PROJECT CONTROL: SATISFYING THE CUSTOMER

Proceedings

of

ESCOM 2001

2-4 April 2001
London, United Kingdom

Proceedings of the 12th ESCOM (European Software Control and Metrics Conference)
Conference including the SCOPE track on Software Product Quality.

Editors

Katrina D. Maxwell
Datamax, France

Serge Oligny
Université du Québec à Montréal, Canada

Rob J. Kusters
Eindhoven University of Technology & Open University,
The Netherlands

Erik P.W.M. van Veenendaal
Improve Quality Services & Eindhoven University of Technology,
The Netherlands

ESCOM 2001 CONFERENCE OFFICERS

General Chair
John Jenkins (Middlesex University, United Kingdom)
ESCOM Programme Chair
Katrina Maxwell (Datamax, France)
Associate Programme Chair
Serge Oligny (Université du Québec à Montréal, Canada)
SCOPE track Chair
Erik van Veenendaal (Improve Quality Services & Eindhoven University of Technology, NL)
Publications Chair
Rob. J. Kusters (Eindhoven University of Technology and The Open University, NL)
ESCOM-METRICS Co-ordinator
Adrian Cowderoy (Nexusworld.net, UK)
Conference Manager
Monica Kaayk (Escom Conference Ltd., UK)

ESCOM 2001 PROGRAMME COMMITTEE

Galtierro Bazzana, Onion, Italy
Miklos Biro, MTA SZTAKI, Hungary
Lionel Briand, Carleton University., Canada
A. Winsor Brown, USC, USA
Giovanni Cantone, University of Rome, Italy
John Elliott, DERA, United Kingdom
Fred Heemstra, Open University, The Netherlands
Ross Jeffrey, University of New South Wales, Australia
Pekka Forselius, STTF, Finland
Michael O'Duffy, Centre for Software Engineering, Ireland
Shari Lawrence Pfleeger, Software/Systems Inc., USA
Roberto Meli, Data Processing Organisation, Italy
Peter Mellor, CSR, United Kingdom
Ingunn Myrtveit, The Norwegian School of Management, Norway
Domenico Natale, SOGEI, Italy
Martin Shepperd, Bournemouth University, United Kingdom
Rini van Solingen, CMG, The Netherlands
Erik Stensrud, Stensrud Consulting, Norway
Richard D. Stutzke, SAIC, USA
Jos Trienekens, Eindhoven University, The Netherlands
June Verner, Drexel University, USA
Isabella Wieczorek, Fraunhofer (IESE), Germany
Claes Wohlin, Blekinge Institute of Technology, Sweden

Conference Office
Adrian Cowderoy, ESCOM Conference Ltd
30 Willow Tree Glade, Calcot, Reading
RG31 7AZ, UK.
Phone: +44(UK) 118 9 427 970
Fax: +44(UK) 118 9 542 591
Email: office@escom.co.uk
URL: http://www.escom.co.uk/

ESCOM 2001 SPONSORS
CSR, The Centre for Software Reliability, UK
ESCOM 2001 PARTNERS
The conference is held in association with SCOPE.
The conference is co-located with the IEEE 7th International Symposium on Software Metrics

TABLE OF CONTENTS

TRACK A: FIRST NEWS

SCOPE TRACK

6. Conclusions

If measurement and process are mutually dependent they should be planned, stored and reused together. Measurement programmes require cross-project structures for data storage, analysis, threshold tuning and validation. The TAME model provides the mechanism for achieving these goals.

Would the author recommend every software development company to drop their existing organisational structure and adopt TAME? In reality, no. No one would seriously consider swapping their organisational structure in a revolutionary way. So you might say, what's the point of the work? The TAME model provides an example of an organisational structure with facilities to support measurement and for development to be supported by measurement. It isn't necessary that the TAME model be adopted. However, it is desirable is that those in software development gain an understanding of the inter-dependencies of measurement and process, and understand the facilities that need to be put in place in order to take advantage of them.

7. References

[1] Basili, V.R. and Rombach, H.D., "The TAME project: Towards Improvement-oriented software environments", IEEE Transactions on Software Engineering, 1988. 14(6), p. 758-771.

[2] Lott, C.M. and Rombach, H.D., "Measurement-based guidance of software projects using project plans. Information and software technology", 1993. 35(6/7), p. 407-419.

[3] Basili, V.R. and Rombach, H.D., "Support for comprehensive reuse", report No. CS-TR-2606, 1991, University of Maryland.

[4] DeMarco, T., "Controlling software projects", Yourdon Press, New York, 1982.

[5] Shepperd, M., "Foundations of software measurement", Prentice Hall, 1995.

[6] Curtis, B., Kellner,M.I. and Over, J., "Process modeling", Communications of the ACM, 1992. 35(9), p. 75-90.

[7] Sutton, S.M., Heimbigner, D. and Osterweil, L.J., "Language constructs for managing change in process-centered environments", ACM SIGSOFT Sofware Engineering Notes, 1990. 15(6), p. 206-217.

[8] Sutton, S.M., et al. "Programming a software requirements specification process", in 1st IEEE International Conference on the Software Process, Redondo Beach CA, 1991.

[9] Osterweil, L.J., "Software Processes are Software Too", in 9th International Software Engineering Conference, IEEE Computer Society Press, 1987.

[10] Shepperd, M.J. and Ince, D.C., "Derivation and validation of software metrics", Open University Press, 1993.

[11] Avotins, J. "Towards an object-oriented metric modeling method", in OOPSLA'96 - workshop: OO product metrics, 1996.

[12] Guindon, R. and Curtis, B., "Control of Cognitive Processes during Design : What tools would support software designers?", in Proceedings of CHI'88, New York: ACM, 1988.

[13] Basili, V., Caldiera, G. and Rombach, H.D., "The goal question metric approach", in Encyclopedia of software engineering, Wiley, 1994.

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 well 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).

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 ($A_1, A_2 \ldots$):

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

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

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

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:

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

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:

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

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^t_{ij} = resources devoted the *k-th* day to the *ij* subproject
- A_{tK} = available resources of the type t in the *k-th* day

It is important to remember the following definition of Eij (Eq. 6): $Eij = m \ / \sum_{k=1}^{m} \sum_{t=1}^{F} e^t_{ijk} = \sum_{t=1}^{F} Tijt$. 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.

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.

SouthernSCOPE – A Method for Acquiring Custom Built Software or Let the Customer Manage Software Acquisition

Anthony L Rollo, Terry Wright

Abstract

The current size of the global industry developing custom built software is estimated to be in excess of $US200bn annually. The performance of the software industry in developing successful projects is poor with an estimated 32% of projects terminating before delivering anything to business and only 11% completing within budget. Of the remainder that do complete, the average budget overrun is 87% [2]. A Recent survey of the International Software Benchmarking Group data repository has shown that the costs paid by business for comparable software can vary 7 fold [3]. This litany of software failure makes the development of techniques to allow business to retain control of software acquisition compelling and ever more urgent.

SouthernSCOPE is a framework strategy for software acquisition, which is designed to allow project business sponsors the ability to control and manage acquisition projects. The method allows the business sponsor to retain control of project scope and expenditure in terms that business can understand. The risk of budget overrun is considerably reduced as the sponsor is always in control of the scope; this ensures good value for money. The potential for bidding low and then loading the price of changes is largely eliminated.

The State Government of Victoria, Australia introduced the initial ideas in 1996 and has since developed several projects refining the method in the light of experience. This has resulted in the development and introduction of the SouthernSCOPE method as the standard purchasing policy of the state. This paper will explore the underlying ideas behind the method and the experience of the State government of Victoria in developing this unique approach to software acquisition. The paper will also review a case study of a software acquisition project.

1. Introduction

The State Government of Victoria Australia has an annual software budget of around $50 Million (AUD), (29 Million Euro). The state government is a leader in the use of IT in the provision of services to its citizens. The UK government has based a large part of its e-government initiative on ideas originated by the state government. Yet despite a good record of achievement in the field of government on-line the state government was faced with the problem of acquiring software within budgets and of suitable quality. These are problems that any organisation faces when acquiring software. The difficulty is well categorised by a recent report. An estimated 32% of projects terminate before delivering anything to business and only 11% complete within budget. For projects that complete the average budget overspend is reported by the CHAOS report [2] as being some 87%. This report also stated that the situation was no better than the previous survey some years earlier. "There is no evidence that budget performance of software projects has changed"[2]. These figures are depressingly familiar to IT managers and stand as an indictment of the global software