

Software Maintenance: Analysis of estimates based on requirement specification in the context of New Technologies, a Case Study

Juan Carlos Granja-Alvarez
Granada University, C/ D. Saucedo A. s/n . E18071- Granada (Spain) ,
Tel: +34 958242365, EMail: jcgranja@ugr.es

1. Introduction

At the outset of a project, and taking as the start point the specification of requirements, the estimation of certain variables (effort, time, etc.) comprises a crucial element, not only for the development group but also concerning the image offered to the client by the software administration group. Therefore, accuracy in estimating such variables is highly relevant to customer satisfaction and, consequently, the continuity of the project.

To address this problem and to provide the software process with an engineering approach, various authors have proposed models to perform such estimation processes, such as the COCOMO model and function points. Both of these have undergone considerable modifications to adapt them to the evolving nature of software development. However, one of the main problems still affecting them is their application to new forms of development.

Making estimations in areas for which there is no previous experience is a difficult task, and even more so if its performance requires a historical data base (which does not yet exist). The problems involved in making estimates of developments for which very limited historical data are available (as is the case of new technologies) were taken into account in the present study of estimation inaccuracy. This study focuses on the crucial question of the specification of requirements, with regard to their application in the field of new technologies. Our estimation is based on the metrics of function points, which enable us to carry out a study without requiring prior information and which is perfectly compatible with the field of the measuring point, as mentioned in the new version of the COCOMO model.

2. Use cases

Use cases describe high-level interactions between the system to be developed and the elements that interact with it. Use cases are narrative descriptions of domain processes concerning the sequence of events of an actor (external entity) that uses a system to complete a process that occurs in the domain and is characterized by interaction.

3. Empirical results

With respect to the stage of requirements analysis, we measured the function points against the use cases achieved by each of the groups. The functionality of these use cases varied between 4 and 5; doubtless, each comprised a different degree of complexity and effort:

Table 1

Funcionality	Group	Lines of code	Mean Complexity
F1	Group 1	378	1
F2	Group 1	450	2.1
F3	Group 1	280	2
F4	Group 1	350	2.2
F1	Group 2	411	2.5
F2	Group 2	326	2.8
F3	Group 2	285	3
F4	Group 2	502	2.1
F5	Group 2	263	2.1
F1	Group 3	329	1
F2	Group 3	680	2
F3	Group 3	478	2
F4	Group 3	195	2.4

7. Conclusions

One of the conclusions to be derived from the present study is that very good estimates may be obtained from use cases for software maintenance. Undoubtedly, the natural internal language of use cases brings with it a high degree of subjectivity, but nevertheless this can be partially overcome by combining their use with graphic resources such as use case diagrams. The combined use of these two elements enables us to deduce the factors necessary to make an estimate, i.e. in the present case, to deduce the function points.

Analysis of the function points measures functionality from the client's viewpoint, that is, on the basis that the user asks for information and receives replies; this notion is perfectly compatible with the principle of use cases.

For estimates of function points to achieve a more accurate fit with reality, and taking into account that the estimation of function points refers to the use or otherwise of a given development methodology, we believe that due consideration of the latter would lead to more realistic results, as we have shown that the level of compliance and the use of a given methodology may have a considerable influence on the final results achieved. Furthermore, the consideration of the methodology in the function points should be carried out such that the methodology in question influences both the function types and the general characteristics of the application. The way in which software development is carried out is not a new approach in the context of function points.