Distributing Human Resources among Software Development Projects[1]

Macario Polo, María Dolores Mateos, Mario Piattini and Francisco Ruiz

Summary

This paper presents a method for estimating the distribution of human resources to be assigned to development projects. The estimation calculates the optimal distribution from the economical point of view (i.e.: that which less costs). To accomplish this, we built an economical model to distribute human resources among projects in one organisation. The equations and algorithms used to do this are presented in the paper. We also briefly present a tool which does these estimations.

1. Introduction

In the last few years, an increasing demand for personnel qualified in Information Technologies has been observed, which greatly exceeds the offer: in 1998 there was in Europe 500,000 free positions in IT, with an expected increase of up to 2.4 million people for the year 2004 [1]. Some reports from the European Commission also confirm this tendency [2][3], which is reviewed by daily newspapers almost every month. The situation in the U.S.A. is quite similar: the yearly number of visas for foreigners qualified in Information Technologies has been increased to the point of 300,000 ones, although big software organizations have solicited a greater number of licenses.

With this situation, software organizations must do an adequate planning of their human resources among the different projects that they are carrying out. An additional difficulty is to fix the number of development projects to be accepted by a software organization, when many times it is known that the organization has not (and will not have) people enough to execute all the projects in time and budget. However, the rejection of a software project will cause probably the loss of that customer for the organization, and maybe a negative impact on other potential customers.

Therefore, it is important to accept projects, but it is also a basic activity to distribute the human resources in an adequate way among all of them, and this one is the main goal of this paper.

We propose to make an economical model of the portfolio of software development projects, in order to estimate the optimal quantity of human resources to be devoted to each project, during every day of the considered period. We have successfully tested the method with several projects which use different life cycle models, although we have reasons to believe that it can be easily adaptable to still-non-studied life cycles. This method is a very advanced evolution of the work presented last year in this same forum [4].

The paper is organized as follows: Section 2 explains the model we use to represent a portfolio of development projects from an economical point of view, which includes several equations. In Section 3, an algorithm to calculate the distribution is presented, as

[1] This work is part of the MPM and MANTIS projects. MPM is developed with Atos ODS, S.A. and partially supported by the Ministerio de Ciencia y Tecnología, Programa de Tecnologías de la Información y las Comunicaciones (FIT-070000-2000-307); MANTIS is partially supported by the European Union and CICYT (1FD197-1608TIC).

well as CRED, a tool we have developed to do estimations. Section 4 exposes our conclusions and future and immediate work lines.

2. Economical model of a portfolio of development projects

One of the primary goals of the software organization is to obtain economical benefits from the developments. Independently of the life cycle selected for a given project, this one can be considered as the union of several subprojects. Sometimes, the consecution of a subproject will be a must to start another one (in the waterfall life cycle model, the requirement analysis comes before the design phase), but other times, the life cycle selected allows to develop in parallel different parts of the system. Depending on the contract between the customer and the supplier organizations, maybe some of those subprojects must be delivered according to a previously signed scheduling. For example, a system developed using the Unified Process can be delivered in several releases, each one with some increment of functionality with respect to the previous one. These successive releases constitute partial results whose date of deliver, prize and possible sanctions by delay may be covenanted in the development contract.

2.1. Maximization equation

The software organization must take into account all these factors to do an adequate resource assignment to each project and subproject, with the primary goal of maximize its economical benefits. This can be represented by the following equation:

$$Max\,B = I - C = \sum_{i=1}^{N} (Ii - Ci) \quad \textbf{Eq. 1}$$

In Eq. 1., *B* represents the benefits produced by the *N* development projects in the portfolio; *I* and *C* respectively are the total incomes and costs, whereas *Ii* and *Ci* represent the incomes and costs of the *i-th* project.

As every project is an aggregation of subprojects, Eq. 1 can be rewritten in this way:

$$Max\,B = \sum_{i=1}^{N} \sum_{j=1}^{Ni} (Iij - Cij) \qquad \textbf{Eq. 2}$$

In Eq. 2, *Ni* is the number of subprojects if the *i-th* project, and *Iij* and *Cij* are, respectively, the incomes and costs of the *j-th* subproject of the *i-th* project (thereinafter, the *ij* subproject). Depending so on the contract as on the selected life cycle, sometimes *Iij* will be zero, since not all subprojects will be delivered to the customer and they will not produce any income. In the other side, *Iij* is typically covenanted in the development contract, since the customer organization needs to know the project budget before its assignment to the development organization.

Costs of each subproject are different: each one takes more or less time than another one, and the people in charge of them have different levels and salaries (analysts, programmers, tests engineers...). Another influencing factor on the costs of a subproject (and, therefore, on the costs of the complete project) is the existence of delays in the date of delivering and of sanctions by delays. With this assumption, costs of the *ij* project can be represented in this manner:

$$Cij = Rij + Qij \quad \textbf{Eq. 3}$$

Rij are the costs of the resources devoted to the subproject, whereas *Qij* is the quantity to be paid by sanctions due to delays in the *ij* subproject. We suppose that a subproject needs only one type of resource (analysts, for example): otherwise, it could be divided into more subprojects. With this consideration, *Rij* can be expressed in this way:

$$Rij = \sum_{t=1}^{F} (Tijt \cdot ht) \qquad \textbf{Eq. 4}$$

In Eq. 4, *Tijt* is the number of hours of the *t-th* resource needed to execute the *ij* subproject; *F* is the number of different resources (analysts, programmers, test engineers, etc.); *ht* represents the cost of one hour of resource of the *t* type.

The second variable of Eq. 3, *Qij*, depends on the scheduled date of delivering (or scheduled duration) for the *ij* project and on the real date (or real duration). At a first sight, *Qij* can be expressed in this way:

$$Qij = Sij \cdot (Eij - Pij) \quad \textbf{Eq. 5}$$

In Eq. 5, *Sij* is the covenanted sanction for the *ij* subproject (obviously, if such subproject will not be delivered, it will not have any sanction, although it could have some influence on next subprojects). *Eij* is the real number of days used to finish the *ij* project, whereas *Pij* is the scheduled duration, also in days, which depends on the total number of hours of effort required by the *ij* project ($\sum_{t=1}^{F} Tijt$).

Pij was estimated by the software organization using some estimation method and is covenanted in the contract. The value of *Eij* depends on the number of hours devoted to the subproject. In a given subproject, we will reach the value of *Eij* when the sum of the hours devoted each day to the *ij* project will be equal to the number of required hours of the *t-th* resource, *Tijt*. In other words:

$$Eij = m \quad / \sum_{k=1}^{m} \sum_{t=1}^{F} e_{ijk}^{t} = \sum_{t=1}^{F} Tijt \quad \textbf{Eq. 6}$$

In Eq. 6, e^{t}_{ijk} is the number of hours of the *t*-th resource devoted to the *ij* project in the *k-th* day.

At his point, we have characterized all the variables we need to rewrite Eq. 1:

$$\begin{aligned} Max\, B = I - C = \sum_{i=1}^{N} (Ii - Ci) = \sum_{i=1}^{N} \sum_{j=1}^{Ni} (Iij - Cij) = \sum_{i=1}^{N} \sum_{j=1}^{Ni} [Iij - Rij - Qij] = \\ = \sum_{i=1}^{N} \sum_{j=1}^{Ni} \left[Iij - \left(\sum_{t=1}^{F} (Tijt \cdot ht) \right) - Sij \cdot (Eij - Pij) \right] \end{aligned} \quad \textbf{Eq. 7}$$

In Eq. 7, the only unknown quantity is *Eij*, since the rest have previously assigned values.

2.2. Constraints

To estimate the optimal distribution of resources, we must identify now the possible constraints to be applied to Eq. 7.

The first constraint represents the fact that the sum of the hours of a given resource devoted during a concrete day to all subprojects cannot be greater than the available number of hours of that kind of resource in that day (*A*₁, *A*₂…):

$$\begin{cases} e^t_{1j1} + e^t_{2j1} + \ldots + e^t_{Nj1} \le A_{t1} & \forall t \in [1, F], \forall j \\ e^t_{1j2} + e^t_{2j2} + \ldots + e^t_{Nj2} \le A_{t2} & \forall t \in [1, F], \forall j \\ \ldots \end{cases}$$

The *t* subindex is used to group resources by its type (i.e.: analysts, programmers...). Eq. 8 summarizes the previous equations:

$$\sum_{i=1}^{N} e^t_{ijk} \le A_{tK}, \ \forall t, k \qquad \textbf{Eq. 8}$$

The next constraint means that the time devoted to a subproject must be equal to the time required by it:

$$\sum_{j=1}^{Ni} e^t_{ijk} = Tijt, \ \forall i, k, t \qquad \textbf{Eq. 9}$$

The following simple constraints determine the minimal values of the variables that intervene in Eq. 7:

$$\begin{array}{ll} Iij \ge 0 \quad \forall \mathrm{i, j} & \\ hij \ge 0 \quad \forall \mathrm{i, j} & \\ Sij \ge 0 \quad \forall \mathrm{i, j} & \\ T^t_{ij} \ge 0 \quad \forall \mathrm{i, j, t} & \textbf{Eq. 10 to 15} \\ e_{ijk} \ge 0 \quad \forall \mathrm{i, j, k} & \\ A_{tK} \ge 0 \quad \forall \mathrm{j, k} & \end{array}$$

2.3. Complete model (for reference)

With all the equations stated in the previous subsections, the portfolio of projects can be modelled with the following equation:

$$\left\{\begin{array}{l}
\text{Max } B = \sum_{i=1}^{N}\sum_{j=1}^{Ni}\left[Iij - \left(\sum_{t=1}^{F}\left(Tijt \cdot ht \right) \right) - Sij \cdot \left(Eij - Pij \right) \right] \\
\text{subject to :} \\
\sum_{i=1}^{N} e_{ijk}^{t} \leq A_{tK},\ \forall t,k \\
\sum_{j=1}^{N} e_{ijk}^{t} = Tijt,\ \forall i,k,t \\
Iij \geq 0 \quad \forall \mathrm{i, j} \\
ht \geq 0 \quad \forall \mathrm{i, j} \\
Sij \geq 0 \quad \forall \mathrm{i, j} \\
Tijt \geq 0 \quad \forall \mathrm{i, j, t} \\
e_{ijk}^{t} \geq 0 \quad \forall \mathrm{i, j, k} \\
A_{tK} \geq 0 \quad \forall \mathrm{j, k}
\end{array}\right.$$

The meaning of each variable in the left side is the following:

- N = number of projects in the portfolio
- Ni = number of subprojects in the *i-th* project
- Iij = incomes by the *ij* subproject
- F = number of different types of resources
- $Tijt$ = number of hours required of resource type *t* by the *ij* subproject
- ht = cost of each hour of resource devoted to the *ij* subproject
- Sij = sanction to be paid by each day of delay in the deliver of the *ij* subproject
- Eij = real number of days used to finish the *ij* project
- Pij is the scheduled duration (in days) of the *ij* subproject
- e_{ij}^{t} = resources devoted the *k-th* day to the *ij* subproject
- $A_{t\,K}$ = available resources of the type *t* in the *k-th* day

It is important to remember the following definition of *Eij* (Eq. 6): $Eij = m \ / \sum_{k=1}^{m}\sum_{t=1}^{F} e_{ijk}^{t} = \sum_{t=1}^{F} Tijt$. This equation is the basis to estimate the optimal distribution of resources.

3. Resources estimation

All we need to estimate the resources to be assigned to every development project in the portfolio is to resolve the equation shown in Section 2.3. As not all the constraints are lineal (see Eq. 6), the simplex method cannot be used. Some of the candidate methods to resolve this kind of equations are Genetic algorithms or Simulated annealing. However, as the number of unknown values is very little, we can find a very good approximation to the optimal solution testing all the possible combinations.

3.1. Algorythm

A “coarse” version of the algorithm could be the following:

```
max =-MAXDOUBLE
for k=0 to maxDuration
      for i=1 to N do
            for j=1 to F
                  for e=0 to A[t,k]
                        e[i,j,k,t]=e
                        if Benefit()>max then
                              max=Benefit()
                              saveParameters(i,j,k,e)
                        endif
                  end
            end
      end
end
```

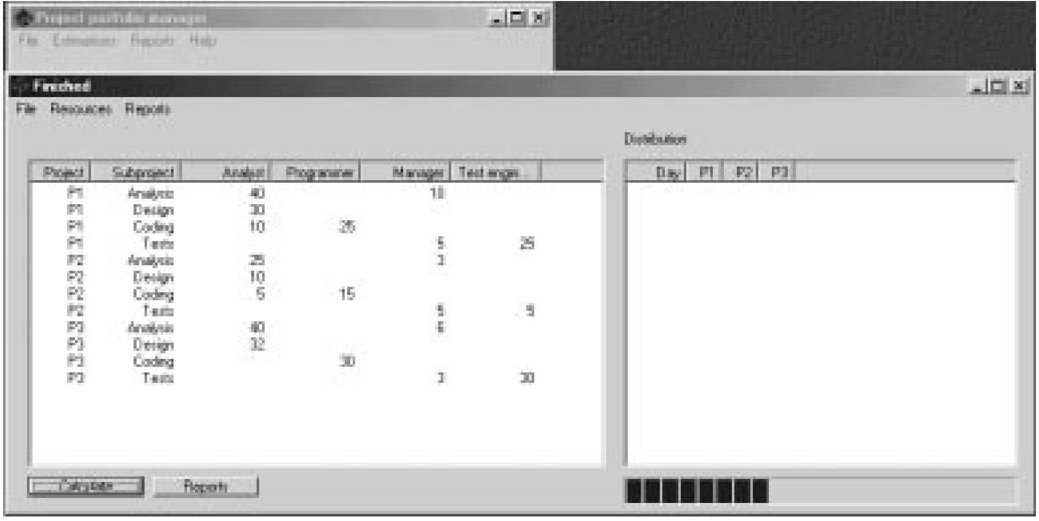
Figure 1. First version of the algorythm.

Obviously, the code in Figure 1 can be very optimized through the application of the constraints and some other rules. For example:

- We will assign resources (assignment of values to the *e* array in Figure 1) when:
  - The *ij* project needs them
  - In the *k-th* day, there are available resources of the type which is being tested.
- We will calculate the benefit (Eq. 7) when the current combination which is being tested implies the finalization of all the projects in the portfolio.
- The *e* array can be a vector (with the consequent memory saving): only some transformation rules are required.

3.2. CRED: a tool for estimating resources distribution

In order to do the automatic estimation of resources, we have developed CRED, a tool which helps in the calculus of the resources distribution. It implements an optimised version of the algorithm previous. Figure 2 shows one the CRED’s screens, in the moment of doing an estimation.

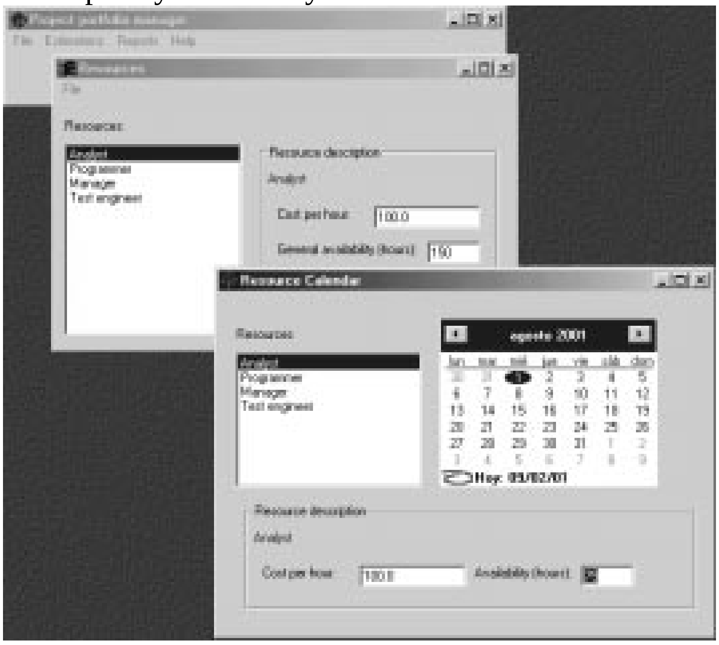


Project	Subproject	Analyst	Programmer	Manager	Test engin...
P1	Analysis	40		10	
P1	Design	30			
P1	Coding	10	25		
P1	Tests			5	25
P2	Analysis	25		3	
P2	Design	10			
P2	Coding	5	15		
P2	Tests			5	5
P3	Analysis	40		6	
P3	Design	32			
P3	Coding		30		
P3	Tests			3	30

Day	P1	P2	P3

Figure 2. CRED during the calculus of a distribution.

CRED allows the tracking of the projects, to change the assignation, etc. In this manner, several kind of reports, statistics and graphics can be obtained, in order to analyze efforts, resources, etc., and compare them against a baseline. Figure 3 shows the screen used to change the temporary availability of resources.


Project portfolio manager
File Estimations Reports Help
Resources
File
Resource Calendar
agosto 2001
lun mar mié jue vie sáb dom
Hoy: 09/02/01

Resources:
Analyst
Programmer
Manager
Test engineer

Resource description
Analyst
Cost per hour: 100.0
General availability (hours): 150

Resource description
Analyst
Cost per hour: 100.0
Availability (hours):

Figure 3. Changing resources availability.

Figure 4 shows a view of the class diagrams used to build the tool: resources needed by a subproject are determined by the association with the “ResourceNecessity” class.

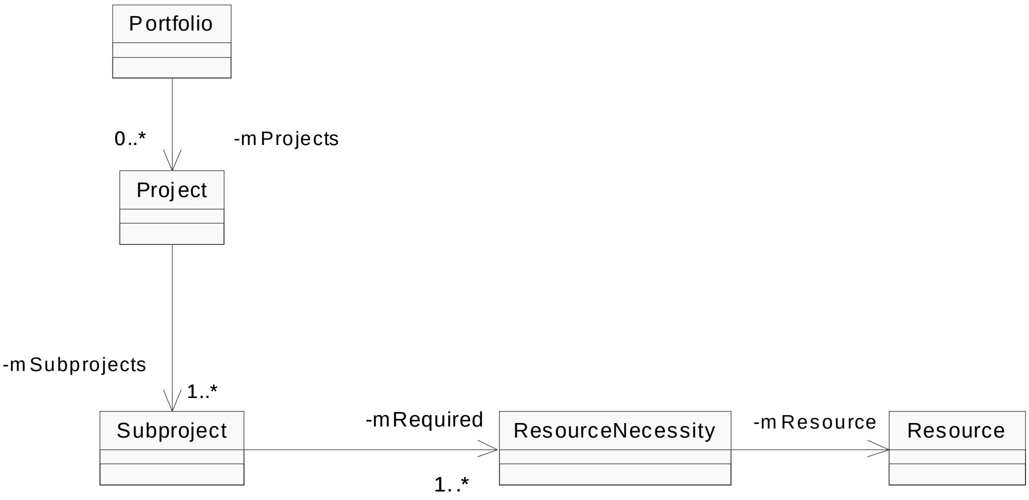


Figure 4. Conceptual diagram.

3.3. Context of CRED

CRED is now being adapted to allow its full integration in MANTIS [5], an integral environment for managing maintenance projects. MANTIS uses XMI to exchange information among the tools that compose it. In this context, CRED should be capable of sending its information to MANTIS in a standardized format based on XML, which is one of our current works.

4. Conclusions and future work

This paper has presented a method and a tool to estimate the distribution of resources to be assigned to development projects. The estimation attempts to assure an optimal distribution from the economical point of view.

Both the method and the tool must be modified in order to take into account:

- Dependencies (i.e.: a subproject cannot begin before the end of another one)
- Indirect costs (i.e.: the organization cannot accept a project because it will not have available resources)

5. References

[1] Bourke, T.M. (1999). Seven major ICT companies join the European Commission to work towards closing the skills gap in Europe (Press Release). Available also at (Nov., 7, 2000): http://www.career-space.com/whats_new/press_rel.doc

[2] European Commission (1999*). The competitiveness of European enterprises in the face of globalisation.* Available at (Nov, 7, 2000): http://europa.eu.int/comm/research/pdf/com98-718en.pdf

[3] European Commission (2000). *Employment in Europe 2000.* Available at (Nov., 7, 2000): http://europa.eu.int/comm/employment_social/empl&esf/docs/empleurope2000_en.pdf

[4] Polo, M., Piattini, M. and Ruiz, F. (2000). *Planning the non-planneable maintenance.* Project Control, The Human Factor: Proceedings of the ESCOM-SCOPE 2000 Combined Conference. Munich, Germany.

[5] Ruiz, F., Piattini, F. and Polo, M. (2001). An Integrated Environment for Managing Software Maintenance Projects. In van Bon (Ed.): World Class IT Service Management Guide, 2nd edition. Addison Wesley.