

Software costs: a pragmatic approach

G. Legall, M.F. Adam, M. Lepallec & B. Moreau

*Département Evaluation et Validation des
Protocoles, Centre National d'Etudes des
Télécommunications, LAA/EIA, 22300 Lannion,
France*

*Direction des Relations Industrielles, Service de
l'Industrie, ECEF, Immeuble Pérésud 7 Bd
Romain Rolland, 92128 Montrouge Cedex, France*

INTRODUCTION

Since the beginning of the 1980s CNET Lannion has carried out studies on the quality of software (Adam [1], Le Gall [2]). Clients and suppliers now agree that quality should be seen as a basic requirement which serves to support and extend the activities geared towards increasing productivity and competitiveness. It follows then that quality control of a piece of software must not only be specified, planned and then applied but also assessed from a financial point of view in order to judge its profitability.

Quality must be specified, which means that we need terms of reference by which we can determine, in a quantitative and measurable way, the objectives which are to be achieved: this is the main aim of these studies on software quality which are designed to determine :

- . the relevant measurements to be used in assessing the quality of software ;
- . the methods, tools and structures which provide the best results (reliability estimation, project management and tracking, quality control, ...) ;
- . the techniques which enable us to assess very early in a project the risks of non-quality ;
- . the costs associated with global implementation of a quality assurance process and the conditions of application (human and technical resources).

This process is, of course, a dynamic one. In order to define quality and the methods by which it can be measured, each new element must be taken into account in carrying out the following projects. For this reason, these terms of reference must be included in software development contracts and, consequently, their application must be assessed from a financial point of view (investment and return on investment). The study of costs therefore falls within this context.

OBJECTIVE AND STARTING POINT

The main objective of the study in the long term is to research how to control the development of FRANCE TELECOM software, whatever the origin: in-house development, off the shelf purchases, development through contract study... This involves the gathering and storage of information on the quality of the end product, the development time and costs, the cost and efficiency of the controls which are implemented, the quality of the manufacturing process (tools, methods, manpower...).

654 Software Quality Management

The initial stage of this study involved the implementation of an information gathering process on development time and costs of software projects as well as on the description of associated procedures, not only in the CNET but also in France Telecom industry department, which is attached to the Direction des Relations avec les Industriels (DRI/SI) [Industrial Relations Direction]. This information gathering process was made possible by devising a questionnaire (Le Pallec [3]) which feeds a database on the distribution of costs and time associated with control and production activities corresponding to the software life cycle. The data gathered enable us to analyse correlations between information on production processes and information on product quality (Le Gall [4], Derriennic [5]). This is also used for expert systems which are currently being tried out at the CNET :

- . ELYS (Lysys [7]) : which estimates development time and costs for a software project with a breakdown of tasks per stage ;
- . ESQUAL (Lysys [6]) : which assesses the risk of non-quality of a software project before each stage of development of a piece of software.

Of course, it must be said that before a reliable result can be obtained, several dozen software projects must be observed so as to provide a sufficient amount of data for us to start using the data analysis techniques and looking for correlations between all the information gathered.

PRELIMINARY STUDIES

The study has taken three separate directions which all lead to a single objective in the long term, which is the control all the processes implemented as part of software development activities. This control comprises :

- . an ability to estimate the size and complexity of items of software in order to be able to assess the costs of these projects better ;
- . a possibility of introducing clauses in the contracts drawn up to support the tracking, control and assessment activities of the processes and the products, and which are based on specifications which have been validated through experimentation and are accepted by industry ;
- . preserving know-how as the party which places orders, in order to progressively improve this process.

This is why the following actions have so far been carried out simultaneously :

- . assisting in the process of selecting a supplier when there is a call for tenders or else assisting with negotiations on development prices and costs with a single supplier for a given project. This relatively new procedure has not been developed in this presentation as it needs to be confirmed by the results of several experiments ;
- . adding a questionnaire to the contracts which is designed to gather information on software development time and costs and on how these are broken down, as well as a description of the procedures used for development for the purposes of building up a database on this subject ;
- . within the context of a pilot project, to find out into how much detail we can go in setting up and monitoring a quality assurance process, which includes overall research on costs; the purpose of this action being to validate the specifications of CNET regarding the setting up and control of a quality assurance process and to avoid this upsetting the projects (surplus costs for example).

These two experiments are described in the following pages.

Distribution of the questionnaire

This feasibility study was carried out simultaneously at the CNET on study contracts leading to the development of software and at the DRI with the department entrusted with studies and financial control of supply contracts. There is a clear difference between these two contexts:

- . at the CNET, the questionnaires concentrate on a description of the procedures used to develop the software and are introduced in the contracts thus becoming contractual in nature. Experience has clearly shown that this respect for contracts is nevertheless at the discretion of the technical staff responsible for the contracts who insist more or less on this clause completely depending on how much they have been made aware of its importance ;
- . at the DRI, until 1991, the method was to request replies from suppliers who were known to the department and on projects which had already been completed. The interest of suppliers lay in determining the position of their project compared to those of the others suppliers, although they knew that this reply did not take into account the confidential information on other projects (names of companies and projects ...).

Since 1991, the methods used by the DRI and the CNET have been identical and are aimed at using the questionnaire as a contractual document.

The execution of this part of the study has been difficult and time-consuming for the following reasons :

- . response time (awaiting the end of the project to obtain information on it or taking the decision to reply to the questionnaire submitted by DRI) ;
- . choice of target (sometimes an operation is planned by the CNET but is never implemented because it is of low priority, but the time spent in preparing and negotiating is still the same).

The questionnaire includes questions about :

- . the nature of the software (new project, modified, field of application,...) ;
- . the overall size of the software (according to 3 parameters : re-use, modification, creation) ;
- . the methods and tools used ;
- . the effort and the time dedicated to a development (according to production and control activities, according to the stages of the life cycle and according to the processes (management, development, support, ...)) ;
- . the degree of automation of tasks ;
- . the limitations of the project (performance, complexity, quality, ...) ;
- . the needs (quality of specifications, conditions of use,...) ;
- . the development teams (experience in similar projects, the used languages, ...) ;
- . the quality assurance plan for the project (ISO 9000, PAQ certification, ...).

Gross results This paper only includes the results which relate to costs and time. It is worth pointing out that out the 12 questionnaires which have been completed, only 10 were sufficiently complete or reliable to be taken into account and supplied in Table 1 (productivity) and 6 for Table 2 (division of costs, ...).

Apart from project 1, overall productivity in these projects seems to be divided between two categories :45 to 55 code lines, per day and per person for one project category (4 out of 10) and from 77 to 88 for a second category (3 out of 10). Equally, the productivity recorded during the development stages seems to show this division into two parts for productivities of between 100 and 150 code lines, per day and per person. This estimate will need to be confirmed subsequently according to the categories of language used, of aspects relating to the automatic generation of codes from high level representations (code or test generation from formal specifications, ...),



656 Software Quality Management

Project number	1	2	3	4	5	6	7	8	10	11
Total output in Kilo Lines	64	917	217	172	76	252	31	499	9,3	84
Total effort (man x month)	10	830	216	98	110	164	25	500	6	96
Overall Productivity (lines per day)	320	55	50	88	36	77	62	50	78	44
Development effort	6,5	500	111	57	80	110	11	/	5	/
Development productivity	492	92	98	151	48	115	141	/	93	/

Table 1 : productivity overall and of development (requirement + design + coding

Currently, the most important thing is to show that the relations of trust between clients and suppliers allow the required knowledge base to be set up. This offers interesting possibilities, such as comparing the productivity of development processes in a given area.

Project	number	1	2	3	5	6	7
Total Number	Com	16	404	54,9	30	112,4	11,9
of Lines in Kilo	Dec + Sta	48	513	161,5	46	140	20,4
Lines	TOTAL	64	917	216,4	76	252,4	31,3
Effort in Phases (in man/month)	Req +Pre D.	1,5	260	32	40	59	4
	DD.+Cod	5	240	79	40	51	7
	UT+Int	2,5	320	94	26	37	10
	Validation	1	10	4	4	17	4
	TOTAL EFFORT	10	830	209	110	164	25
Duration of Phases in Months	Req +Pre D.	2,5	8	7	31	27	4
	DD.+Cod	5,5	5	5	30	13	3
	UT+Int	4	7	9	32	8	2
	Validation	1	4	1	3	4	4,5

Table 2 : example of size distribution, effort and time of the project

Com : Comments ;

Dec + Sta : declaration + Statements ;

Req + Pre D. : Requirements + Preliminary Design ;

DD. + Cod : Detailed Design + Coding ;

UT + Int : Unit Testing + Integration.

These results can be presented in the form of a graph as in figure 1 below. What this figure shows is that, as regards the major concern over decreasing software development time for the new services, it is difficult to reduce delays by acting solely upon the acceptance phase.

On the other hand, maximizing code production by means of unit tests using automatic generation techniques may allow this result to be achieved. However, it may be possible to gain extra time during the stages of external specification, call for tenders, negotiation and drawing up of contracts before the software life cycle stages begin (development aspect).

The limited number of projects studied allows little more than a trend to be detected. We will have to await a more representative sample before the results can be considered truly meaningful.

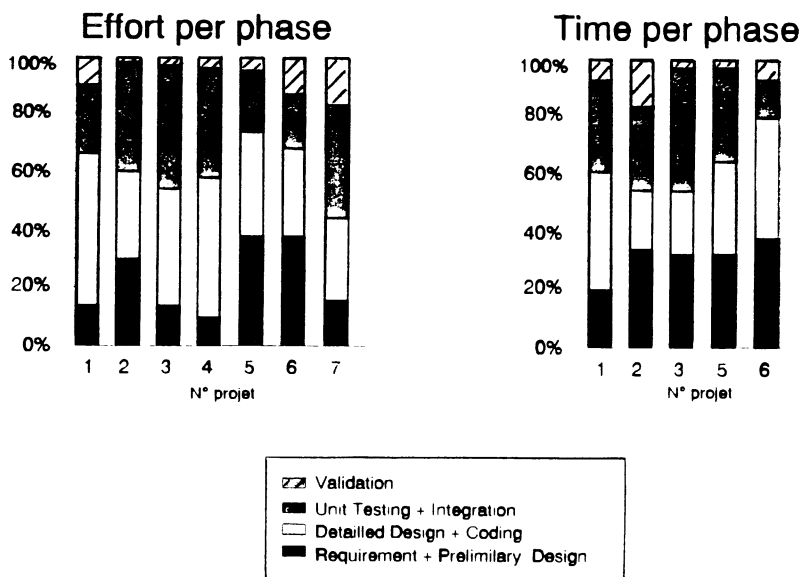


Figure 1 : Effort and time repartition per phase

Difficulties encountered It is still too early to carry out any statistical analysis of the sample we currently have (12 projects). Not only is the number of projects very small but it contains projects of different kinds and different sizes developed in different contexts (study contracts or industrial products,...) In the long term the projects will need to be analysed in clearly defined categories. The work carried out by the ISO/IEC - JTC1/SC7 in this field will be valuable in determining these software categories.

Another difficulty lies in the replies given to the questionnaires: the productivity of a large project is assessed globally without differentiating between the various sub-projects of which it is made up, and only in relation to the total number of lines created and total effort involved. This number can be defined in different ways, which makes interpretation difficult (work is due to begin in this field at ISO/IEC - JTC1/SC7 in order to standardize this assessment by applying the FPA (Function Point Analysis) method.

There are other difficulties which are due to the way in which the questions themselves are worded: sometimes the questions put are too subjective and not sufficiently factual, which makes it difficult for suppliers to approach the question with a good degree of confidence and to provide reliable and full replies.

Furthermore, a supplier's reply is influenced by his management environment (regardless of the recommendations made in the questionnaire) and will vary in accuracy: for example effort is sometimes given in MxM and other times in MxD, days sometimes being working days (20d/month) and sometimes calendar days (30d/month) and may or may not include days off. Sometimes there is no distinction between the validation and acceptance phases.

It has also been noted that the number of questionnaires which can be used for data analysis decreased in proportion to the number of data chosen for this type of study. This problem is directly linked to the completeness or reliability of the replies.

Interest The fact that a supplier is able to reply to this questionnaire shows that his project management is serious. Furthermore, the information requested may, at the start of a project, lead the person in question to ask himself certain questions on their project and to assess the different options which are available to him. These may also highlight risks of non-quality for the project, not only due to the supplier but also due to the client. The questionnaire therefore can be a means for the supplier to detect these points and to modify the framework of the project to reduce the risks of both parties.

It would appear that the suppliers themselves would be interested in it once the number of projects recorded allowed objective comparisons between projects to be made. They are therefore keen to know what their position is compared to that of other similar projects in terms of respect for deadlines, effort, quality, productivity,

The systematic distribution of this questionnaire allows a CNET specific database to be fed with information (and protected of course by cryptographic methods,...). It can be interrogated remotely by any authorized person (in order to draw up statistics for example). As for the CNET, the recording of this information allows it to :

- . carry out studies aimed at categorizing software projects according to objective criteria ;
- . associate life cycle models to these software categories ;
- . carry out research on the patterns of distribution of costs and time, according to the stages of the life cycle patterns associated with the software categories ;
- . carry out analyses of the correlation between the processes used during the life cycle of the software and the quality of the final product (expressed in terms of metrics or failures),....

It should also be noted that the existence of the database has been declared to the CNIL (Commission Nationale Informatique et Liberté) and that a favourable response has been received.

Pilot project

The aim of this other study was to specify and monitor the quality of all the aspects of a software development process (Le Gall [8]) for the purpose of :

- . assessing the impact of a metrology which has been specified at the outset by the client on the quality of the software, on order to verify the feasibility of it and then to monitor its implementation ;
- . experimenting with the gathering of information on anomalies during the entire project in order to study the relationship between anomalies and metrics ;

. obtaining all the information relating to project management in order to compare the costs and production times with those which are dedicated to technical and quality control, for documents as well as source software, the final objective being to assess the monitoring not only from a technical point of view (effectiveness) but also a financial one (feasibility).

Context The pilot project involves the development in language C of the software of protocol card D for an ISDN traffic simulator. The following recommendations have been inserted in the contract :

- . obligation to create this software within the framework of a quality assurance plan, in accordance with current standards (ISO 9000 or AFNOR Z 67 130) and approved by the CNET ;
- . inclusion in this quality assurance plan of the CNET's recommendations :
 - to fill in the questionnaire which relates to the description of the production process implemented ;
 - to supply all the information arising from the project management (cost and time elements, both for production and monitoring) ;
 - to include within the quality assurance plan the CNET recommendations relating to quality control and measurement during development ;
 - to supply all the quality control reports relating to all the stages of the life cycle ;
 - to supply fault reports for all the stages of the life cycle, as defined by the CNET, according to the item to which they relate (document, procedure).

Cost analysis This In the Table 3 :

- . Quality Control includes cross-readings of documents and code, corrections made after these readings, quality control carried out by the quality engineer ;
- . QCSC (quality clauses specified by the CNET) includes the changes made to the software as recommended by the CNET, the non-regression tests, the production of fault and operation reports ;
- . the external validation is made by CAP SESA REGIONS client.

Phases	Production	Quality Control	Quality Control/ Production	QCSC Application	QCSC Application/ Production
Documentation	206	25,5	12%	10	5%
Coding - Unit Testing	327	20	6%	30	9%
Integration Validation External Validation	130	9	7%	20	15%
Miscellaneous (C Language adaptation)	20	5	25%	3	15%
Total	683	59,5	9%	63	9%

Table 3 : quality control and production costs of the different phases (M/D)



660 Software Quality Management

Functions	GE	AI	NW	DL	PH	Total
Nb of modules	8	3	8	9	11	39
Nb of procédures	45	10	292	215	98	660
Nb of lines	11327	2634	53191	25530	22055	114737
Production Cost (MxD)	17,5	8	138	81,5	82	327
Quality Control Cost (MxD)	2	1	5	5	7	20
QCSC Application Cost (MxD)	4	2	10	8	6	30

Table 4 : function size - development and control costs

The cost of quality control is 17% for the specification and documentation writing stage, 15% for the creation stage (coding-unit test) and 18% overall for the life cycle in total (Table 3).

If we look at the way this investment is distributed, at the creation stage, between the five functions which make up the software (Table 3 and 4), we can see that this cost is not proportional to the size of the software. The quality control costs of the NW function, which is the most important in terms of the size of the software produced, were no greater than those of the DL and PH functions (figure 2).

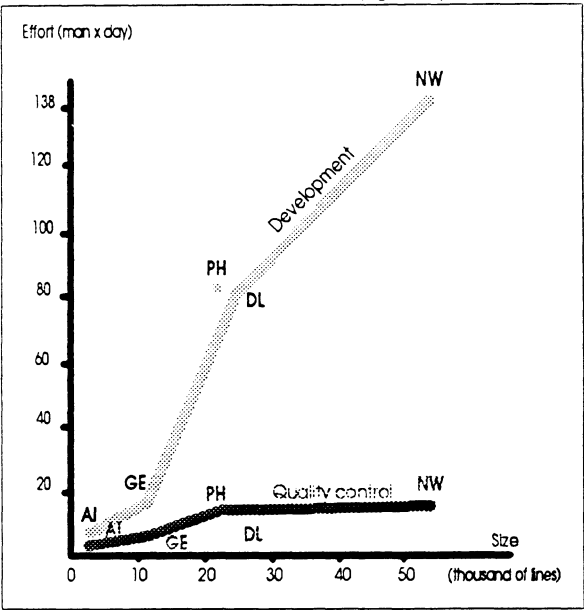


Figure 2 : Development and quality control efforts vs Software sizes

In Table T5 we can see a decrease in the number of faults detected from one stage to another and conversely, the growth in the average analysis, correction and validation cost. For one fault detected during testing, the 0.05 days spent at the coding stage drops to 0.2 days at the integration stage and then rises to 1.12 days at the costing stage, which gives a ratio of 1 to 20. The total time spent on analysis also shows a net increase (from 3.64 days on coding-unit tests to 11.26 days). This is why it is important to dedicate time to re-reading during the preceding stages, in order to reduce the overall costs due to failures. During the 5 months of the external validation phase, 19 failures were detected. Among these, only 10 related to the software and the others were errors in the documentation (2) or errors of implementation by the client.

Phases		Coding - Unit Testing	Intégration - validation	External Validation
Number of detected faults	Re-reading	122	9	/
	Testing	74	49	10
Cumulative time spent in analysis (in days)	Re-reading	4,54	0,94	0
	Testing	3,64	9,8	11,26
Cumulative time spent doing corrections (in days)	Re-reading	5,07	0,5	/
	Testing	2,24	6,62	6,02
Cumulative time spent on validation (in days)	Re-reading	5,67	0,38	/
	Testing	2,65	4,31	2,12
average time spent in analysis (in days)	Re-reading	0,04	0,1	/
	Testing	0,05	0,2	1,12
average time spent doing corrections (in days)	Re-reading	0,04	0,05	/
	Testing	0,03	0,13	0,6
average time spent on validation (in days)	Re-reading	0,05	0,04	/
	Testing	0,04	0,09	/

Table 5 : evolution of analysis, correction and modification costs due to faults

During the external validation, the time spent on validation is not significant because validation has usually been left to the client.

662 Software Quality Management

Assesment of the pilot project in terms of costs This study has demonstrated the validity of assessing the costs and effectiveness of quality control procedures, of which the investment, taking all information gathering procedures together, will have represented 18% of the total cost of this pilot project. On the one hand, this result will need to be confirmed in other projects, on the other hand complementary studies will have to be carried out in order to estimate the return of this quality investment during the operational stage.

Similarly, it will be interesting to continue the study on the evolution of the costs of faults according to the stage at which these are detected. However, it is reasonable to believe that this investment would only have accounted for 7 - 10% had certain information gathering procedures not been planned within the context of this project, which would have been more acceptable than the 18% figure arrived at.

CURRENT DIRECTION

Currently, the replies to the questionnaires and the assessments supplied by the tools are only one element of the process of selecting the supplier who will be entrusted with developing the software. However, the experiments carried out so far lead us to make the following considerations :

- . the estimates supplied by ELYS on the time and costs corresponding to the planned developments make it possible for us to know whether the proposals made by the suppliers are or not in line with the forecast curves of this tool ;
- . sometimes considerable differences are noted and the questionnaire which is completed on the methods, tools, development teams ... allows the sources of these differences to be determined. This then allows us to direct well-targeted requests for further information at the supplier ;
- . when it comes to making the decision, it is neither the expert system nor the systematic consideration of the best price proposal which determine the choice, if the replies given to the questionnaire raise doubt as to the supplier's capacity to keep to his promises.

To conclude, all these actions have demonstrated the feasibility of this kind of cost control and software development time verification procedure. Industrialists can reply to the questionnaires without unveiling confidential information on their know-how and within the context of a quality assurance process which is increasingly recommended. Furthermore, if they so wish, all the information they provide leads to follow-up information being provided by the client (CNET or F.T.) on the position of their project in relation to those which have already been registered, which can lead them to review certain parts of their procedures, methods,

The questionnaire which has been used so far is currently being revised with the aim of correcting the shortcomings of the first version. Every subjective question is systematically being replaced by a factual one. This work is being carried out within the framework of a cooperation with the CSELT on the study and implementation of a common procedure between CNET and CSELT on the quality assurance of software, using the skills of both research companies. The result of this action is to modify the tools which use the questionnaire as an entry point. It is the reason why a pre-study of the new functions of ELYS is being carried out simultaneously, in order to associate with it an function which will observe software costs.

The questionnaire and the associated tools (ELYS and ESQUAL) will be used at the very beginning of the software's life cycle and in a systematic way. During the call for tenders stage, to have a common base on which to compare the replies given by suppliers to a demand expressed by F.T., at the final stage to assess different factors relating to this software project (productivity, effectiveness of the software development process, ...).

CONCLUSION

The process presented in this document is only the result of a feasibility study. However it does demonstrate how sensible it is for a company to gather in one place all the information which relates to the development of software. This then allows the company to control future developments more effectively, because it has a mass of information which will help it in negotiations with its suppliers, both in terms of planning and software costs and in terms of assessing the development process set up by the suppliers.

The implementation of this process allows one to formalize the procedures which lead to the selection of suppliers, it being a basis on which to identify the factors of software cost and to determine the way in which these costs are distributed, as well as to gather the information needed to know or act with the aim of increasing productivity of software projects, and optimizing development time.

BIBLIOGRAPHY

- 1 Adam, M.F., Le Gall, G., Moreau, B., Valette, N. 'Towards an Observatory Aiming at Controlling the Software Quality' *2nd IEE/BCS Conference*, Liverpool, England, 11-15 Juillet 1988
- 2 Le Gall, G., Adam, M.F., Derriennic, H., Moreau, B., Valette, N. 'Studies on Measuring Software' *IEEE, Jsac Telecommunications Software Quality and Productivity*, volume 8 numéro 2, pp 234-246, February 1990
- 3 Le Gall, G., Le Pallec, M. 'Description du Logiciel et de ses Procédures de Développement' Reference FT.DRI/SI/ECEF/3/327/MLP
- 4 Le Gall, G., Derriennic, H. 'An Experiment in Using Conventional Software Reliability Growth Models in Telecommunications' *12th International Symposium on Reliability and Maintainability* Tokyo, Japan, 4-8 June 1990
- 5 Derriennic, H., Le Gall, G. 'Utilisation des modèles à intensité de défaillance dans le suivi de la validation de logiciels du domaine des Télécommunications' *8ième Colloque de Fiabilité et Maintainabilité*, Grenoble, France 6-8 October 1992
- 6 Lysys company 'ELYS manuel utilisateur' Paris, France, 1988
- 7 Lysys company 'ESQUAL manuel utilisateur' Paris, France, 1988
- 8 Le Gall, G., Adam, M.F., Derriennic, H., Lassudrie, C., Lucas, J.P., Valette, N. 'Contrôle de la Qualité lors d'un Développement de Logiciel' *Third EOQC*, Madrid, Spain, 3-6 November 1992