Software Quality Assurance and Maintenance for Outsourced Software Development

Nelly Maneva

Institute of Mathematics and Informatics, BAS, 1113 Sofia, Bulgaria
Email: neman@math.bas.bg
and
American University in Bulgaria, 2700 Blagoevgrad, Bulgaria
Email: nmaneva@aubg.bg

Abstract. The paper describes the author's experience, gained during a complex and long outsourcing project. The applied Software Quality Program is presented and some ideas for its improvement are given. The problems of project management, software configuration management and maintenance are clarified. A feasible approach to these problems – both incremental and measurable – has been suggested. In conclusion the author shares a few possible directions for further research.

1 Introduction

Nowadays the outsourcing is a common practice especially for the software companies in the South-Western part of the Europe. *Outsourcing* means that software engineering activities are contracted to a third party who does the work of lower cost (definitely) and higher quality (hopefully). Due to the shortage of software people with appropriate qualification, skills and experience, or lack of other facilities (hardware, development environment, communications, etc.) the contractor looks for a vendor to solve these problems. Outsourcing can be *partial* – when only a few parts of the software system are contracted, or *complete*, when the whole software system under development is contracted. According to another classification the outsourcing can be planned or ad-hoc. The *planned* outsourcing is a part of company strategic business plan to conquer new markets, to enter the market very fast or to create an attractive image, not having the resources needed. The *ad-hoc* outsourcing can be an attempt to deal with a project behind schedule or to solve unexpected software problems.

From the other side the benefits for the vendor are to obtain a real and pure software work keeping his/her people busy and comparatively well paid. This can be a way to avoid some software company pre-project activities (market research, tendering, bidding) and a number of post-project activities (software product promotion, distribution, copyright protection, etc.). All these activities require specific, more business-oriented knowledge and skills, which are not common for software people.

Theoretically, the outsourcing can be considered as a mutually beneficial deal. Unfortunately, the real-life story is a bit different and we will try to explain why.

2 A Case Study - our Outsourcing Project

2.1 The Project

In the early 90's a number of consecutive software development projects were carried out in Bulgaria, ordered by the software development department of a big German corporation. All projects were in the field of courseware development and other types of dedicated software. The last project was the biggest one [1]. The purpose was to develop a software tool for the analysis, planning and design of computer-based training programs. As a member of the development team I would like to share our experience gained during this complex and long outsourcing project. Further in this paper we will call the representatives of the German company, the initiator of the project, with a general name *customer*, and the Bulgarian team – *developer*.

Bearing in mind the well known from the literature software engineering recommendations, both sides made an attempt to define explicitly the initial conditions, constraints and risk factors and to outline the basic principles of project management and quality assurance program.

The following constraints were imposed on the Bulgarian developers:

- the development cost and the project deadline had to be determined in advance and could not be reconsidered;
- the general characteristics of the hardware platform and the software tools for the project had to be determined by the customer, but the selection and purchase had to be made by the developer;
- the developer had to articulate and submit for approval the project management scheme and the quality assurance plan;
- during the development the customer had to be responsible for the requirements definition, external specifications and the independent testing, and the developer had to be responsible for the detailed design, programming, internal testing and preparation of the maintenance documentation.

At the beginning of the project some risk factors have been identified and analyzed, namely:

- the requirements definition was incomplete and fuzzy, not allowing the precise estimation of the size and complexity of the desired software system;
- the distant development was to some extent new for all participants and the lack of experience required finding out the appropriate solutions by testing several alternatives;
- in this distant development the daily communications by e-mail should be in a language (English) foreign to both sides.

The SQA program has been created on the following three assumptions:

- the quality of the software product depends on the quality of the people involved in the project and their proper management;
- the quality of the software product depends on the quality of the development process;
- the quality of the software product in a remote development project depends heavily on the quality of the communications system.

The Project Management program described the gradual development, the responsibilities of the teams and a detailed schedule for all distant activities and face-to-face joint meetings.

2.2 Project Results and Their Evaluation

The work on the developed tool had several continuations and finally ended (due to some financial reasons) a year ago. Now from the distance of the time passed we can evaluate the project and to summarize the results of this experimental outsourced software development.

We can report a successful Software Quality Assurance (SQA) program. The defined set of three assumptions really managed to ensure an efficient SQA policy. Next we will remind each assumption and will describe our brief comment for it.

Assumption 1. The quality of the software product depends on the quality of the people involved in the project and their proper management.

The size and the structure of the teams, the requirements for education, qualification and experience of the developers, the scope of their work and responsibility have been planned carefully and followed strictly. The high professional level gave us the possibility to solve all arising problems (both algorithmic and technical). The developers were motivated not only by their salaries but by the challenges of the project itself. A number of research papers, describing some problems and their innovative solutions have been published during and after the project. On the other side the customer's financial stability and the high level of its organizational culture created an environment, facilitating the fruitful work on the project.

The lesson: Select the right partner – to be qualified, loyal and highly motivated.

Assumption 2. The quality of the software product depends on the quality of the development process.

A model for a gradual development has been implemented. Two consecutive prototypes have been created. Unfortunately, the preliminary defined schedule was modified several times due to very often changes of the requirements.

The lesson: It is not possible to predict the duration and cost of each project phase. So as a step-wise procedure for project scheduling should be applied: a rough schedule with definition of the basic phases at the beginning of the project and a refinement only for the next phase, specifying the expected results, the acceptance criteria and the proportional payment.

Assumption 3. The quality of the software product in a remote development project depends heavily on the quality of the communications system.

The whole organization of the communication (who, when, how often, the control and distribution rules, etc.) was described in the so called communication protocol. Those days when the Internet made its first steps we encountered many technical problems. Fortunately they are not valid any more because of the reliable and fast Internet connections now.

The lesson: A communication protocol has to be established at the beginning of the project and has to be strictly followed.

3 A Revised Approach to Outsourced Software Development

The experience from a set of distant projects and the additional research work give us the possibility to propose a number of improvements to the overall development style in software outsourcing.

3.1 Our Approach to SE Activities

In [4] we describe our general approach to the software engineering activities, performed in a software company. Here we will present its basic principles only to make possible for the reader to understand how they can be applied to outsourced development. The basic characteristics of our approach are the following:

a) the approach is an incremental. At the beginning it can be applied only for one software engineering activity – e.g. Software Quality Assurance. After the evaluation at the end of the period for experimental use, the approach can be expanded to cover other activities, too.

b) the approach is based on the idea to keep the control over the whole project all the time. But it is well known that we cannot control what we cannot measure. So we propose to use the universal technique of comparative analysis [3] and a set of supporting tools so as to measure the progress in any observed activity;

c) the approach is adjustable to the specific features and style of work in a given software company. For each applied technique at least two different levels of complexity should be defined. After the corresponding cost/benefits analysis the senior managers of the company can decide which activities will be covered and to what extend. In our opinion this makes the approach more pragmatic and feasible.

The introduction of this approach to a given software company comprises the following steps:

- defining the main goal to be achieved (e.g. an efficient development process, high quality software products, stable and attractive market image, etc.)
- selecting a few basic SE activities to start with;
- for any activity - developing a systematic methodology and a plan for its implementation;
- starting a pilot project to examine the validity of the concrete approach version;
- evaluating of the results and making a decision how to proceed.

3.2 A Program for an Efficient Outsourcing Development

We would like to apply our approach to the outsourcing development. Following the above mentioned steps, first we have to define the main goal of the program. The goal is to keep the control over the people, product, process and project. This goal leads to the initial set of three activities – two with already recognized importance - Project Management and Quality Assurance - and a new one, underestimate till now - the Software Configuration Management. We decide to include it because the analysis of project results shows the need of special efforts for change management.

Activity	Method/Technique	Comment
Pre-contract and Contract Activities		
Select the software parts to be outsourced	Cost/benefits analysis	
Choose the right developer	Comparative analysis	Criteria: qualification, experience in shipped products, standard and stable development environment, QA practice
Create a contract and assess it	Formal review through checklists	To achieve the SQA goals state a number of milestones with defined deliverables, acceptance procedures and payments
Management		
Personnel management	People management capability maturity model	A number of key practice areas for software people should be identified
Product management	Comparative analysis to measure the progress	Criteria: size, complexity, functionality, flexibility, etc.
Process management	Incremental development through consecutive prototypes	Well specified local and umbrella activities
Project management	A rough schedule for the whole project and a detailed schedule only for the current and the next stage	Project tracking, risk analysis and mitigation
Software Quality Assurance		
Testing	Static and dynamic methods	Unit testing at the developer’s site and complete and repeatable system test at customer’s environment
Quality evaluation	Quality evaluation models and methods	Software ranking for a set of quality characteristics [3]
Quality control	Reviews, software metrics, comparative analysis	Create a SQA plan and follow it
Software Configuration Management		
Change control	Regulations for changing any item, “freeze” requirements	Auditable and repeatable creation of software using a controlled project library
Configuration control	Configuration audit for each increment	Release notes - to restrict the dependency on the developer

Table 1.

After the final acceptance test of the software system the initial set of activities should be expanded with the maintenance activity.

In order to develop a systematic program for each activity we have used the existing programs from the experimental project, enriching them according to the lessons learned and some ideas, described in [2] and other research papers. The outline of the basic activities content is given in the Table 1.

3.3 Maintenance of the Outsourced Software Products

The problems of maintenance after the end of the outsourcing project are not well studied till now. In our project after the 12-months guarantee period for corrective maintenance there was a number of additional contracts for adapting and enhancement. Generally speaking, three approaches are possible:

a) maintenance is done within the developer's organization. This was our case. It seems to work for a complete outsourcing and can be even cost-efficient because no need for additional efforts to understand the programs so as to change them. The main disadvantage was that many members of the developer's team had already other primary responsibilities and couldn't devote much time to this "old" project;

b) maintenance is done within the customer's organization. It seems to work for both partial and complete outsourcing projects, but it is possible only in case there is a permanent group, working together with the developers especially to be prepared for the future maintenance;

c) maintenance is contracted to a third part. This situation is a hypothetical – who will dare (and for how much money?) to maintain software, created by others?

We believe that after a successful SQA and Project Management program accomplishment the maintenance efforts will be pretty small, but anyway we need prescriptions for them. We start an investigation on this topic and we hope that our general approach will work for maintenance, too.

4 Conclusion

The current paper summarizes the experience gained from an outsourcing project. The evaluation of the SQA and Project management programs has been made and some suggestions for their improvement has been proposed.

Our future research will be directed to standardization of the proposed procedures thus facilitating their practical use. We will try to find a small and not very sophisticated project suitable for distant development so as to examine and approbate the feasibility and the usefulness of our new approach.

References

1. Eskenazi A., N.Maneva, R Radev. Project Management and Quality Assurance in a Distant Software Development Project. *Proceedings 5th SQM Congress*, Bonn, (2000)
2. Applying Software Quality Assurance to Outsourced Software Development. http://www.testlabs.com/white_papers.html
3. Maneva M., N. Maneva. Software Product Ranking. *Proceedings 23th Spring Conference of the UBM*, (1994) 238-244.
4. Maneva N. An Approach to Software Quality Assurance (to appear).