Quality Planning and Innovation for Competitive Design of a Software Tool

Stelian BRAD
Technical University of Cluj-Napoca

Developing new software products brings into equation a lot of challenges because a software product is characterized by invisibility, flexibility and complexity per unit effort. To set up a reliable business plan in new software product development, an accurate evaluation of product market utility, of the expected quality and of the necessary budget to support appropriately the development process is required. Because software products are "invisible" during the analysis, design and implementation phases, innovative approaches are required to handle properly both product and related-budget formulation. This paper describes a novel methodology for software product innovation, planning and control, capable to manage at superior levels the development process of complex software products. The exemplification on a software application for quality cost management is afterwards presented.
Keywords: *software innovation, software development, software quality planning, quality cost management*

Introduction

A software product has three main characteristics that make it different than hardware products, as follows: invisibility, flexibility and complexity per unit effort [8]. Considering only the characteristics of invisibility and flexibility, for developing new software products a lot of challenges occur in terms of effort estimation, budget formulation and deadline projection [1]. Countless studies have shown that the success rate in new software product development is in average 10% [1], [8], [10]. A major cause for such poor results is the poor definition of the software product in the early phases of its development process [2].

A reliable business plan in new software product development requires an accurate evaluation of the budget necessary to support the development process. A baseline budget is usually elaborated for this purpose [3]. Project evolution is monitored to see departures from the plan in four dimensions: delays in meeting target dates (deadlines), shortfalls in quality, inadequate functionality and costs that exceed the targets [8].

In most of the cases, the development of large software applications (that usually require over one year or several years to work out the first release) is financed via venture capital or investment funds. For such cases, monitoring the project cost evolution over time against the initial plans is crucial to support the financing line. In practice, a relative new approach has gained in popularity for cost monitoring. It is called Earned Value Analysis (EVA) and its core philosophy is to assign a "value" to each task or work-package, as they are defined in the project plan [3], [8].

During the development process of a new software product, managers and investors are interested to know how much "value-added" is brought into the product in every relevant stage of the project. In this respect, they ask for monitoring the earned value as the project progresses. Because software products are "invisible" during their development phase, accurate effort estimation is quite difficult. A reliable approach for effort estimation is the PERT method [3], [8], but this requires a good expertise in the problem domain of the software application, as well as a very detailed project plan from the very beginning of the project. When new product is brought into equation, difficulties in effort estimation are even higher. But, to calculate the key financial indicators from the business plan (i.e. internal rate of return, return on investment, net present value and payback period) it is necessary to know how much effort is required to fulfill the project [4], [10]. If the

tasks are over-estimated, projects become less attractive. If the tasks are under-estimated, the risk of failure is very high, too. On the other side, investors need a "tangible" proof of the value each task within the project plan incorporates, because they want to see that budget is allocated respecting the rule of "value-for-money" [2].
Starting from these circumstances, this paper describes a robust approach for complex new software product planning and innovation in order to meet stakeholders' expectations. The methodology has been tested on developing a novel software application for quality cost management.

Innovation in new software product development

The first important stage in new software product development is to formulate the innovative idea that grounds the business. Countless opinions reveal that 80% of the commercial success of a new software product depends on the idea which stands to the base of that product. In this respect, specific algorithms to enhance creativity are recommended. The author has experienced with very good results the combination of two advanced tools, ARIZ and ASIT, for setting up innovative ideas of new software products [2].
Innovative software products could come up, in principle, along three main axes. The first axis is based on the rule of exploiting information technologies to increase the efficiency of some business processes. Examples in this respect are various portals for data management, applications for mobile marketing and mobile telephony, applications in the area of business intelligence, applications in the field of data security over Internet, applications in the financial sector, and applications in video-conferencing and i/e-learning, etc., where the expertise in technologies is the essential one and the expertise in problem domain comes afterwards.
The second axis of innovation is mainly based on exploiting the exceptional know-how in a certain field; know-how that is implemented within a software product for a more efficient use. In this case, the implementation technology, even very important, plays only a secondary role in the innovation scheme. Examples in this respect are software products in areas like CAD, CAM, CAE, CAQ, CAR and FEA, as well as research-oriented software, process and equipment control-oriented software, software for surveillance, any kind of expert systems and so on.
The third axis of innovation for software products is the one which harmoniously integrates and exploits as much as possible from the first two axes of innovation.
Software products belonging to the first axis usually necessitate higher efforts and risks in the design and implementation component of the innovation process and few in the research component (both fundamental and applicative). Software products from the second axis mainly involve higher efforts and risks in the research component and few risks in the implementation one. Software products from the third category involve substantial risks and efforts both in research and implementation.
The higher the project risk, the higher the benefits over the product life-cycle if the project is successfully completed. From the perspective of protecting businesses dealing with software products, those from the first category are more vulnerable; competitors can easily find out which is the core of the product and are able to come up onto the market with substitutes, copies and reproductions in relatively short time. The best protected are businesses dealing with software products from the third category, because they involve the highest know-how, both from scientific and technological perspectives.
The case study later on introduced in this paper will exemplify a way of approaching the process of idea generation for a new software product.

Multi-phase quality planning in new software product development

In order to engineer value within a new, complex software product, the intangible components of that product have to be quantifiable in a reliable way. For this purpose, an original, customer-driven methodology for quality planning of software products is proposed in figure 1. An algorithm based on multi-phase planning (AHP based ranking, S-VOCT-AFD based requirements definition and five-phase QFD based planning,) and on innovation tools (TRIZ/ARIZ), integrated with the business goals (budgets, IRR, NPV, etc.), as well as with robust monitoring tools of project value (EVA) is illustrated in figure 1.

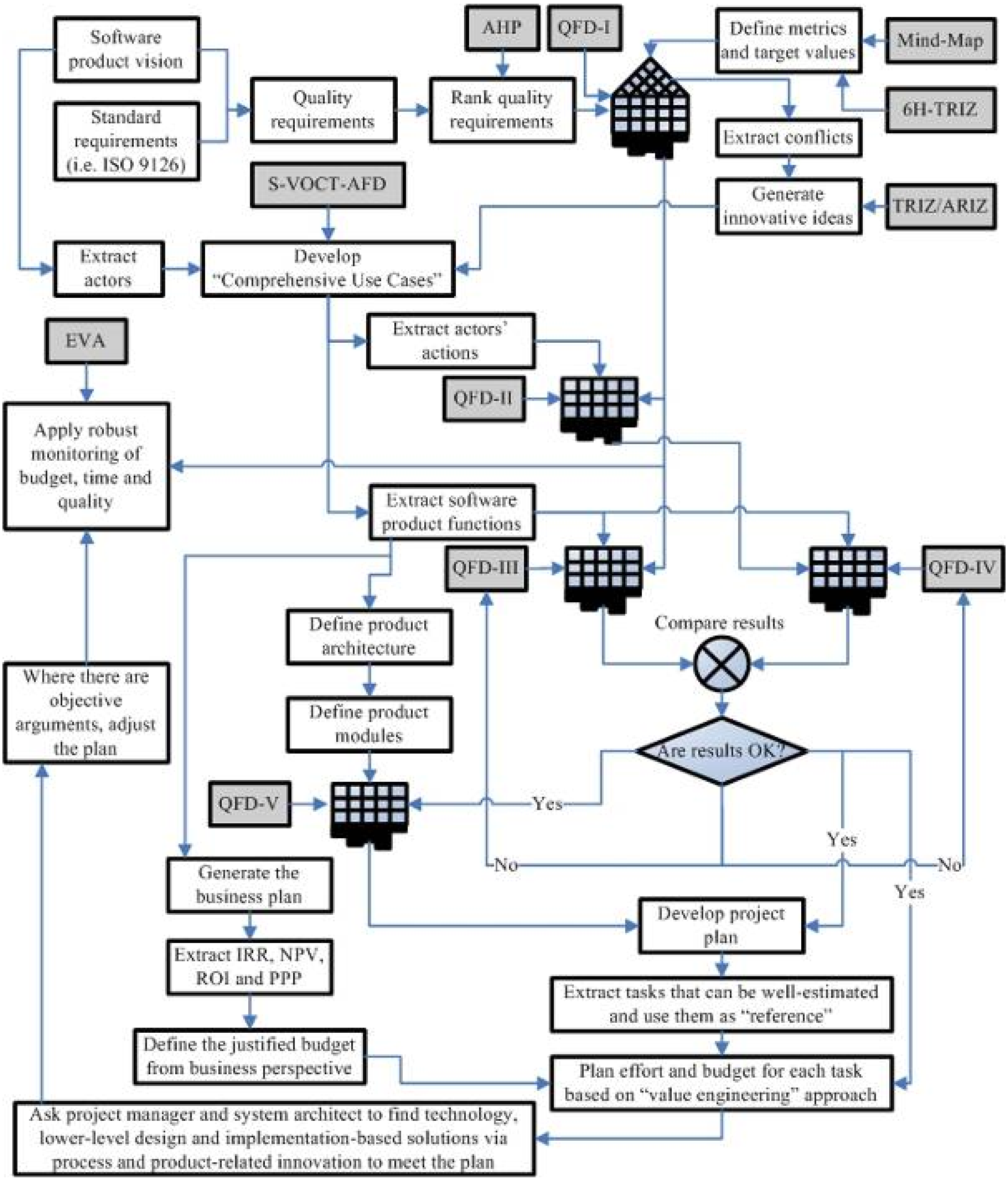

Fig.1. Roadmap for value engineering in new software products

A complex software project might take from 2500 to 8000 man-days (or even more) until the first version of the product is released. The work required to carry out the tasks mentioned in the roadmap from figure 1 can be covered by the project manager and the

system architect (working together) in maximum 2 weeks (30 man-days), which represents a very-very small percentage (0.3-1.2%) of the total project effort, but with significant positive contributions on costs and quality.
The methodology starts with the definition of quality requirements. They are extracted both from the vision document of the new software product and from the quality standards applied for software products (e.g. ISO 9126). To express these requirements in engineering language, measurable characteristics are further defined. In the specific language, they are called metrics. For each metric, a target value and a measuring approach are also considered. Metrics are afterward ranked using the QFD method [5]. On this way, the relative value weight of each metric is determined. Where conflicts between metrics are identified, innovation is required. A robust approach is the application of TRIZ/ARIZ [2].
A major step within this methodology is the elaboration of the so-called "comprehensive use cases". They are special worksheets that describe the problem domain. It is a complex document that could cover tens of pages. For each "actor" (entity that interacts with the system), one or several comprehensive use cases might be worked out. A comprehensive use case includes many sections and special-defined questions and workflows such as to cover comprehensively all aspects related to a certain action an actor is going to perform in relation with the system (the software application).
Data that are generated from this step make possible the extraction of actions and software product functions. A quality control algorithm of the planning phase for the software product functions is further considered in the methodology. If the results are satisfactory, the project plan is worked out. It includes the tasks required to implement software product functions /modules.
For investing into a new software product, a business plan is also required. According to the estimated market potential of the respective software product, some financial indicators are calculated. They are: payback period (PPP), internal rate of return (IRR), return on investment (ROI) and net present value (NPV). From the business perspective, these indicators reveal the justified budget to be allocated for the new software product development. Knowing the overall justified budget of the project, the blended man-day rate, the value weights of the product functions and the tasks required to implement them, as well as having the possibility to perform an accurate estimation of the effort for some of the tasks (2 or 3 tasks that are simpler; called "reference tasks"), the effort for the remaining tasks can be easily planned such that value to be engineered into the product. The result of the planning process automatically involves challenges for the team to identify those technologies and solutions for design and implementation that are able to meet the planned target costs for each task. In many cases, innovation is required. It might be possible that current technologies to be less reliable for meeting the targets (to bring value in all tasks). If clear arguments are brought in this respect, the project budget and the financial plan must be readjusted.
Within this framework, the monitoring of time, budget and quality can be performed at higher parameters. Having clear defined and planned metrics (see QFD-I in figure 1), test-cases for software quality evaluation can be designed in parallel with software product implementation and the testing process can be better performed (usually, it is a challenge in software development the way testing is performed to cover all essential functional aspects and to identify all "bugs" before launching the product onto the market).
Using the EVA approach [2], [8], time and budget are also kept under a tight monitoring and control. In EVA, two important indicators are periodically calculated (in most of the cases, monthly): schedule performance index (SPI = BCWP/BCWS) and cost performance index (CPI = BCWP/ACWP), where BCWP is the budgeted cost of work performed, BCWS is the budgeted cost of work scheduled and ACWP is the actual cost of work performed. A project goes well when

CPI and SPI are both greater than 1 [2], [8]. CPI and SPI are performance indicators and they are viewed as "value-for-money" indices [8]. A supplementary indicator called "critical ratio" (CR) is also calculated [3]: $CR = CPI \times SPI$. If $CR = 1$, things go well and if $CR > 1$, project is going very well. When $CR < 1$ but it is higher than 0.8, things are going bad but it is not a big danger for the project. For the case where $CR \leq 0.8$, project manager should highlight the "red flag". When $CR < 0.6$ the top management must be informed and urgent measures have to be taken.

Also, for the statistics of project financial performance, some other indicators have to be calculated [8]. They are: budget variance ($BV = ACWP - BCWS$), schedule variance measured in cost terms ($SV = BCWP - BCWS$) and cost variance ($CV = BCWP - ACWP$). CV is an indicator of the accuracy of the original cost estimates.

Case study – novel software tool for quality cost management

In the followings, a case study concerning to quality planning and innovation of a software product is introduced. The product may be positioned in the second axis of innovation, as it was shown in section 2 of this paper. The first stage brings ASIT method into equation to elaborate the vision of a new software product [2]. The step-by-step application of ASIT method in this case is further presented.

The subject: To identify a novel idea of software product in the field of business processes.

The product: To define a desired, useful and utilizable product that comes in front of some social, economic and technological opportunities. Identification of product opportunities is a combination of art and science that requires a permanent and comprehensive scanning of trends in the social layer, of economic forces and of technological emergencies. Analyzing business processes from a social perspective, it is observed that one of the driving forces is quality. Analyzing the quality problems from a technological perspective, one of the trends is to plan, monitor and control various aspects of quality. Analysing the problem of planning, monitoring and controlling quality-related problems within business processes from an economic perspective, the trend is to reduce internal losses, to quantify the costs of poor quality, to reduce the transaction costs (coordination costs), to optimize the resources allocated for various internal projects, etc. In this context, the idea is to develop a software product for planning, monitoring and control of quality-related costs at all levels and within all components of a business system.

The universe: The top management wishes to know in every moment the performances of all processes expressed in monetary units in order to make appropriate interventions.

ASIT principle: From the five ASIT principles, unification has been selected [2].

The generic result: Within the quality cost management model an intelligent agent will be integrated for generating automatic recommendations depending on the performances at a given moment.

The new product: An expert system for advanced management of quality costs and for optimizing the effort of continuous improvement of organizational processes to the level of the overall business system.

The new created value: Implementation of such expert system within an organization brings the following benefits: (a) ensures a link between business performances and quality performances; (b) provides a perspective in monetary units of the efforts involved in quality improvement; (c) provides effective solutions to identify, prioritize and select the zones where improvements should be done; (d) helps to take appropriate decisions, based on data and facts; (e) reduces the fire-fighting problems of the top management; (f) increases the responsibility of employees on medium term; (g) increase the market value of the business on medium term, by quantifying less tangible assets.

In order to put this product idea into practice, some major, know-how-related challenges should be over-passed: (a) to elaborate an exhaustive quality cost structure, with a standard database, able to cover all business processes and able to be customized to any kind of business system; (b) to elaborate the mathematics behind the intelligent agent in order to support the automatic generation of recommendations in a robust way even if variations in the structure of the quality cost system occur (where the user is able to add, freeze, delete cost items, etc.); (c) to elaborate an algorithm for assessing the overall performances of the business with respect to the total quality costs. The step-by-step application of ARIZ method for identifying viable paths in formulating the intelligent agent is further presented [2].

Problem definition: Identify a simple and reliable solution for the algorithm of intelligent agent.

Mini-problem: The algorithm must quantify the relative impact that each monitored activity has within the system, must know the influence of an activity upon the other activities, must establish the relative difficulty to improve the performance of each activity and must correlate all these elements even if within the system new activities are added or some existent activities are deleted or frozen.

System conflict: In order to solve the problem of quantifying intangible dimensions, complex mathematical formulation might be in place; they require long time for research and high costs. Also, to increase the capacity of customization, the risk of introducing noise factors in the system in raised up. This affects the robustness of the intelligent agent.

Problem model: To solve properly the first problem, some elements of the expert system should be responsible for quantification. To solve properly the second problem, some elements of the expert system must filter and refine any new information introduced in the system during the customization process.

Conflict domain analysis and resources: The conflict domain refers to the quantification process of activities with respect to various aspects (impact, influence, difficulty, etc.). The available resource is the team involved in designing the algorithm, as well as various methods of quantification, already known.

Final ideal result: The algorithm to be accessible to any expert in quality problems such as to build it without difficulty.

Contradiction: The goal is to build a robust algorithm to various noise factors but easy to apply in the same time. Also, the intent is to have an algorithm that can quantify intangible dimensions very fast, without major efforts and long time for researches. In terms of TRIZ parameters [2], the first conflict is translated as: try to improve system robustness to various external shocks without increasing the complexity of methodology. The second conflict is translated in TRIZ language as: high precision to measure system's performances using low quantity of substance.

Elimination of contradiction: Using specific tables of TRIZ method (see [2]), the following inventive principles are recommended to solve the first conflict without compromises: 1.1 extraction – extract only the necessary part or property from the system; 1.2 inversion – instead of doing the action dictated by specifications, do the opposite one; 1.3 replace a rigid system – put a flexible solution instead a rigid one. For the second conflict, TRIZ method leads to the following inventive principles: 2.1 extraction – as for the first conflict; 2.2 universality – make the system for multiple purposes; 2.3 color changing – change the transparency of the system using additives to make visible some hidden parts.

Solution: Extraction leads to the idea of quantifying activities against the key, unanimously accepted criteria of business performance. Inversion leads to the idea of using a set of integrated methods instead of a single method for quantification, as well as to the idea of keeping rigid the top layer of the process structure (which is very generic) such as to ensure through it the control of the flexible cost structure. An integrated set of methods (IAM-QFD-CAST I) was used to design the algorithm [2]. Color changing leads to the idea of integrating fuzzy rules to quantify linguistic variables.

The methodology presented in figure 1 has been used to plan the development of the software product. The project had to be performed in 16 months and was financed based on an initial estimation of 1320 man-days effort. For this application, 42 requirements have been extracted and ranked using the AHP method. In the QFD-I stage (see figure 2), a set of 43 metrics has been weighted.

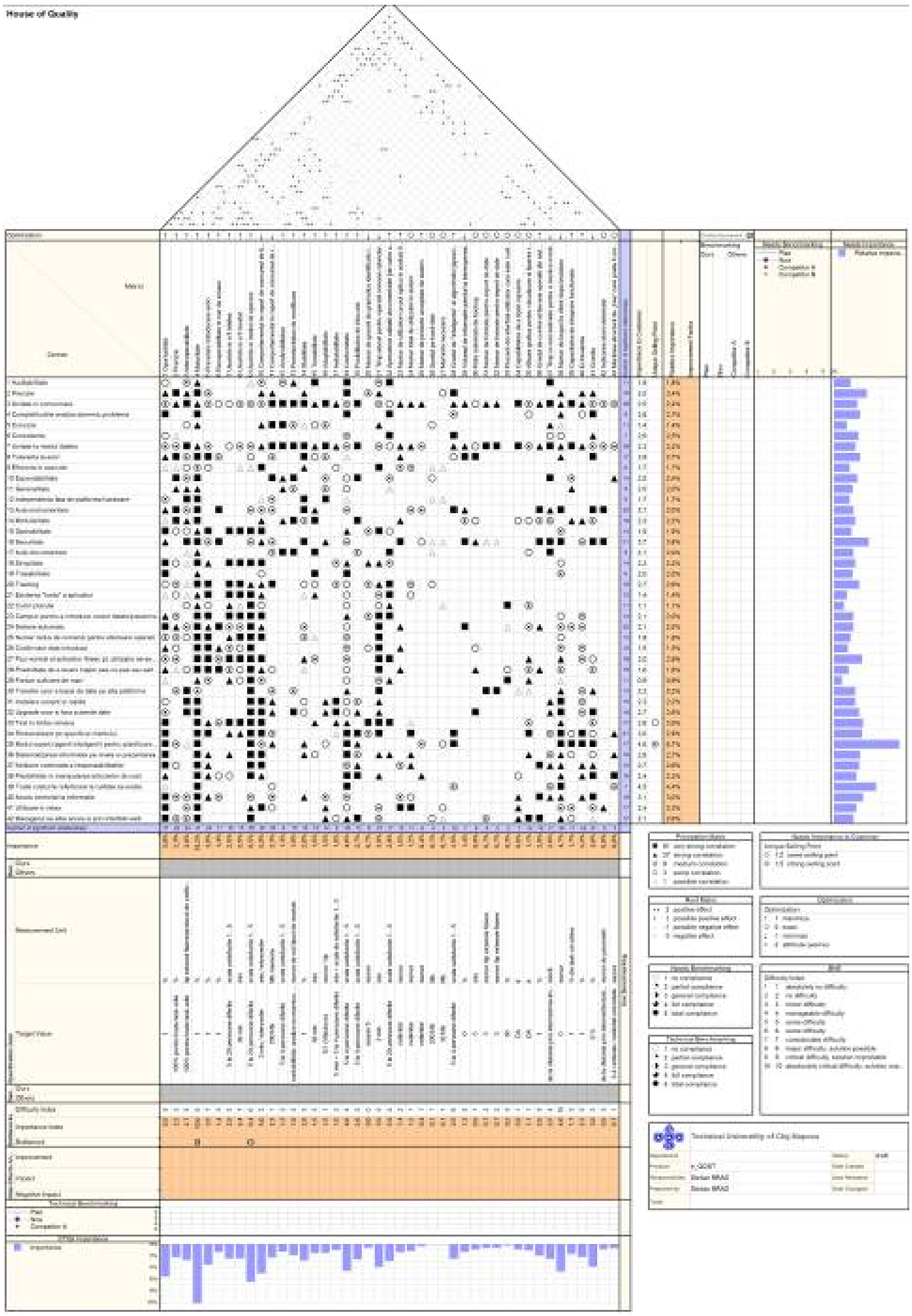

Fig.2. Software quality planning: the QFD-I phase

The software application consisted of 66 key functions. These functions have been weighted and adjusted in the QFD-III and QFD-IV stages. The core set of aggregated actions of the actors were counted at 14 and weighted in the QFD-II stage. For the application of QFD method, Qualica QFD™ software package was used [5]. Also, over 20 conflicts between metrics have been identified in the planning phase (QFD-I). To identify innovative solutions to these conflicts, the TRIZ method was considered. It was effectively applied via Innovator Workbench[TM] software package. The whole work of quality planning was performed within 35 man-days (2.6% of the total estimated effort). Based on this information, project planning against an economic justified budget for development was properly elaborated. Technical team involved in software design and development had to adopt innovative solutions and appropriate architecture and technologies to meet the time and financial schedules.

Concerning to “design for easy customization”, solutions emerge from several directions. The first direction emerges from the concept of defining and structuring quality-related costs. Mapping the quality cost system over the structure of the business system represents a reliable way of increasing flexibility and customizability of a quality cost management system. In this respect, the well-known EFQM model is taken as reference [7]. This means, the quality cost structure shows like a tree-structure, with quality cost items distributed on affinity groups. The tree-structure starts with a set of 9 blocks, as in the case of EFQM model: (1) leadership (LDP); (2) strategy-policy-marketing (SPM); (3) personnel management (PMT); (4) resource management (RMT); (5) core processes (CPS); (6) employee satisfaction (ESN); (7) customer satisfaction (CSN); (8) society satisfaction (SSN); (9) business performances (BPS). Each block includes a set of standard main processes (MP). Within each main process, a set of standard activity-modules (AM) are further formulated. To the level of each standard activity-module, a set of standard value-added elementary activities (AM) are defined.

Quality costs are collected to the level of elementary activities. Each cost item is assigned to one of the four categories of quality costs: prevention, appraisal, internal failure or external failure. Researches led to a standard tree-structure of quality costs consisting of: 92 standard main processes, 227 standard activity-modules and 512 elementary activities. To have a better image around the size of the standard database, a listing of elementary activities covers over 100 A4 pages. A generous reservoir of information and resource of education is stored within this standard database, because it represents a compilation of huge amounts of data about business models and quality cost systems from various structured and unstructured sources. The space of this paper limits the provision of more details around this issue.

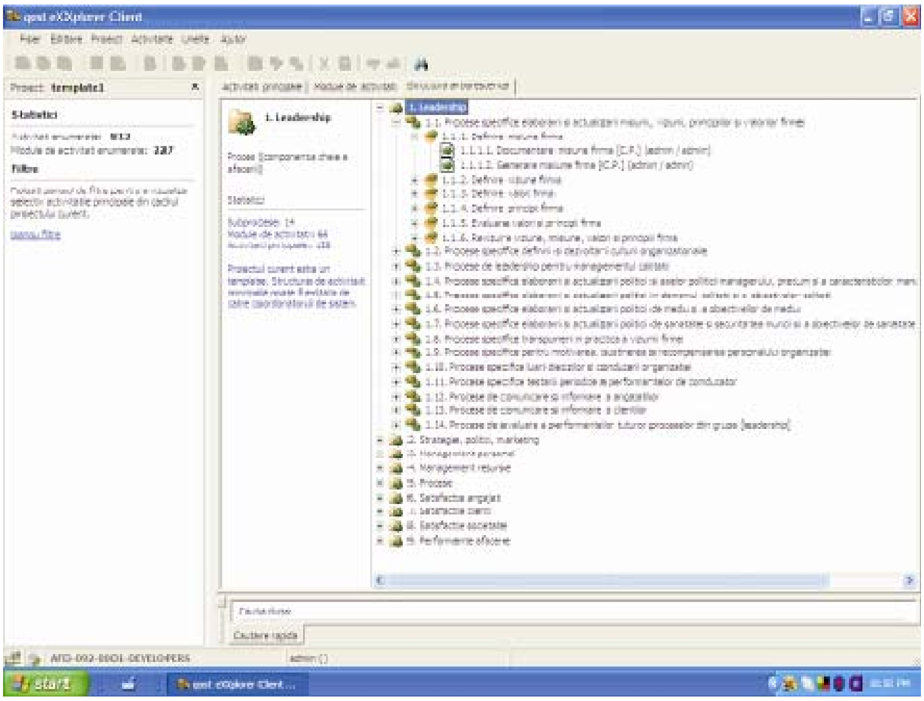

Fig.3. Fragment of the business process oriented tree-structure of the quality cost system

The standard database of quality costs follows a natural way of connecting the quality cost system to the business system and provides the essential information for customizing the quality cost system [6], [9]. A company has the possibility to adapt, enhance, reduce and adjust the standard database to its specific needs in a friendly manner and from any point of the tree-structure, as long as the used "language" is of managers, not of accountants. In other words, one can add, delete, freeze/unfreeze activity-modules, elementary activities and quality cost items. Figure 3 illustrates a fragment of the standard database tree-structure, as it was implemented in the software application (the highlights in figure 3 are on the leadership block, which includes 14 standard main processes, where the first main process includes 6 standard activity-modules and where the first activity-module includes 2 standard elementary activities).

The menu of the software application is written in Romanian because of its target market. The concept of the standard database, as well as the quantity and the content of information which the database incorporates, represent, all together, the key issue for fast and facile design of a customizable quality cost management for any given business system, independently of its maturity, profile and specificity in time and space.

The second direction to increase customizability and flexibility occurs once the quality cost management system is implemented within a software tool that incorporates features for distributing data collection to any person in the network. Figure 4 exemplifies the way this was achieved in this case study.

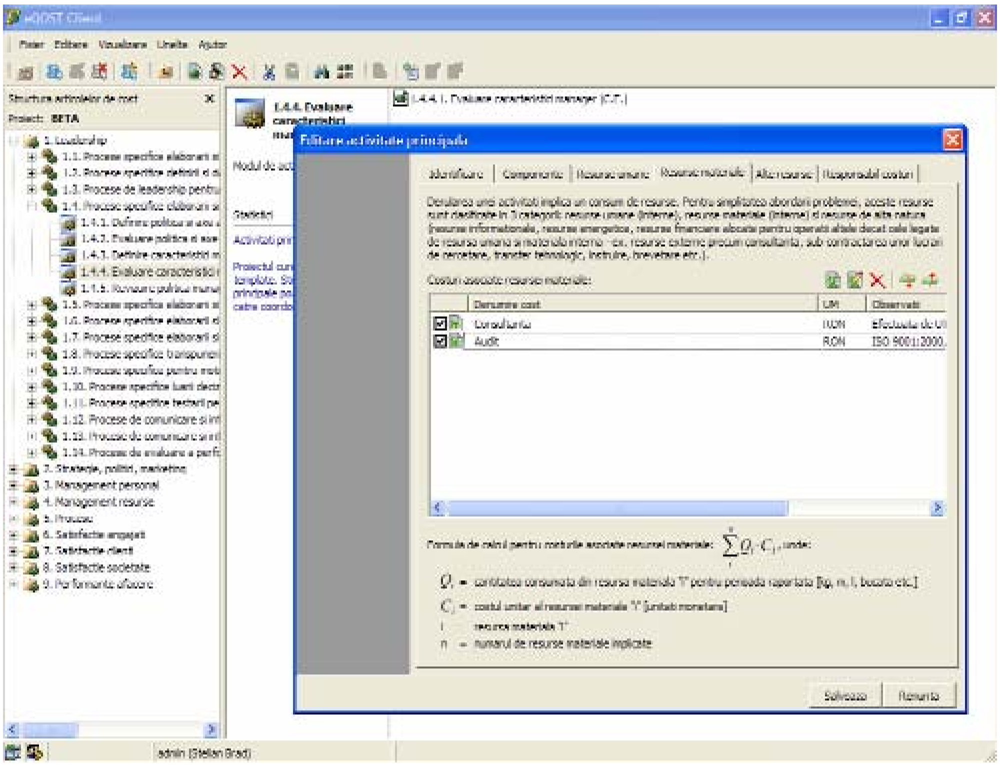

Fig.4. Use interface for quality cost definition and data introduction

The third direction materializes from the solution which defines the quality cost items. If quality cost items are defined in a specialized language (understandable only by accountants) the flexibility of the quality cost management system is very much affected. In this respect, quality cost items were defined in a natural language, easy understandable by any person in the company (from white collars to blue collars) [6], [9].

Figure 5 illustrates screenshots of the reports displayed by the intelligent agent. The front screenshot shows the report in .pdf format, the middle screenshot reveals the first set of priorities and the backside screenshot highlights supplementary areas of interventions.

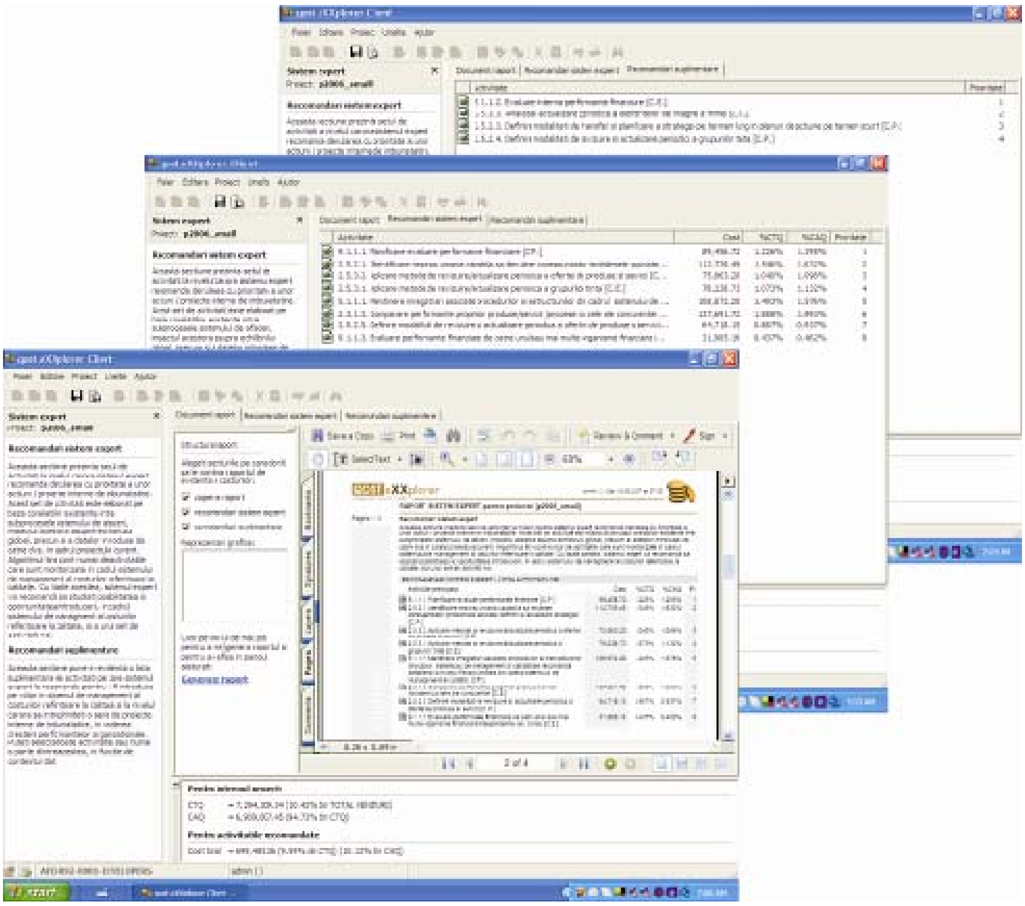

Fig.5. Screenshots with top priority activities automatically generated by the expert module

The following top level features characterize the software tool to meet some other challenges: 1) possibility to be used in distributed business structures via Intranet; 2) controlled access to data and information; 3) comprehensive and detailed consideration of all items related to quality costs (to cover all processes, as recommended by the EFQM business excellence model); 4) flexibility in handling the quality cost items (insertion, completion, deletion, adaptation, etc.); 5) controlled allocation of responsibilities for supplying the system with data and information (who, what, how, when, etc.); 6) systematization of the collected information on several levels of detailing and facile presentation for data analysis; 7) integration of the intelligent agent to generate automatic recommendations for continuous improvement of quality performances; 8) detailed help to make the system accessible to any person (for data introduction); 9) customizable to business specificity; 10) friendly and ergonomic graphic user interface (GUI).

The client-server type architecture was adopted, allowing several users in the organization to work with the application in the same time and from different locations. A user accesses the application through a personal account; each user has specific rights, depending on his or her role in planning, monitoring or analyzing the quality costs.

Completing each project's data (values corresponding to costs/resources) is done by several users, as specified by the system administrator. A project can be analyzed by the top management and by analysts. For each project, several reports can be obtained, such as:

(a) The technical report: it shows the progress of cost monitoring process. The completion level of the project at a given time is shown (global or for each involved user), to-

gether with information regarding to how milestone goals were reached;
(b) Cost accounting: total/partial numeric values/sums for different types of costs or resources, goal status (what was obtained vs. what was planned), various graphic representations (e.g. balance of certain resource or cost types) and interpretation for the obtained values;
(c) The expert module: it automatically generates, based on scientific methods, a list with activities to act upon, for reducing poor-quality costs.
Screenshots with four types of reports, selected from a multitude of report-types (defined via special designed filters), are shown in figure 6 (tree-structure representation, pie-charts, bar-charts and global data in .pdf format).

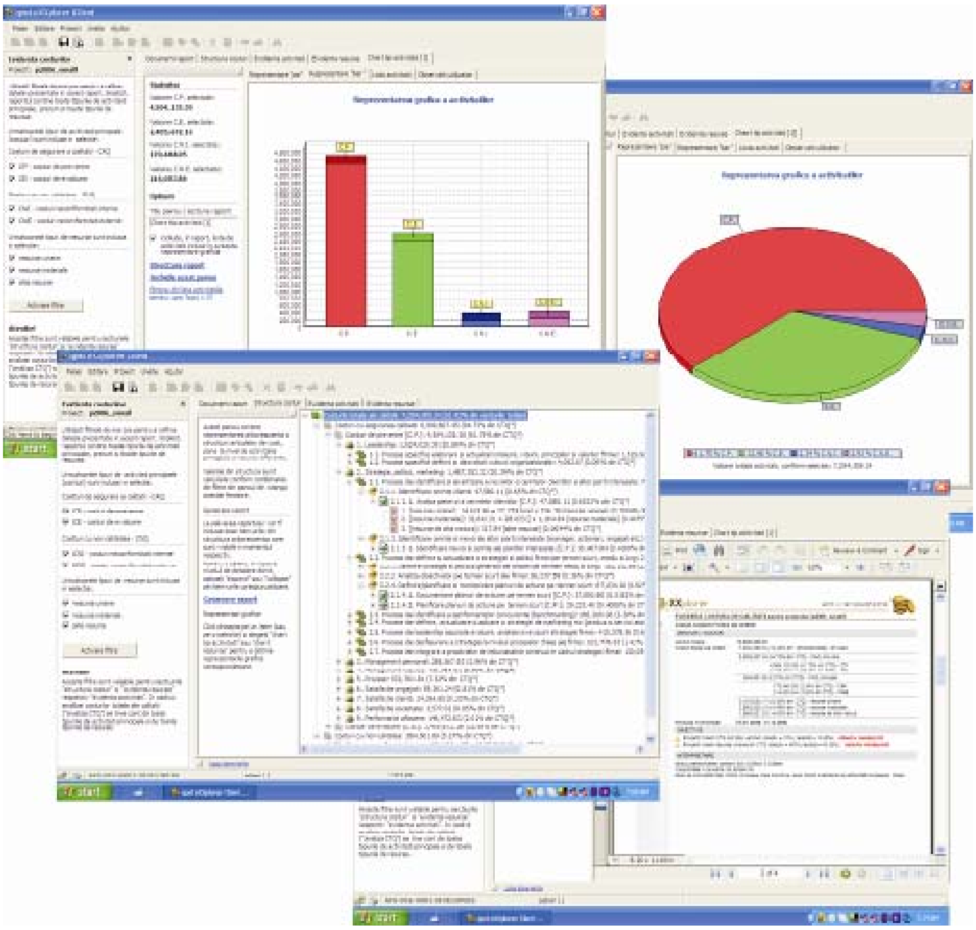

Fig. 6 Screenshots exemplifying type of reports

Other characteristics of the software application are: detailed help; customization to any business type; designed to involve every employee (if necessary) in cost monitoring, not only the accounting department; built-in messaging system, through which users are automatically notified about new projects and about project milestones (or they can just communicate with each other); automatically generated log, containing all operations performed in the system, so that the system administrator can easily track-down problems.

Conclusions

A novel methodology for innovation and quality planning in the case of new and complex software applications is introduced in this paper. It takes the idea of concurrently integration of market dimensions, technological dimensions and business dimensions in the design and decision-making process with a strong support of advanced tools of inventive problem solving. The methodology has been successfully applied for defining a new software product – an expert system for quality cost management. Final results have been validated in real conditions – a large enterprise from the chemical sector.

Acknowledgements
Financial support from the Romanian Ministry of Education and Research, within the grant CEEX/INOVEX 140, is acknowledged with gratitude.

References

1. Brad, S., Crisan, L.: *Software Quality Planning in IT Outsourcing Companies Using the 4-Phase QFD Method*. microCAD International Conference, Miskolc, Hungary, ISBN 963 661 620 5, pp. 15-20, 2004.
2. Brad, S., Ciupan, C., Pop, L., Mocan, B., Fulea, M.: *Manualul de Bază al Managerului de Produs în Ingineria şi Managementul Inovaţiei*, Ed. Economică, ISBN 978-973-709-265-6 / 973-709-265-1, 700 pp., 2006 (in Romanian).
3. Burke, R.: *Project Management. Planning and Control*, John Wiley & Sons, ISBN 0-741-57525-8, New York, 1998.
4. Cagan, J., Vogel, C.: *Creating Breakthrough Products*, Pearson Prentice Hall, ISBN 0-12-969694-6, International Edition, 2002.
5. Cohen, L.: *Quality Function Deployment. How to Make QFD Work for You*. Addison Wesley Longman, Massachusetts, 1995.
6. Freiesleben J. *On the Limited Value of Cost of Quality Models*. Total Quality Management. **15**(7), pp. 959-969, 2004.
7. Nabitz, U.: *Improving the EFQM Model: An Empirical Study on Model Development and Theory Building Using Concept Mapping*. Total Quality Management, **12**(1), pp. 69-81, 2001.
8. Roth, P.: *Software Project Management*, McGraw-Hill, ISBN 007 709505 7, Cambridge, 1999.
9. Schiffauerova, A., Thomson, V.: *A Review of Research on Cost of Quality Models and Best Practices*. International Journal of Quality & Reliability Management, **23**(6), pp. 647-669, DOI: 10.1108/02656710610672470, 2006.
10. Trott, P.: *Innovation Management and New Product Development*, Pearson Prentice Hall, ISBN 0-273-68643-7, International Edition, 2004.