allocates the resources to the software engineering task assignment. The approach uses a unique desirability function (DF) which provides a unified metric representation of the suitability between the complete set of skills that are available among candidates and skills required to perform a task" [2].

"All the experienced project leads should identify the particular skill set that is required to do a particular project. Then, from the available pool of candidates, each candidate is scored on each skill based on previous performance, educational background, or a combination of both. In the proposed approach, they suggest a 0 – 10 scale, 0 being the least desirable and 10 being most desirable. Once the ranking of each candidate concerning all required skills is done, the scores are used as input to desirability functions to compute a unified score. Finally, the candidates with higher desirability scores will have the chance of getting selected for the particular project. The resource allocation process in software engineering can be optimized by finding the skill set that provides the optimal candidate for the particular task. To formulated in this way, desirability functions seem to provide a unified metric that characterizes the fitness of engineers based on a set of predefined skills. Once the desirability function of all available engineers is computed, decision-makers may use this information by simply choosing the most desirable engineer to allocate the resource" [2].

Computing desirability function: selection of candidates for some particular tasks with availability factor and candidate vector is identified then each candidate can be evaluated against their set of required project skill set on a scale of 0-10, where 0 represents the lowest score and 10 the highest. Finally, to assess the rank of each required skill set, a weight vector W is created where r_i represents the importance of the s_i skill using the previously identified 0 – 10 scale. Once the information of X, S, and W is collected, desirability values will be computed using the desirability matrix.

"After the individual desirability values(D) of each skill is computed, then the overall candidate's D value may be computed. As seen, each overall value of D is computed as the geometric mean of all m individual desirability of candidates 1, 2, …, n" [1]. and can decide on the best fit candidate for required skill" [2].

Junchao Xiao, Qing Wang et.al, [3]has given a "constraint-driven human resource scheduling method "[3] where it considers few parameters such as precedent activity which is nothing but the activity that must be completed before the currently doing activity, skill of the employee, the scope of the task that has to be allocated, budget constraint .considering all these parameters resource allocation will be done. These are all divided into required constraints and optional constraints.

(1) Required constraints like Activity type constraint, Skill constraint, Workload constraint, and Budget constraint (2) optional constraints like Schedule-preferential constraints, cost-preferential constraint, and same product development experience constraint. These optional constraints are used to satisfy some optimization goals. With such models a "constraint-driven optimal human resource scheduling method"[3] is realized as follows: Firstly a Directed Activity Graph(DAG) is constructed then a Current Schedulable Activity Queue(CSAQ) is constructed and when the length of CSAQ is zero then the scheduling will end with success if the length is not zero then check the activities in the queue to see whether schedulable workload and time of the capable human resources can satisfy all the required constraints. If certain activity cannot satisfy all required constraints then no sufficient resource can be scheduled for it. If all activities in the current CSAQ have been checked it means that in CSAQ no activity can be scheduled for execution under current resource condition. If certain activity satisfies required constraints then the state of such activity is set to "checked" and the information such as CSAQ, workload, Scheduling time, and scheduling state of human resource will be pushed to stack as a state. Human resources are scheduled for the activity constraints and the appointed optional constraints can be satisfied, and the workload of the scheduled human resources and state of schedulable time should be updated. In a search tree, this activity is added as the sub-node of the previous successful scheduled node. If the backtrack is empty it means that there is no selectable way to schedule which indicates that the scheduling ends with failure. The non-scheduled activities can tell what kind of human resources are scarce. Otherwise, popup a state from State Stack to update the current CSQA, all the human resources, and activities.

Mingshu Li, Juan Li et. al,[4] has analyzed the effect of staffing pattern on software projects. It has been observed that cost and staffing levels are low at the beginning, and in the intermediate phase, it becomes peak and drops rapidly as projects draw to the conclusion phase. As the resources demand varies across the project's lifecycle, it is desirable to allocate resources to the project phase-by-phase as needed.
A special project is formed to integrate the functional areas, facilitate communication and avoid misunderstanding. This work is taking an initial attempt to examine the relationship between project performance and soft factors. This work is taking an initial attempt to examine the association between project performance and software factors. This paper has observed some staffing patterns as follows:

1. Rapid-team-build-up pattern: Here staffing levels peak in the requirement phase and decrease in later phases.
2. Fix-Staff pattern: Here team is fixed across the project lifecycle. Due to sufficient communication and learning time among the team members, it is assumed that the quality is increased.
3. Design-Construction-centric pattern: staffing levels are high in the design and construction phase and low in other phases. The software productivity and quality are low in this case.
4. Implementation-centric pattern: Staffing levels are high in the construction phase. The software productivity is low in this kind of pattern.
5. Test-Centric pattern: Here the staffing levels are low in the early development phases, but increase in the testing and transition phase. In this pattern, the software quality may be high as the staffing level is stable in early phases but the productivity is hard to determine.
6. Classical-Rayleigh pattern: Here staffing levels are low at the initial phases gradually increase, peak at the construction phase and drop at the later phases. The software quality is high and the productivity is high.
7. Minimum-Design pattern: Here the staffing level is high during requirement and construction phases and drops during the design phase due to design reuse. In this type of pattern, the quality and productivity will be low.

"A box-plot of post-release defect density by staff patterns, it is observed that the defect density is low in Fix and Test patterns and high in Rayleigh and Rapid patterns. The Staffing pattern should be considered when allocating the resources to the projects. Test pattern is good for low defect density. Fix pattern is not bad as its defect density is low. The design pattern has high productivity and medium defect density. Rayleigh pattern produces high defect density and high productivity. Rapid pattern seems bad for both low quality and productivity" [4].

Nazia Bibi et.al, [5] have proposed "resource allocation optimization using Search-Based Software Engineering(SBSE). This paper addressed the Resource allocation(RA) problem such that it achieves objective functions that include minimum project duration and project cost and maximum resource utilization" [5]. Input variables that are considered are:
Time: Time spent on each activity can be calculated using three-point estimation formula:

$$T_e=(T_o+4T_m+T_p)/6$$

Cost: Three-point estimation formula is used to calculate the cost of each activity is given as:

$$C_e=(C_o+4C_m+C_p)/6$$

Human Skills: Human resources proposed certain skills to perform specific tasks. This can be calculated by the formula:

Skillset=Skillset*EI Rating

Output variables: optimized resource allocation plan will be the output result.

"3 SBSE algorithms namely MOGA, MOPSO, ENSES algorithms will be implemented for optimal RA, and results obtained from these algorithms are analyzed using Histogram and Pareto Front methods respectively. SBSE algorithms are effectively used to solve RA problems" [5].

Waqar Aslam et. al,[6] have proposed a "Quantitative Task allocation framework in a Distributed Agile Software Development(DASD) model "[6]. This model will assign the roles to their employees based on the employee experience and capabilities. "for developing any given project, consider if there are 'm' different roles with a possibility of each role having different requirements a unique requirement of all roles is collected. A Boolean matrix of all roles is created, in the matrix 0 indicates absence of requirements and 1 indicates presence of requirements" [6]. Next an arbitrary role is taken and evaluate the suitable team member, the member with the highest score is selected to fulfill that role. In case of any two team members get the same score, one with the lowest index value is selected. The roles which are tagged are subtracted from the next evaluation. This process will be repeated until all roles are assigned. Now consider if there are 't' team members and m roles to play. If t ≠ m this situation is termed as Unbalanced Load and it indicates that either there are too few team members or too many team members as compared to roles. If t = m then the situation is termed as Balanced Load. Ideally t set of tasks are essential to be generated so that team members can assume one role each. Multiple roles of the same nature can result, but unique identification of all roles can overcome this issue. Based on decision preferences due to time constraints on project completion duration, more team members can be hired on an Ad hoc basis.

The decision of allocating tasks to team members is done by aiming for a "quantitative method" [6] which considers capability requirement per role that is needed to execute a project. There is a selection for each role based on employee experience, past appraisal for undertaking similar tasks.

Mehdi Farhangian et. al,[7] have proposed an "Agent-Based Modelling of Resource Allocation in software projects based on personality and skill. In this paper, M. Farhangian and others proposed a computational model for measuring the performance of the team by skill competency and personality" [7]. They designed various scenarios with different degrees of dynamic level. It is studied whether resource allocation leads to performance advantages with respective dynamic tasks. "It has taken 2 team formation mechanisms in software projects into consideration, the first mechanism is Myers-Briggs type indicator (MBTI) which is based on human personality, a scale is established between 0 to 100 for each of four personality dimension of the employees like introverted/Extroverted, Intuitive/Sensing, Thinking/Feeling, Perceiving/Judging using these parameters final score for matching personality is constructed and the second mechanism is Belbin Team Roles (BTRs) which is based on roles of the individuals in a team, few constraints based on Belbin's findings are framed such that at least one Plant is essential in a team which has high creativity requirement, at least one Completer is essential in a team with a high urgency requirement, at least one Evaluator is essential in a team with a high complexity requirement, at least one Resource Investigator is essential in a team with a high complexity" [7]. Based on the above rules and constraints an Agent-based model is developed for assigning the task to maximize the utility of the system. Further, it is examined whether if personality distribution affects 2 different task allocation methods. They proposed a multi-agent tool that can be used by managers and researchers to investigate the effect of their employees and resource allocation strategies in a real environment. They provided a comprehensive model for the team managers to investigate the impact and effectiveness of different task allocation strategies in various dynamic environments and employees with different attributes in terms of skill and personality.

Thiago Jorge et. al, [8] have proposed "Human Resource Allocation automation in a Software Process that uses Cluster Analysis " [8]. It proposed a way in which one can explore the opportunities how best to allocate human resources from their historical information available. Factors related to human resource allocation process are taken from software project managers like (I) Individual Factors: personal interest, attitude, confidence, experience, knowledge, role to performance, (ii) Project Criteria: project type, cost, criticality, risk, complexity. For that collected data Mining techniques are applied to analyze the data.

The model uses "Case-Based reasoning" [8] to store, handle and recover historical allocation from organizational repository and feeds as input to a clustering algorithm, and this clustering algorithm will determine the likenesses between current allocation and historical allocations. Factors that are related to product accomplishment and project performance should be used for the evaluation of each allocation. The model has to learn from an action

performed by the person responsible for allocation and from team evolution. Suppose when a manager selects some random person to a given task who was not a similar case then all that new knowledge must be stored and fed to the model instantly.

Shampa Chakraverty et. al,[9] This paper proposed a model for reasoning about the groups called "Formal Concept Analysis (FCA) and a population-based Genetic Algorithm(GA).FCA is used to formally analyses realms of the team and task in terms of various skills and multi-objective GA is used to optimize the mapping of these two types of concepts given reliability, maximize skill utilization and ensure day and night continuity in a Collaborative Software Development (CSD)" [9].

The CSD framework consists of 2 processes (1) Generate Lattice: which uses FCA to generate two concept lattices one for available execution team and skills that the team should possess and the other concept which gives the relation between software development task and skills needed by them. (2) Process Allocation task: This uses GA to generate an optimized set of mapping between Task and Team concepts that optimizes certain pre-decided goals for CSD. Software development task s must be executed by different teams that are geographically discrete to achieve common goals. Any team in a particular region has unique skills that depend on several factors like work experience, training, socio-cultural background, and education qualification of team members. Next a Task concept lattice which is a conceptual hierarchy of all concepts Task-Skill context. Team context lattice is a graphical representation that shows the common skills of the team, as now a team and task concepts are known now the next step is to map these two concepts for achieving the goal. GA-driven optimization of Team concept and Task concept allocation is done: Here it generates an initial population with random but feasible encodings of a fixed number of chromosomes Nc is generated. Then Fitness of the generated chromosome is determined by three parameters Skill Utilization (SU), Continuity Factor (CF), and Fault Tolerance (FT). Next selection operators like Generation gap, Elitism are applied. Then Genetic operators like Single-Point Mutation and crossover are applied to the population. The GA process ensures that all the Team concepts and Task concepts are paired at least once even after mutation and crossover if at all the constraint is violated chromosome pair is taken to rectify it . GA-driven optimization will give optimal solutions.

Zhu Qing et. al, [10] present a practical approach for "Human resource configuration strategy which is based on life cycle model. According to this model each person is distinguished according to their ability, quantitative data is obtained to conduct allocation. This model goes with the following assumptions and Modelling constraints" [10]:

Assumptions:

a) It is expected that people with the same character capacity coefficient in any organization.
b) The workload of each employee character is relative to the capacity coefficient.
c) The total human resource value of a specific character should not be less than the total organizational demand.
d) In practice, every character has a different capacity coefficient in different organizations shows different products in the same organization.

Modeling constraints.

a) The total human input for certain characters must not exceed their human resource hold value.
b) The total effective workload for all characters in any organization should not be less than the required organization value.
c) Let the character do its job with maximum coefficient of efficiency which is termed as productivity first principle.
d) Managers need to assign less work such that they can engage themselves in management works which is termed as Management role first principle.

Hui Yi Chiang et. al, [10] have proposed a "framework for human resource allocation based on Integer Programming which takes consideration into both the efficiency and cost of human allocation. Suppose if *m* staff members are developing a single project that is to be exercised during time period and needs *n* skills to finish the task. Define f as number of staff with skill *s* from start time *a* to end date *b*. More than one staff could hold any kind of skill and he could be associated with several skills, and each staff can be assigned at most one skill. The communication efficiency of a staff member is related to overall project efficiency" [11].

Assume that the effectiveness is calculated by summarizing the efficiency of all skills overall staff members to maximize the total value, and the consecutive skill efficiency would be distributed if the staff member doesn't have maximizing efficiency of skill To demonstrate the above framework four variant were taken into consideration Developing project with cost variant is the elementary principle in the evaluation. The key factor

including the negative efficiency and communication are relevant to maximizing project efficiency, which is directly related to project quality and success.

Ansgar Lamersdorf et. al,[11] has proposed "An Empirically -based Task allocation in Global Software Development (GSD). As distributed software development can be done in several different criteria that possess different characteristics this work mainly focuses on offshore development within the organization. According to this paper, four methods are used to allocate the task allocation in global software development is used: Literature study, interview study, empirical case study, and modeling" [12].

a) Literature study: The result of the literature study can be used for identifying various causes that affect the distributed development and also used to identify the variables that are required to extend the distributed model toward the GSD.
b) Interview study: The experts in the interview will identify the process and product characteristics in the distributed development. By reconstructing the past allocation decision impact of global software development is studied.
c) Empirical study: It allows quantitative analysis of distributed software development. The study will be accompanied by qualitative interviews which can identify detailed characteristics and causes of measurement results.
d) Modeling: This is used for adapting existing models according to these results of empirical study and also for integrating the model into a decision support for task allocation. The main goal of model development is to reproduce the observed behavior and help the managers to assign the task that experts agree.

A comparison of these methods is given in the below table:

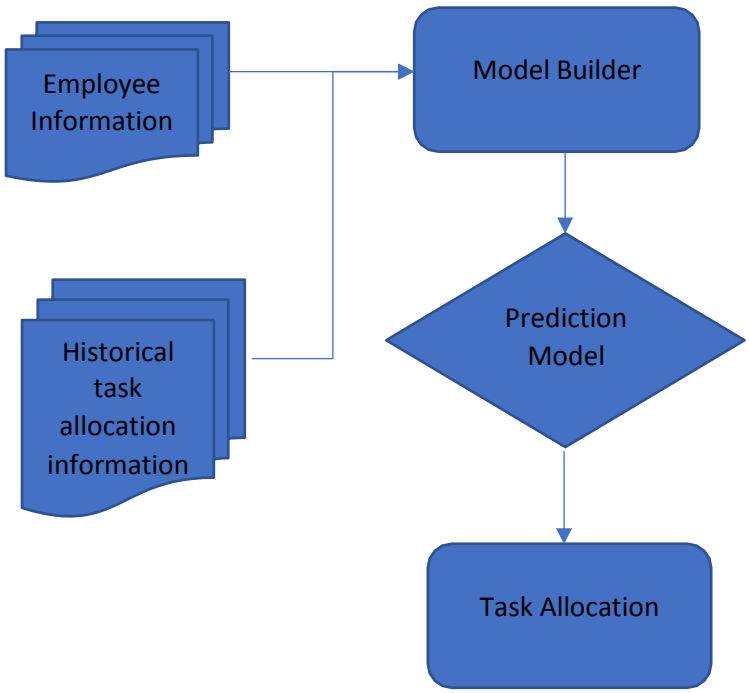
S.No.	Methodology used	Performance
1	Best-Fitted Resource (BFR) Here 4 tables are constructed namely: TRS-Task Require Skill table SR-Skill Relationship table RSS-Resource Skill Set table BFR-Best Fitted Resource table	It is analyzed that only one main skill is considered to assess the suitability of a candidate with a software task.
2	Life Cycle Model.	The capacity coefficient mainly depends on two factors. Suitability reveals the matching degree between each character and job While, proficiency mainly corresponds to the professional knowledge, work experience.
3	Desirability Function	"It uses DF function to provide a unified metric for representation of the suitability between the complete set of skills of the candidates and skills required for a given task" [2].
4	Cluster Analysis.	"Factors related to human resources allocation like (i) Individual Factors: personal interest, attitude, confidence, experience, knowledge, role to perform, (ii) Project Criteria: project type, cost, criticality, risk, complexity. Later DM techniques are applied to analyze the data" [8].

5	Genetic Algorithm.	"It maps skills and objectives types of concepts given reliability, maximize skill utilization and ensure day and night continuity in a Collaborative Software Development (CSD)" [9].
6	Search-based software engineering.	"This method addresses RA problem such that it achieves objective functions that includes minimum project duration and project cost and maximum resource utilization" [5].
7	Integer Programming.	"This method is based on Integer Programming (IP) which takes consideration into both the efficiency and cost of human allocation" [11].
8	Agent-Based Modelling of Resource Allocation.	Here a computational model for measuring the performance of the team by skill competency and personality is used.
9	Quantitative framework for Task allocation.	Here the roles are assigned to their employees based on their experience and capabilities. Consider if there are m roles to develop given project with a possibility of each role having different requirements" [6].
10	Data Mining Techniques. Team size is a software development effort predictor.	According to this paper, four methods are used to allocate the task allocation in global software development is used: Literature study, interview study, empirical case study, and modeling" [12].

III. Methodology

It is evidenced from the literature survey that ML was used only in engineering stage of project development. If it is used for management phase it saves lot of time of managers where they spend lot of time in routine task of monitoring and controlling the team. If this section of task gets automated by machine learning where the system uses the historical data and takes decision in allocating task for a team member then the managers can spend their valuable time in other tasks which can lead to increase in productivity of the project.

Fig:3.1: System Architecture



IV. Conclusion

All the papers have discussed about allocation of a task to a team member by considering the manager's decision which will not map to today's modern software development. All the methodologies which are discussed need some kind of manual processing of data before deciding to allocate the task, but in real-world to implement methodologies like Cluster Analysis, Integer Programming, SBSE, GA will need a lot of manual work and cost to implement. We suggest a Machine learning-based technique to allocate the task for the employee which uses the historical project data that is existing in the organization by considering the history of the employee and assuring that the right person is assigned the right task where he is good at. This proposed model will take the employee information and previous project allocation information, then apply ML algorithms to allocate the task.

REFERENCES:

[1] Luis Daniel Otero, Grisselle Centeno, Alex J. Ruiz-Torres, Carlos E. Otero, A systematic approach for resource allocation in software projects, Computers & Industrial Engineering, Volume 56, Issue 4,2009.

[2] C. E. Otero, L. D. Otero, I. Weissberger, and A. Qureshi, "A Multi-criteria Decision-Making Approach for Resource Allocation in Software Engineering," in *2010 12th International Conference on Computer Modelling and Simulation*, Cambridge, United Kingdom, 2010, pp. 137–141, DOI: 10.1109/UKSIM.2010.32.

[3] J. Park, D. Seo, G. Hong, D. Shin, J. Hwa, and D.-H. Bae, "Human Resource Allocation in Software Project with Practical Considerations," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 25, no. 01, pp. 5–26, Feb. 2015, DOI: 10.1142/S021819401540001X.

[4] J. Xiao, Q. Wang, M. Li, Ye Yang, Fan Zhang, and Lizi Xie, "A constraint-driven human resource scheduling method in software development and maintenance process," in *2008 IEEE International Conference on Software Maintenance*, Beijing, China, Sep. 2008, pp. 17–26, DOI: 10.1109/ICSM.2008.4658050.

[5] N. Bibi, A. Ahsan, and Z. Anwar, "Project resource allocation optimization using search-based software engineering — A framework," in *Ninth International Conference on Digital Information*

Management (ICDIM 2014), Phitsanulok, Thailand, Sep. 2014, pp. 226–229, DOI:10.1109/ICDIM.2014.6991431.

[6] W. Aslam and F. Ijaz, "A Quantitative Framework for Task Allocation in Distributed Agile Software Development," *IEEE Access*, vol. 6, pp. 15380–15390, 2018, DOI: 10.1109/ACCESS.2018.2803685.

[7] M. Farhangian, M. Purvis, M. Purvis, and T. B. R. Savarimuthu, "Agent-Based Modeling of Resource Allocation in Software Projects Based on Personality and Skill," in *Advances in Social Computing and Multiagent Systems*, vol. 541, F. Koch, C. Guttmann, and D. Busquets, Eds. Cham: Springer International Publishing, 2015, pp. 130–146.

[8] T. J. A. Santos, A. M. Lima, C. A. L. Reis, and R. Q. Reis, "Automated support for human resource allocation in software process by cluster analysis," in *Proceedings of the 4th International Workshop on Recommendation Systems for Software Engineering - RSSE 2014*, Hyderabad, India, 2014, pp. 30–31, DOI: 10.1145/2593822.2593830.

[9] S. Chakraverty, A. Sachdeva, and A. Singh, "A genetic algorithm for task allocation in collaborative software development using formal concept analysis," in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, Jaipur, India, May 2014, pp. 1–6, DOI: 10.1109/ICRAIE.2014.6909305.

[10] H. Y. Chiang and B. M. T. Lin, "A Decision Model for Human Resource Allocation in Project Management of Software Development," *IEEE Access*, vol. 8, pp. 38073–38081, 2020, DOI: 10.1109/ACCESS.2020.2975829.

[11] Yichuan Jiang and Jiuchuan Jiang, "Contextual Resource Negotiation-Based Task Allocation and Load Balancing in Complex Software Systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 5, pp. 641–653, May 2009, DOI: 10.1109/TPDS.2008.133.

[12] P. C. Pendharkar and J. A. Rodger, "An empirical study of the impact of team size on software development effort," *Inf. Technol. Manag.*, vol. 8, no. 4, pp. 253–262, Nov. 2007, DOI: 10.1007/s10799-006-0005-3.

[13] Santos, T. J. A., Lima, A. M., Reis, C. A. L., & Reis, R. Q. (2014). Automated support for human resource allocation in software process by cluster analysis. *4th International Workshop on Recommendation Systems for Software Engineering, RSSE 2014 - Proceedings*, 30–31. https://doi.org/10.1145/2593822.2593830.

[14] Zhu, Q., & Ren, Z. (2010). Research on human resource configuration strategy in software engineering. *Proceedings of the International Conference on E-Business and E-Government, ICEE 2010*, 2764–2767. https://doi.org/10.1109/ICEE.2010.69

[15] Lamersdorf, A., & Rombach, D. (2009). Empirically-based decision support for task allocation in global software development. *Proceedings - 2009 4th IEEE International Conference on Global Software Engineering, ICGSE 2009*, 281–284. https://doi.org/10.1109/ICGSE.2009.38

[16] A. Barreto, M. D. O. Barros, and C. M. L. Werner, ``Staf_ng a software project: A constraint satisfaction and optimization-based approach," *Comput. Oper. Res.*, vol. 35, no. 10, pp. 3073_3089, Oct. 2008.

[17] G. Antoniol, M. Di Penta, and M. Harman., "Search-based techniques for optimization software project resource allocation." *In Genetic and Evolutionary Computation–GECCO, Springer Berlin Heidelberg*, January 19, 2004, pp.1425-1426.

[18] Z. Anwar., and A. Ahsan, "Comparative analysis of MOGA, NSGA-II and MOPSO for regression test suite optimization." *International Journal of Software Engineering (IJSE)*, vol. 1, No. 7, pp. 41-56, January 2014.

[19] Q.Bai, "Analysis of particle swarm optimization algorithm." *Computer and Information Science*, CCSE, Vol. 3, No. 1, pp 180-184, Feb 2010.

[20] WN. Chen, and J. Zhang, "Ant colony optimization for software project scheduling and staffing with an event-based scheduler." *IEEE Transactions on Software Engineering*, Vol. 39, NO. 1, January 2013, pp 1- 17.

[21] A. Lamersdorf and J. Münch, "A Survey on the State of the Practice in Distributed Software Development: Criteria for Task Allocation," in *2009 Fourth IEEE International*

Conference on Global Software Engineering.

[22] I. Richardson, V. Casey, D. Zage and W. Zage, "Global Software Development – the Challenges," University of Limerick, Ball State University: SERC Technical Report 278, Limerick, 2005.

[23] A. Fernández-del Viso Torre, A. Lamersdorf and J. Münch, "A Riskdriven Model for Work Allocation in Global Software Development Projects," in *Sixth IEEE International Conference on Global Software Engineering*, 2011.

[24] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Pub Co, 1989.

[25] André, M., Baldoquín, M.G., Acuña, S.T.: Formal model for assigning human resources to teams in software projects. Inf. Softw. Technol. 53(3), 259–275 (2011)

[26]. LePine, J.A., Buckman, B.R., Crawford, E.R., Methot, J.R.: A review of research on personality in teams: accounting for pathways spanning levels of theory and analysis. Hum. Resour. Manage. Rev. 21(4), 311–330 (2011)

[27]. Wood, R.E.: Task complexity: Definition of the construct. Organ. Behav. Hum. Decis. Process. 37(1), 60–82 (1986)
[28]. Zoethout, K., Jager, W., Molleman, E.: Task dynamics in self-organizing task groups: expertise, motivational, and performance differences of specialists and generalists. Auton. Agent. Multi. Agent. Syst. 16(1), 75–94 (2007)
[29]. Jiang, G., Hu, B., Wang, Y.: Agent-based simulation approach to understanding the interaction between employee behavior and dynamic tasks. Simulation 87(5), 407–42(2010) [30]. Acuña, S.T., Gómez, M., Juristo, N.: How do personality, team processes, and task characteristics relate to job satisfaction and software quality? Inf. Softw. Technol. 51(3), 627–639 (2009)
[31] Myers, I.B., McCaulley, M.H.: Manual: a Guide to the Development and Use of the Myers-Briggs Type Indicator. Consulting Psychologists Press (1985)
[32] Belbin, R.M.: Team Roles at Work. Routledge, London (2012)
[33] Myers, I.: The Myers-Briggs Type Indicator. Consulting Psychologists Press (1962) [34]. Jung, C.G.: Psychological Types: or the Psychology of Individuation. Harcourt, Brace(1921)
[35]. Belbin, R.M.: Management Teams: Why They Succeed or Fail. Bulletin of the British Psychological Society (1981)
[36]. Stevens, K.T.: The effects of roles and personality characteristics on software development team effectiveness. Dissertation, Virginia Polytechnic Institute and State University (1998)
[37] P. Jalote and B. Vishal, "Optimal resource allocation for the quality control process," in *Proceedings of the 14th International Symposium on Software Reliability Engineering*, Denver, Colo, USA, November 2003.
[38] W. Heijstek and M. R. V. Chaudron, "Effort distribution in model-based development," in *the 2nd Workshop on Model Size Metrics, and the 10th International Conference on Model*
Driven Engineering Languages and Systems, Nashville, Tenn, USA, 2007.
[39] P. Yiftachel, D. Peled, I. Hadar, and D. Goldwasser, "Resource allocation among development phases: an economic approach," in *Proceedings of Economic Driven Software Engineering Research Workshop, the 28th International Conference on Software Engineering*, pp. 43–48, Shanghai, China, 2006.
[40] P. L. Bannerman, E. Hossain, and R. Jeffery, Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? in *Proc*.
45th Hawaii Int. Conf. System Sciences, Maui, HI, USA, 2012, pp. 5309–5318.
[41] H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, and B. Fitzgerald, Global software development challenges:
A case study on temporal, geographical and socio-cultural
distance, in *Proc. 2006 IEEE Int. Conf. Global Software Engineering*, Florianopolis, Brazil, 2006, pp. 3–11.
[42] M. Jones and U. Mortensen, *Guide to Software Project Management*. Noordwijk, Netherlands: ESA Board for Software Standardisation and Control (BSSC), 1995.
[43] M. André, M. G. Baldoquín, and S. T. Acuña, ``Formal model for assigning human resources to teams in software projects," *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 259_275, Mar. 2011.
[44] S. T. Acuña and N. Juristo, ``Assigning people to roles in software projects," *Softw., Pract. Exper.*, vol. 34, no. 7, pp. 675_696, Jun. 2004.
[45] F. Q. B. da Silva, A. C. C. Franca, T. B. Gouveia, C. V. F. Monteiro, E. S. F. Cardozo, and M. Suassuna,
``An empirical study on the use of team building criteria in software projects," in *Proc. Int. Symp. Empirical Softw. Eng. Meas.*, Sep. 2011, pp. 58_67.
[46] A. R. Gill, J. Jaafar, M. Omar, S. Basri, and A. Waqas, ``A rule-based model for software development team composition: Team leader role with personality types and gender classi_cation," *Inf. Softw. Technol.*, vol. 74, pp.105_113, Jun. 2016.
[47] H.-T. Tsai, H. Moskowitz, and L.-H. Lee, ``Human resource selection for software development projects using Taguchi's parameter design," *Eur. J. Oper. Res.*, vol. 151, no. 1, pp. 167_180, Nov. 2003.