A Project Human Resource Allocation Method Based on Software Architecture and Social Network

Lixin Zhou
School of Software and Microelectronics, Peking University, 102600
Beijing, China
Lxzhou@uiuc.edu

Abstract —— **The allocation of human resource is one of the most essential aspects in a project management. A situation with employees working on tasks that they are not well-suited for can lead to a significant loss of time and resources in addition to a sub-par product or service. The simple difference between a good and bad task assignment for employees can easily result in major differences in a company's bottom line.**

In this paper, we present a project human resource allocation method based on software architecture and social network in a project organization. An algorithm for matching employees and tasks is put forward which are based on task attributes, software architecture, employee skills and employee preference and social network relations. As a result, we have created a new human resource allocation method in a project management.

Keywords – Human resource, work package, WBS, software architecture, interpersonal relation.

1. INTRODUCTION

The human element is one of the most important but frequently overlooked aspects of managing IT projects. A project's success directly relates to the quality of talent employed, and, more importantly, the manner in which management deploys talent on the project.[1] Some implemented solutions often ignore employee preference of which tasks they work on which inhibits employee productivity [2]. Those that do take employee preference into consideration are non-deterministic, thus making it more difficult to evaluate the business process in an efficient manner. [3]

In the process of task assignment, we don't consider the relations between tasks. In a software development project, software architecture has impacts on organizational structure. As long ago as 1968，the close relationship between an architecture and the organization that produced it was a subject of comment. Conway [4] makes the point as follows：

Take any two nodes x and y of the system. Either they are joined by a branch or they are not.（That is，either they communicate with each other in some way meaningful to the operation of the system or they do not.）If there is a branch，then the two（not necessarily distinct）design groups X and Y which designed the two nodes must have negotiated and agreed upon an interface specification to permit communication between the two corresponding nodes of the design organization. If on the other hand，there is no branch between x and y, then the subsystems do not communicate with each other, there was nothing for the two corresponding design groups to negotiate，and therefore there is no branch between X and Y.

Conway was describing how to discern organizational structure（at least in terms of communication paths）from system structure，but the relationship between organizational and system structures is bidirectional，and necessarily so.

The interpersonal relations also have impacts on the task assignment.

We have taken into consideration the satisfaction and ownership of the problem by the employee and also the attributes of the tasks for which they are most qualified for and to distribute the work load as evenly as logistically possible. In this method, we also consider the relation between tasks and the interpersonal relations. The existing applied approaches to this problem involve a manager assigning tasks without considering the relation between tasks effectively and the relations between employees.

2.THE ALGORITHM PRINCIPLES

2.1 Work Breakdown Structure (WBS)

The WBS is a deliverable-oriented hierarchical decomposition of the work to be executed by the project team, to accomplish the project objectives and create the required deliverables. The WBS organizes and defines the total scope of the project. The WBS subdivides the project work into smaller, more manageable pieces of work, with each descending level of the WBS representing an increasingly detailed definition of the project work. The planned work contained within the lowest-level WBS components, which are called work packages, can be scheduled, cost estimated, monitored, and controlled. [5]

In this paper, work packages, the lowest-level WBS components are defined as follows:

Work package ::= work package id + name + Work package skill requirements.

Work package Skill Requirements ::= {analysis, OO Design, C++, Java, UML, ……}

2.2 Software Architecture

The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships between them.[6]

The impact of an architecture on the development of organizational structure is clear. Once an architecture for the system under construction has been agreed on，teams are allocated to work on the major modules and a work breakdown structure is created that reflects those teams. Each team then creates its own internal work practices（or a system-wide set of practices is adopted）. For large systems，the teams may belong to different subcontractors. The work practices may include items such as bulletin boards and Web pages for communication, naming conventions for files，and the configuration control system. All of these may be different from group to group， again especially for large systems． Furthermore， quality assurance and testing procedures are set up for each group，and each group needs to establish liaisons and coordinate with the other groups.

Thus， the teams within an organization work on modules. Within the team there needs to be high－bandwidth communications：Much information in the form of detailed design decisions is being constantly shared. Between teams，low-bandwidth communications are sufficient and in fact crucial (Fred Brooks's contention is that the overhead of inter-team communication, if not carefully managed，will swamp a project）This，of course，assumes that the system has been designed with appropriate separation of concerns.

Highly complex systems result when these design criteria are not met. In fact，team structure and controlling team interactions often turn out to be important factors affecting a large project's success. If interactions between the teams need to be complex，either the interactions among the elements they are creating are needlessly complex or the requirements for those elements were not sufficiently “hardened” before development commenced. In this case，there is a need for high-bandwidth connections between teams，not just within teams，requiring substantial negotiations and often rework of elements and their interfaces. Like software systems，teams should strive for loose coupling and high cohesion.[6]

So, if two work packages have relations according to the software architecture in a project, we define the relation between the two work packages using a Work Package Relation Weight,

Work Package Relation Weight ::= {wpw_{ij} | $0 < wpw_{ij} < 1$, wpw_{ij} is a decimal and ∃relations between work package i and work package j according to the software architecture in a project }

2.3 Social Network

There are many kinds of interpersonal relations between employees, such as classmate relation, neighbor relation and townee relation. We define the relation between the two employees using an Employee Relation Weight,

Employee Relation Weight ::= {erw_{ij} | $0 < erw_{ij} < 1$, erw_{ij} is a decimal and ∃relations between employee i and employee j }

2.4. The n-body problem[1]

One reason creating a good team dynamic is so difficult is that the number of working relationships grows as a polynomial function of *n*, the number of people on the team. I call this the *n*-body problem. Figure 1 depicts this problem.

In fact, you can easily prove (by induction) that for *n* people on a team, there are $n(n - 1)/2$ possible working relationships, and any of them can sour. Further, the quality of a working relationship is not transitive. For example, Fred might work well with Jane, and Jane might work well with Bob, but this does not necessarily imply that Fred and Bob work well together. Finally, complicating these interactions are cultural differences that you must consider when building and managing teams, planning projects, and dealing with difficult personal situations, according to MacDonald. Taken another way, it is unwise to ignore interpersonal interactions and view staff as simply “headcount.” As noted software developer Fred Brooks postulated in what is now known as Brooks’ law, adding manpower to a late software project just makes it later . [7]

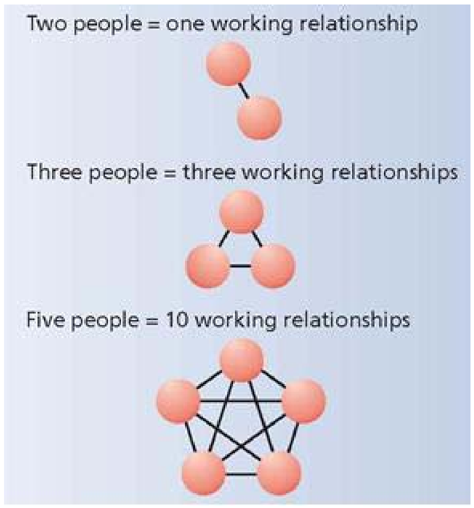
Two people = one working relationship
Three people = three working relationships
Five people = 10 working relationships

Fig.1 The number of working relationships grows as a function of the number of team members.

3. A PROJECT HUMAN RESOURCE ALGORITHM

An employee is defined as follows :

An employee ::= employee skills + preference + potential

Employee skills ::= {analysis, OO Design, C++, Java, UML, ……}

Preference ::= decimal

Potential :: = decimal

The steps of the algorithm presented in this paper are as follows,

Step1. Assign a work package to an employee who has the skills the work package requires and calculate the skill match score,

SkillMatchScore =

$$\overrightarrow{WorkPackageSkill\,\mathrm{Re}quirements} \bullet \overrightarrow{EmployeeSkills}$$

Step2. Add employee preference and potential impacts to the score,

WorkEmployeeMatchScore =
SkillMatchScore * preference * potential

Then, a system level score is calculated as,

SystemScore = ∑ WorkEmployeeMatchScore (for all work packages and employees)

Step3. Add software architecture impacts to the system score,

If two work packages have relations and the two work package have been assigned to the same development group, then,

SystemScore = SystemScore – WorkPackageWeight/2

If two work packages have relations and the two work package have been assigned to two different development group, then,

SystemScore = SystemScore – WorkPackageWeight

Step4. Add interpersonal relations impacts to the system score, If two employees have interpersonal relations and the two employees are in the same development group, then,

SystemScore = SystemScore + EmployeeRelationWeight

If two employees have interpersonal relations and the two employees are in two different development groups, then,

SystemScore = SystemScore – EmployeeRelationWeight

Step5. Repeat Recursively from step1 to step4 for another work package assignment solution.

Finally we will obtain different solutions with different system scores. We will choose the highest score work package assignment solution.

4. EXAMPLE

An E-business system has three subsystems, one is the product information management subsystem (PIMS), one is the custom information management subsystem (CIMS), another is the order information management subsystem (OIMS). In the process of development, a Client-Server software architecture style is adopted. So the E-business system has been divided into 6 modules : product information management subsystem client, custom information management subsystem client, order information management subsystem client, product information management subsystem server, custom information management subsystem server and order information management subsystem server. The skills which modules require are as shown in Table 1.The relations between one module and other modules are shown in table 2.

Table 1

Work packages (Modules)	Skill requirements
PIMS client	HTML, JSP, Javascript
CIMS client	
OIMS client	
PIMS server	Java, Database, Tomcat
CIMS server	
OIMS server	

Table 2

	PIMS client	CIMS client	OIMS client	PIMS server	CIMS server	OIMS server
PIMS client		0.3	0.3	1		
CIMS client	0.3		0.3		1	
OIMS client	0.3	0.3				1
PIMS server	1				0.3	0.3
CIMS server		1		0.3		0.3
OIMS server			1	0.3	0.3	

Work Package Relation Weight

The available employees are shown in Table 3.

Table 3

Employee Name	Skills	preference	potential
Jack	HTML, JSP, Javascript, ASP	1	1
Michael	HTML, JSP, Javascript	1	1
Richard	HTML, JSP, Javascript	0.1	0.5
Mary	HTML, JSP, Javascript, ASP	1	1
Robert	Java, Database, Tomcat	1	1
Ben	Java, Database, Tomcat, C++	1	1
Allen	Java, Database, Tomcat	1	1

The interpersonal relations (Employee Relation Weight) between employees are shown in Table 4.

Table 4

	Jack	Michael	Mary	Robert	Ben	Allen
Jack		1		0.5		
Michael	1					
Mary					0.8	
Robert	0.5					0.5
Ben			0.8			
Allen				0.5		

Employee Relation Weight

There are three solutions given in this paper which are shown in Table 5, Table 6, Table 7.

Table 5 , Solution 1

Employee Name	Work package	Team	Score
Michael	PIMS client	Team 1	10.95
Richard	CIMS client		
Mary	OIMS client		
Robert	PIMS server	Team 2	
Ben	CIMS server		
Allen	OIMS server		

Table 6, Solution 2

Employee Name	Work package	Team	Score
Jack	PIMS client	Team 1	14.3
Michael	CIMS client		
Mary	OIMS client		
Robert	PIMS server	Team 2	
Ben	CIMS server		
Allen	OIMS server		

Table 7, Solution 3

Employee Name	Work package	Team	Score
Jack	PIMS client	Team 1	14.5
Robert	PIMS server		
Mary	OIMS client	Team 2	
Ben	OIMS server		
Michael	CIMS client	Team 3	
Allen	CIMS server		

In Solution 1, although Richard has the skills which assigned work package requires, he has lower preference and potential, so the Solution 1 obtains the lowest score. Solution 2 and Solution 3 have assigned the work packages on the same set of employees, but they have different organization structures. The organizational structure of Solution 3 is better than the organizational structure of Solution 2. So, we can select Solution 3. In our software development project management experience, a project adopted the Solution 2 failed, and another project adopted the Solution 3 succeeded.

5. CONCLUSION

In this paper, we present a project human resource allocation method based on software architecture and social network in a project organization. An algorithm for matching employees and tasks is put forward which are based on task attributes, software architecture, employee skills and employee preference and social network relations.

An example has been presented in this paper. The example proves the method is useful and effective.

Reference

[1]Philip Laplante. Remember the human element in IT project management**.** IT Professional, Volume 5, Issue 1, Jan/Feb 2003 Page(s): 46 – 50.

[2]Ivancevich, John M. "High and Low Task Stimulation Jobs: A Causal Analysis of Performance-Satisfaction Relationships." Academy of Management Journal, Vol. 22, No. 2, p. 206-222, 1979.

[3]Lagesse, B. A Game-Theoretical Model for Task Assignment in Project Management. Management of Innovation and Technology, 2006 IEEE International Conference on Volume 2, Issue , June 2006 Page(s):678 – 680.

[4]Conway, M. "How Do Committees Invent ?" Datamation 14(4), 1968.

[5]PMI.PMBOK, Third Edition.

[6]Bass, Len; Paul Clements, Rick Kazman (2003). *Software Architecture In Practice, Second Edition.* Boston:

Addison-Wesley, p. 21-24. ISBN 0-321-15495-9.
[7] Frederick P. Brooks, *The Mythical Man-Month*, Addison-Wesley, 1975.