A Framework for Measuring Business Processes based on GQM

Lerina Aversano, Thierry Bodhuin, Gerardo Canfora and Maria Tortorella
RCOST - Research Center On Software Technology
Department of Engineering, University of Sannio
Via Traiano, Palazzo ex-Poste – 82100, Benevento, Italy
(aversano/bodhuin/canfora/tortorella@unisannio.it)

Abstract

The evolution of business processes and supporting software systems requires their analysis and assessment from both quantitative and qualitative points of view. The analysis and evaluation activities need the support of methodological and technological tools, customizable to the innovation requirements of the chosen processes and supporting software systems.

This paper proposes a measurement framework based on the Goal-Question-Metric (GQM) paradigm. It is generally applicable to any business process and supporting software system after its instantiation. The collaborative software environment WebEv, Web for the Evaluation, is proposed for facilitating the collection and elaboration of the required measures. Finally, the paper describes the application of the measurement framework in a real context.

1. Introduction

Technological innovation and introduction of the new services are the major drivers for Business Process Reengineering (BPR) approaches, aiming to the re-thinking and re-designing of the enterprise core business processes [10, 8]. BPR activities imply the analysis of the current processes of an enterprise and their assessment from both quantitative and qualitative points of view. This entails a continuous monitoring of both external innovation drivers and internal processes status [1]. The supporting software systems have also to be assessed in order to plan their renovation and integration into the new process view. The assessment activities aim at achieving a deep knowledge of the processes and related problems, taking into account organizational and technological issues. They require the definition of formal approaches and tools to be adopted. Their systematic usage guides the process actors to collect the needed data in a correct manner, and decrease the risk of erroneous interpretation and/or superficial description [16].

This paper proposes a measurement framework to guide the activities for assessing business processes and supporting software systems. It has been defined on the basis of the Goal-Question-Metric (GQM) paradigm [2, 3] and aims at helping process analysts to find useful indications regarding the process performance, its critical elements, the output services that can be improved, the new services required, the impact of the chosen changes, the improvement expected and the future of the supporting tools. This paper presents also the design of the improvement. It includes the activities concerning the definition, formalization and characterization of the measurement framework.

The support of a software tool is important when a big amount of data have to be collected and elaborated. In addition, each team member participating to the assessment, wherever he is and whatever the task he is performing, should access the tool. To fulfill this objective, the environment *WebEv* – Web for the Evaluation – has been implemented. It is a collaborative environment, based on Web technologies and supports all the activities regarding the management of the measurement framework and the collection and analysis of the data.

The following section discusses related work. Section 3 describes the experimentation design guiding the *e*Business process evolution. Section 4 presents in major details the measurement framework. Section 5 briefly presents the functionalities of WebEv. Section 6 describes a part of the evaluation conducted in a local Public Administration. Conclusive remarks and future work are given in section 7.

2. Related work

Several approaches have been proposed in literature aiming at supporting the activities for assessing

software systems. In this specific context the challenge of the assessment consists of collecting all quantitative data useful to find the most adequate solution for evolving the software systems in order to meet new requirements [1]. This activity has to be supported by a decisional framework that takes into account several factors including software value, risk analysis, and cost estimation. The construction of an effective measurement framework has to be based on the identification of a wide variety of metrics and on the definition of consequent approaches to collect them.

Many proposed approaches are based on the seminal paper of Verdugo [13]. It defines four feasible alternatives for managing software assets: throwing away the system and redeveloping it, reengineering, freezing, or maintaining it. Rugaber et al. [15] introduced a conceptual framework for selecting a strategy to move a COBOL system into an environment supporting distributed accesses. The framework includes a variety of options and a number of selection criteria that are expressed in terms of groups of factors related to the usage of the existing system, its structure and functionality, and expected evolution. Bennett et al. [5] proposed a decisional framework that assists organizations in making decisions about legacy systems. The model consists of exploring the purposes and objectives of the organization and producing different business scenarios for its future. Then, the business strategy is identified with respect to the organizational goals, and on the basis of the analysis of the characteristics of the existing software systems, staff, and processes.

Approaches have been proposed for supporting the assessment of organizational aspects of the enterprises. Robert Kaplan and David Norton have proposed the Balanced ScoreCard (BSC) approach in 1992 [12]. It is essentially a management tool that provides senior executives a comprehensive measure of how the organization is progressing towards its strategic goals. BSC starts with the analysis of the mission and vision of an enterprise, and then defines the financial objectives to be achieved. Based on the financial objectives, BSC defines the customer measures that are needed to produce the desired financial performance. It was initially developed in the business domain, but it has recently been adapted to the software domain. In [6], Buglione and Abram compare the BSC approach with the GQM paradigm [2, 3]. This paradigm is an analytic approach goal-oriented, measurement-based and button-up driven. Its main characteristic is that it uses quantitative evidence to identify where an improvement is needed, and after the improvement, if its initiative has been successful.

The BSC and GQM approaches offer the opportunity to implement a quantitative analysis, and this has led to misinterpret them as either interchangeable or mutually exclusive. Both have a three-tier structure and both strive to determine measures from business goals, but the key point differentiating the two is the strategy that in the BSC allows achieving a proper alignment of business and operative goals. Becker and Bostelman [4] have tried to integrate the best of BSC and GQM and have proposed a common framework where the integration of BSC and GQM allows for a more detailed focus on the organizational measurement.

By exploiting the previous experiences, this paper proposes the definition and application of a measurement framework based on the GQM for the assessment of both business processes and supporting software systems.

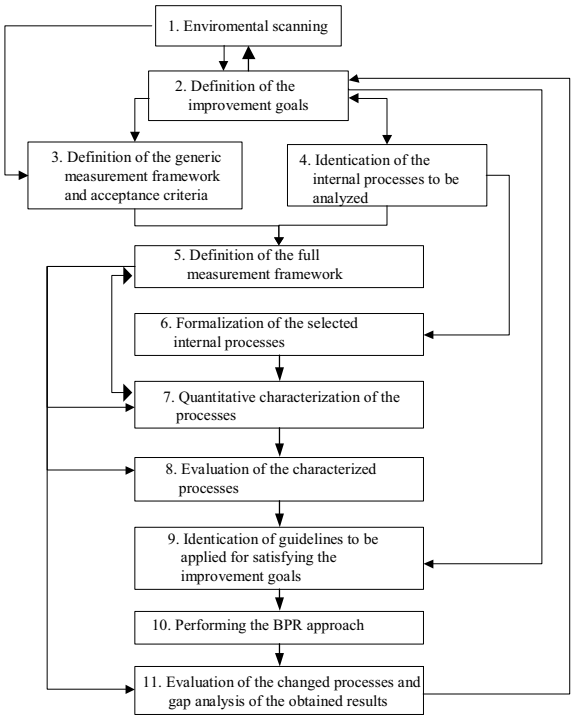

Figure 1. The Experimentation Design

3. The Improvement Design

Figure 1 describes the steps to be performed during analysis, evaluation and improvement of *e*Business processes. The figure highlights that the steps have to be applied in a disciplined manner in order to analyze the chosen processes and evolve them on the basis of the needs and obtained results. The improvement design includes also steps referred to the definition of the measurement framework. Adopting the GQM paradigm, the measurement framework is before generally defined, and it is after refined to the specific chosen context. Then, the first version is general and

refers to the assessment of a generic process without any reference to its specific domain. Its customization to a specific domain is performed when the business processes to be analyzed are selected and it is possible to consider the special needs of the data collection step. This implies the analysis of the defined specific framework in order to verify if the planned measures can be performed in the chosen business process and to identify those measures that cannot be collected. In addition, specific sections are introduced to adequate the framework to the processes to be assessed. A description of the steps in Figure 1 follows.

1. *Environmental scanning*. To entail the continuous monitoring of technologies, laws, and users' needs together with the internal business processes.
2. *Definition of the improvement goals*. To identify the goals to be achieved, considering the results of the monitoring activities and the indications of the organization's management. The step impacts on the identification of the business processes to be evolved.
3. *Definition of the generic measurement framework and acceptance criteria*. To define the parameters, metrics and acceptance criteria which generally characterize the *e*Business processes and supporting software systems. They define the initial version of the measurement framework.
4. *Identification of the internal processes to be analyzed*. To select the processes to be assessed, on the basis of the chosen improvement goals. They represent the internal components potentially impacted by the change that will be performed.
5. *Definition of the full measurement framework*. The generic measurement framework is instantiated to the selected processes to define the specific measurement framework. It can be used in the initial planning activities and in the final evaluation activities.
6. *Formalization of the selected internal processes*. To model the chosen processes for extracting all the information useful in the BPR project. The formalism used in the process model definition is not a relevant issue. The Unified Modeling Language (UML) [10] can be adopted with additional tables to store the quantitative data.
7. *Quantitative characterization of the process*. To assign measures to the metrics. This step has to be performed in a manner as objective as possible. If some measurement is not possible for the current process model, it can be refined to gather additional data. Also the supporting software systems are evaluated for identifying the components to be evolved.
8. *Evaluation of the characterized processes*. Comparing the values of the metrics to the correspondent acceptance criteria, the process and software system components that did not reach the expected results are identified for being considered in the innovation activities.
9. *Identification of guidelines to be applied for satisfying the improvement goals*. Identified the process and software components to be improved, a set of interventions are planned and defined.
10. *Performing the BPR approach*. The BPR intervention is performed in order to improve the selected processes. This can requires the execution of strategies for evolving the supporting software systems.
11. *Evaluation of the changed processes and gap analysis of the obtained results.* The measurement framework is applied again to evaluate the improved processes and software systems, and verify if the improvement goals have been reached.

This paper does not discuss in detail every phases. It aims at describing the underlying measurement framework and presenting its effective application.

4. Measurement framework

The measurement framework has been defined in terms of metrics, mechanisms of data collection, and guidelines to use the collected data. It has been defined on the basis of the GQM paradigm [2, 3] that considers a *Conceptual level*, referred to the definition of the GOALs, an *Operational level*, consisting of a set of QUESTIONs regarding the specific goals, and a *Quantitative level*, identifying a set of METRICs to be associated to each QUESTION.

As already stated, the focus of the measurement framework is the assessment of business processes and supporting software systems. The *efficiency* and *effectiveness* are considered with reference to the business processes. The process efficiency is evaluated as the ratio between the results the process activities produce – output – and the resources they require – input. The process effectiveness measures the achievement level of the process scope, in terms of users' satisfaction and adequacy to the enterprise's required standards, operative procedures, choices and awaited results respect to the obtained ones.

The assessment of a software system quality is evaluated from the user's point of view and defined on the basis of the standard ISO/IEC 9126 [9]. The cited standard considers the following six characteristics: *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, and *portability*. The first four characteristics have been included in the framework, as the user can directly perceive them. In addition, the framework considers the characteristics of the system

related to the quality in use, i.e. *effectiveness*, *productivity*, *safety*, *satisfaction*.

On the basis of these characteristics, five goals have been identified. One is referred to the efficiency. Three goals concern the characteristics of the effectiveness. The last goal refers to the supporting systems. The goals are described in Table 1. For brevity, only two goals are treated in major details, the one referring to the software systems, GOAL-5, and GOAL-1 concerning the process efficiency.

The efficiency of a business processes is evaluated by considering the cost, time and productivity. Table 2 reports the questions and metrics defined for GOAL-1. The measurements are considered for each single component involved in the process, i.e. process activities and used resources. This requires the correct execution of Step 6 of Figure 1 for understanding the process and evaluating it. The formalism used is the UML language. In particular, Use Cases, Activities and Class Diagrams are used [1]. The continuous refinement of the measurement framework when it is not possible to gather all the needed data can require new interviews to the application domain experts. Besides the questions and related metrics, Table 2 contains in the first column the code assigned to each of them, and, in second column, the identifier for each metric. The measures to be collected are mainly numeric. Some evaluations may require lists of items, singularly considered in the other measurements.

As it is convenient to collect data only if it is known how to use them in order to abstract useful information, a set of formulas are defined to combine the single gathered measures. Table 3 contains the formulas defined to identify the values related to the process efficiency. The formulas in the table are defined in terms of the metric identifiers introduced in Table 2. It can be noticed that the process cost is evaluated in terms of costs for the human resources, software systems, hardware components and other resources. On the other hand, the process productivity is defined on the basis of productivity of the single activities, which is evaluated in terms of outputs it produces, their complexity and the time needed to produce them. The time measures are not singularly considered in the process evaluation but are used in the cost estimation.

A set of variation factors regarding the activities can impact the amount and quality of the produced

Table 1. Description of the GOALs identified

GOAL-1*: Analyze a business process with the aim to evaluate its efficiency from the manager point of view.*	It is needed to identify the resources involved in the process and quantify them. For each process activity the cost, the critical level, the amount and quality of the output produced, the factors impacting the process output, may be measured.
GOAL-2*: Analyze a business process with the aim to evaluate the users' satisfaction.*	The Servqual technique [14] is used to measure the quality of services rendered by an organization along five dimensions: tangibility, reliability, responsiveness, assurance, and empathy.
GOAL-3*: Analyze a business process with the aim to evaluate the staff's satisfaction.*	It is intended to investigate on the motivations of the persons involved in the process, their satisfaction, the adequacy of the work environment, the relationships existing among the members of the staff, etc.
GOAL-4*: Analyze a business process with the aim to evaluate the process adequacy from the staff point of view.*	The aim is to verify the existence of standard rules and/or well-defined procedures to guide the application of the process activities, and the use of supporting tools. This information is used to define the expected process quality and the conformance of the actual quality to the expected one.
GOAL-5*: Analyze a software system with the aim to evaluate its quality from the point of view of the end user*	The aim is to evaluate the quality of the software system form the users' point of view. The considered characteristics are defined on the basis of the ISO quality model, and have been previously listed.

Table 2. Questions and metrics of GOAL-1

Code	Metric Identifier	Questions and Metrics
Q1		**How many activities compose the process?**
M1.1	n^A	No. of activities composing the process
Q2		**What are the resources involved in the process?** *(Question proposed for each activity of the process)*
M2.1	l_j^H	List of type of human resources of activity *j*
M2.2	n_j^H	No. of type of human resources in activity *j*
M2.3	n_{ij}^H	No. of human resources of type *i* in activity *j*
M2.4	l_j^{SW}	List of supporting software system of activity *j*
M2.5	n_j^{SW}	No. of different software system supporting activity *j*
M2.6	n_{ij}^{SW}	No. of licences of software system *i* supporting activity *j*
M2.7	l_j^{HW}	List of used hardware resources of activity *j*
M2.8	n_j^{HW}	No. of different hardware resources used in activity *j*
M2.9	n_{ij}^{HW}	No. of hardware resources *i* used in activity *j*
Q3		**What are the costs of the resources used in the process?** *(Question proposed for each activity of the process)*
M2.1	c_{ij}^H	Cost of human resources of type *i* working in activity *j*
M2.2	c_{ij}^{SW}	Cost of the software resources of type *i* supporting activity *j*
M2.3	c_{ij}^{HW}	Cost of the hardware resources of type *i* used activity *j*
Q3		**Which is the allocation time for each resource?** *(Question proposed for each activity of the process)*
M3.1	t_{ij}^H	Allocation time of the human resources of kind *i* in activity *j*
M3.2	t_{ij}^{SW}	Allocation time of the software resources of kind *i* in activity *j*
M3.3	t_{ij}^{HW}	Allocation time of the hardware resources of kind *i* in activity *j*
Q4		**What are the outputs produced?** *(Question proposed for each activity of the process)*
M4.1	l_out_j	List of output categories produced by activity *j*
M4.2	n_out_j	No. of output categories produced by activity *j*
Q5		**Which is the complexity of the output produced?**
M5.1	$compl\_out_{ij}$	Complexity of the output of category *i* produced by activity *j*
Q6		**Which is the production time of the output?** *(Question proposed for each activity of the process)*
M6.1	t_out_{ij}	Production time of the output of category *i* produced by activity *j*

Table 3. Formulas for GOAL_1

Characteristic	Name	Calculation
COST	Total cost of the process	$C = \sum_{j=1}^{n^A} (C_j^H + C_j^{SW} + C_j^{HW} + C_j^{Ot})$
	Total cost of the human resources for the activity *j*	$C_j^H = \sum_{i=1}^{n_j^H} c_{ij}^H n_{ij}^H t_{ij}^H$
	Total cost of SW resources for the *j*-activity	$C_j^{SW} = \sum_{i=1}^{n_j^{SW}} c_{ij}^{SW} n_j^{SW} t_{ij}^{SW}$
	Total cost of the HW resources for the activity	$C_j^{HW} = \sum_{i=1}^{n_j^{HW}} c_{ij}^{HW} n_{ij}^{HW} t_{ij}^{HW}$
	Total cost of the other types of resources for the *j*-activity	$C_j^{Or} = \sum_{i=1}^{n_j^{Or}} c_{ij}^{Or} n_{ij}^{Or} t_{ij}^{Or}$
PRODUCTIVITY	Productivity of the process	$P = \frac{\sum_{j=1}^{n^A} P_j}{n^A}$
	Productivity for the *j*-activity	$P_j = \frac{\sum_{i=1}^{n\_out_j} P_{ij}}{n\_out_j}$
	Productivity of the output of category *i* for the *j*-activity	$P_{ij} = \frac{n\_out_{ij}}{t\_out_{ij}} \cdot compl\_out_{ij}$

output. They consist of critical level of the resources, dependencies, constrains, interfaces, and eventual formalization procedures. In the interest of brevity, the variation factors are not considered here.

Table 4 presents a section of the measurement framework related to GOAL_5. The aim is to evaluate the quality characteristics of a software system, as a user can perceive it. As already cited, the characteristics considered are: functionality, reliability, usability, efficiency, effectiveness, productivity, security and satisfaction. The metrics of Table 4 are at a high level of abstraction. Their values are derived from simpler measures. These metrics can assume a value within a range representing the possible goodness levels. Other metrics can numeric, like those referred to Q40, or Boolean. In any case, some of the considered measurements are subjective. This kind of information is difficult to be interpreted and it should be avoided. In the following, a description of the considered characteristics [9] is given, together with the formulas to be used for evaluating them. The formulas refer to the metrics introduced in Table 4.

- *Functionality* is the capability of the software system to provide functions meeting stated and implied needs when it is used under specified conditions. It is evaluated by the following formula: $F = f_r a_o i_s$
- *Reliability* represents the capability of the software system to maintain a specified level of performance when used under specified conditions. It can be calculated as: $A_s = t_{mf} + t_{mr}$

Table 4. Questions and metrics of the GOAL_5

Code	Metric Identifier	Questions and Metrics
Q39		**Is the software system functionally adequate?**
M39.1	f_r	Adequacy level of the software system to the user requirements
M39.2	a_o	Accuracy level of the output of the system
M39.3	i_s	Interoperability level of the system
Q40		**Is the software system reliable?**
M40.1	t_{mf}	Mean time between failures
M40.2	t_{mr}	Mean time to repair a failure
Q41		**Is the software system usable?**
M41.1	u_1	Attractiveness of the user interface
M41.2	u_2	Existence of a graphical user interface
M41.3	u_3	Existence of an adequate training in the use of the system
M41.4	u_4	Visibility of the status of the software system
M41.5	u_5	Visibility of the action currently performed by the user
M41.6	u_6	Presence of 'undo' and 'redo' functionalities to recover default conditions
M41.7	u_7	Existence of an *interrupt* command to suspend an elaboration to be later recovered
M41.8	u_8	Understanding level of the user interface language
M41.9	u_9	Existence of icons associated to the system commands
M41.10	u_{10}	Existence of a short description of the commands when they are used (tool tip)
M41.11	u_{11}	Use of a significant title for each window and/or screen
M41.12	u_{12}	Use of selection lists favouring the use of adequate values in a correct form
M41.13	u_{13}	Consistence of the language and/or graphics in the working environment
M41.14	u_{14}	Coherence in the presentation of the information
M41.15	u_{15}	Existence of the on-line help for each command
M41.16	u_{16}	Number of accesses to the on-line help and/or manuals
M41.17	u_{17}	Simplicity level of use of the on-line help and/or manuals
M41.18	u_{18}	Comprehensibility level of the on-line help and/or manuals
M41.19	u_{19}	Effectiveness level of the system documentation to solving problems
M41.20	u_{20}	Background colour
M41.21	u_{21}	Foreground colour
Q42		**Is the software system efficient?**
M42.1	r	Adherence to the hardware requirements of the system
M42.2	V	Speedy level of answers of the system
Q43		**Is the software system effective?**
M43.1	n_o	Number of the system outputs
M43.2	n_t	Number of the process output
M43.3	Q_o	Mean quality value of the system output
Q44		**Is the software system productive?**
M44.1	t_{oj}	Production time for each output of activity *j*
M44.2	C_{aj}	Complexity for each output produced by the system in activity *j*
Q45		**Does the software system satisfy the expectations of the user?**
M45.1	S	User satisfaction level
Q46		**Is the software system safe?**
M46.1	s_1	Operative system used
M46.2	s_2	Presence of network
M46.3	s_3	Modality of access to the computers
M46.4	s_4	Modality of access to the software system
M46.5	s_5	Modality of access to database
M46.6	s_6	Management of the user profiles
M46.7	s_7	Availability of a computer for each user
M46.8	s_8	Access of not authorized people
M46.9	s_9	Existence of a backup policy
M46.10	s_{10}	Installing of an antivirus
M46.11	s_{11}	Frequency of antivirus updating
M46.12	s_{12}	Presence of firewall
M46.13	s_{13}	Existence of courses and/or informative brochures on the computer safety

- *Usability* is the capability of the software system to be understood, learned and used by a user and be attractive to the user, when used under specified

conditions. It is measured considering the heuristics proposed in [11], where a software system is considered usable when it is simple to learn, easy to effectively use, simple to remember, inducts few errors with low relevance and is pleasant to use. The formula applied is: $U = \frac{\sum_{i=1}^{19} u_i + f(u_{20}, u_{21})}{20}$ where $f(u_{20}, u_{21})$ is a function that assumes different values for different combinations of the text and background colors

- *Efficiency* is the capability of the software product to provide appropriate performance, relative to the amount of used resources, under stated conditions. It can be calculated as: $\eta_s = rv$
- *Effectiveness* is the capability of the software product to enable users to achieve specified goals with accuracy and completeness in a specified context of use. It can be defined as: $E_s = \frac{n_o}{n_t} Q_o$
- *Productivity* is the capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use. It is calculated by the following formula: $P_s = \frac{n_o}{t_o}$
- *Safety* is the capability of the software product to achieve acceptable levels of risk in a specified context of use. It is expressed by the metric S.
- *Satisfaction* is the capability of the software product to satisfy users' need in a specified context of use. The formula applied is the following: $S_c = Sg(s_4 s_5 s_6 s_{11} s_{12} s_{13}) \sum_{i=1}^{15} s_i$ where $Sg(\ )$ is the sign function.

The values reached by the above characteristics have been translated into the qualitative data *very low, low, medium, high* and *very high*.

5. The WebEv environment

WebEv, Web for Evaluation, is an environment implemented by the authors for supporting the management of measurement frameworks and assessment activities. It represents a practical mean to collect all the available quantitative information regarding the processes and/or software systems of an enterprise, and facilitate its diffusion. The recovered data can be used to identify the quality of the assessed object and provide useful information to be considered during its evolution. WebEv can be used both in the planning and enacting stages of the assessment. In the first stage, WebEv permits the definition of the measurement framework and its customization to the specific context. Then, it supports the configuration and management of the items under assessment and the actual collection of metrics. Using a Web-based interface and, then, collaborating with other experts, the software engineers can define and store in the database the framework to be used in the assessment activities, and inserting the Goals of the analysis, and, then, all the Questions and Metrics required for collecting all the information of interest. The assessors can access via Internet the framework and use the environment facilities for inputting the measures of the metrics, regarding processes, activities and resources, through a web-based user interface. Moreover, WebEv implements the calculations of formulas that value the chosen parameters in terms of the measures of the metrics.

All the information to be managed by the measurement framework through WebEv is stored into a relational database. It contains information about the definition of the measurement framework, in terms of Goals, Questions, and Metrics and its characterization in terms of values the metrics assume. It takes in consideration that the metrics can refer to various kinds of components interacting each other and having different granularity. For example, it is possible to measure aspects related to both processes and supporting systems. In addition, if the evaluation refers to a process, the assessment of all its activities is required and, for each of them, the used resources and the supporting software systems have to be considered.

One of the main requirements of WebEv is the possibility to store a large amount of information that can be accessed from different users acting in different sites and time. With this in mind, Web technologies have been used for its implementation. In particular: the Java programming language, was chosen for its portability and independence from the hardware platform and operating system environment; the Java JDBC (Java Driver Database Connection) API, for the connection with the relational database; the Java Servlet API, for implementing the interface between the Web server and the application. Finally, WebEv is based on the Model-View-Controller (MVC) design pattern that was chosen to achieve maximum reciprocal independence and proper localization of the functionalities.

6. A case study

The effectiveness of the reference framework has been evaluated in an experiment with pilot users. In fact, a peripheral department of the Public

Administration, (a small town in the district of Benevent in Italy) agreed to participate to the pilot project as customer. The Goal definition phase was performed by interviewing the department managers for identifying the processes to be improved. They highlighted that the main need of a public administration body is to increase the citizens' satisfaction and, on the other side, improve the effectiveness of the body employees' tasks. The citizens' satisfaction could decrease due to the long time to queue for putting forward a request and having the related answer. On the other hand, the long time necessary to elaborate an answer was caused by the manual execution of the needed tasks and the time to wait to receive information from external organizations.

In order to analyze the identified process, the administrative manager was interviewed and, at a first level of abstraction, three sub-processes were identified:

- Payment of the land taxation.
- Payment of taxes on cities solid waste.
- Payment for occupation of public ground.

In particular, in the first stage, only the first process was considered. In the following, it will be indicated with the acronym ICI – Imposta Comunale sugli Immobili. It is controlled by law n.504 /1992. The process activities are depicted in Figure 2.

The process is supported by a specific recently developed software system. GOAL_5 has been considered with reference to this software system. Table 5 synthesizes the obtained results. They are the following: the Functionality value is evaluated as Medium because while most of the user requirements are satisfied by the software system, which is specific to support the process, the interoperability level is Medium; the Reliability assumes value High as the interviewed employees affirmed that the system did not present any significant failure or fault since they used it, for this reason it is not possible to provide a quantitative value; the Usability is calculated in terms of 0.84 and considered High for the friendly user interface, and the availability of comprehensive on-line help and documentation; the Efficiency amounts to 0.80, that is High, for the short time required by the system to make elaborations and provide results; the Effectiveness is Medium as not all the output required by the process are produced by the software system; the Productivity evaluates a Medium reflecting the medium complexity of the produced outputs; the Safety results to be Very Low due to the lack of firewall, antivirus, and possibility to access the system and data without user identification; the Satisfaction from the users' point of view is evaluated High, also due to the fact that there few people need to access to the process and then to the system.

GOAL_1 has been applied to the ICI process and the correspondent section of the measurement framework has been evaluated by using WebEv. The metrics and characteristics presented in Tables 2 and 4 have been evaluated. Figure 3 shows the results of their evaluation provided by WebEv. In the interest of clarity the full set of results is presented in Table 6. In particular, Table 6 contains the data regarding the Allocation time and Costs for human, software and hardware resources, and the Productivity for each process activity. The reengineering process we applied by considering also the plan for the eGovernment of

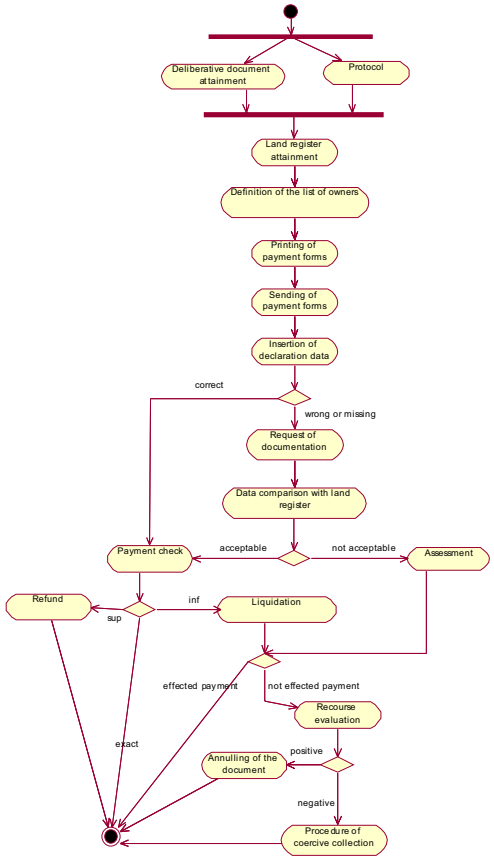

Figure 2. Activity diagram of the ICI process

Table 5. ICI software system data

Characteristic	Quantitative results	Qualitative results
Functionality	*0.51*	*Medium*
Reliability	-	*High*
Usability	*0.84*	*High*
Efficiency	*0.80*	*High*
Effectiveness	*0.55*	*Medium*
Productivity	*0.62*	*Medium*
Safety	-	*Very Low*
Satisfaction	*0,8*	*High*

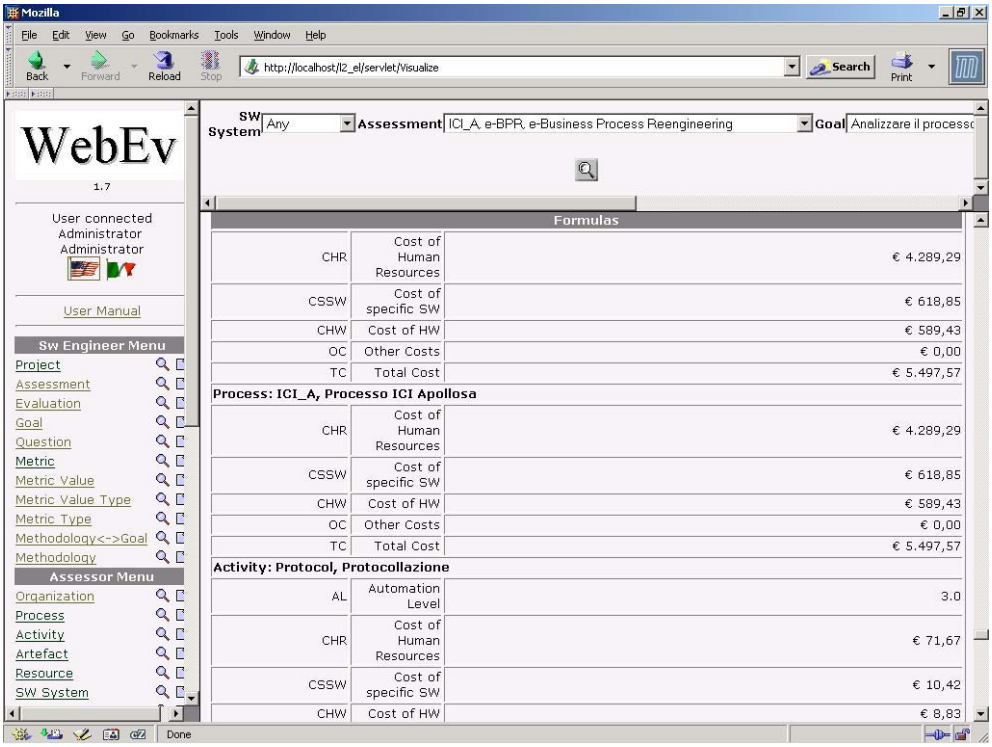

Figure 3. WebEv Screen dump

Table 6. ICI process data

Activity	Allocation time of human resources (min)	Allocation time of specific SW (min)	Allocation time of HW (min)	Activity productivity	Cost of specific SW		Cost of HW		Cost of human resources		Other costs	
					%	€	%	€	%	€	%	€
Deliberative document attainment	*15,00*	*0,00*	*0,00*	*0,040*	*0*	*0*	*0*	*0*	*0,061*	*4,63*	*0*	*0*
Protocol	*253,33*	*190,00*	*190,00*	*0,137*	*0,137*	*10,44*	*0,116*	*8,84*	*0,949*	*71,67*	*0*	*0*
Definition of the list of owners	*40,00*	*30,00*	*30,00*	*0,020*	*0,022*	*1,65*	*0,021*	*1,60*	*0,150*	*11,32*	*0,095*	*720*
Printing of payment forms	*48,00*	*180,00*	*180,00*	*6,667*	*0,130*	*9,89*	*0,273*	*20,76*	*0,180*	*13,58*	*0,013*	*100*
Sending of payment forms	*360,00*	*0,00*	*0,00*	*1,111*	*0*	*0*	*0*	*0*	*1,349*	*101,85*	*0,132*	*1000*
Insertion of declaration data	*650,00*	*650,00*	*650,00*	*0,020*	*0,470*	*35,71*	*0,398*	*30,26*	*2,436*	*183,90*	*0*	*0*
Data comparison with land register	*433,33*	*65,00*	*65,00*	*0,002*	*0,047*	*3,57*	*0,043*	*3,23*	*1,624*	*122,60*	*0*	*0*
Request of documentation	*200,00*	*0,00*	*100,00*	*0,067*	*0*	*0*	*0,107*	*8,10*	*0,750*	*56,58*	*0,006*	*48*
Insertion of payment data	*6000,00*	*6000,00*	*6000,00*	*0,067*	*4,340*	*329,61*	*3,678*	*279,29*	*22,486*	*1697,53*	*0*	*0*
Documentation comparison with land register	*1000,00*	*75,00*	*75,00*	*0,013*	*0,054*	*4,12*	*0,046*	*3,49*	*3,748*	*282,92*	*0*	*0*
Assessment	*2666,67*	*2000,00*	*2000,00*	*0,040*	*1,447*	*109,87*	*1,498*	*113,74*	*9,994*	*754,46*	*0,013*	*100*
Liquidation	*2666,67*	*2000,00*	*2000,00*	*0,040*	*1,447*	*109,87*	*1,498*	*113,74*	*9,994*	*754,46*	*0,013*	*100*
Recourse evaluation	*106,67*	*0,00*	*0,00*	*0,030*	*0*	*0*	*0*	*0*	*0,400*	*30,18*	*0*	*0*
Annulling of the document	*266,67*	*20,00*	*20,00*	*0,060*	*0,014*	*1,10*	*0,012*	*0,93*	*0,999*	*75,45*	*0*	*0*
Procedure of coercive collection	*40,00*	*6,00*	*6,00*	*0,027*	*0,004*	*0,33*	*0,006*	*0,49*	*0,150*	*11,32*	*0*	*0*
Refund	*400,00*	*0,00*	*60,00*	*0,040*	*0,043*	*3,30*	*0,064*	*4,86*	*1,499*	*113,17*	*0,004*	*26,80*
Payment check	*13,33*	*10,00*	*10,00*	*0,040*	*0,007*	*0,55*	*0,009*	*0,67*	*0,050*	*3,77*	*0*	*0*
Total	*15159,67*	*11226*	*11386*			*620*		*590*		*4289,39*		*2094,80*
Total cost of process				*7594,19*								
Process productivity				*0,495*								

the European Union [7]. The aim of the plan is to facilitate the services exploitation by the citizens and improve the relationships between the citizens and the public administration. The plan defines four interaction levels. The first level is informative and entails the on-line availability of the information required to exploit the services; the second level entails a one-way interaction, in which it is only possible to download the modules required; the third level is the two-way interaction, in which it is possible to activate an on-line interaction with the process; the level four entails the execution on-line of the process, including transactions such as payment and delivery.

The data collected for the process and supporting software systems have been analyzed to extract the requirements for their reengineering. In the following, there are some main elaborated considerations:

- The activities more expensive are those massively involving human resources. They are the Insertion of declaration data, Insertion of payment data, Sending of payment forms, Printing of payment forms, Assessment and Liquidations. This is mainly due to the lack of an adequate software system supporting them.
- The Insertion of declaration data is the most expensive activity because all the data contained in the declarations presented by the citizens have to be inserted in the database using a specialist software system. Therefore, the process can be reorganized making the users personally insert their data using a Web-based software system.
- The Sending of payment forms and Printing of payment forms are two activities that can be improved by extending the specific software already used and introducing a new functionality for printing the payment forms and the users' mail addresses. In addition, the system could provide a functionality to electronically send these forms to the citizens.
- A new software system can be introduced to support the Data comparison with land register and Payment check activities. Actually, these activities require a data comparison manually performed by the user. However, all the information necessary is stored in databases, and the needed checks can be automated with the consequent introduction of relevant benefits to the costs of the process.
- In the Definition of the list of owner's activity, the time allocated for the human resources is almost equivalent to the allocation time for the software and hardware systems. However, the activity should not really require the allocation of human resources for long time, as the list should be automatically produced by the system. The rate between the time allocated for human and software resources should be similar to that of the Print payment forms activity.
- Contrarily to the Data comparison with land register activity, the Documentation comparison with land register activity cannot be completely automated because the information to check is on paper. However, the time required for the allocation of human resources is very high and a partial reengineering of the activity is required to reduce it.
- To be coherent with the current laws, the Annulling document activity cannot be improved. Indeed, the tasks involved in this activity have to be manually performed by the employees and, therefore, the time allocated for the human resources is higher than that allocated for software and hardware systems.

To summarize, the process can be generally improved redesigning it and extending the supporting software system for permitting the use of the Web technologies. This should improve the process performance and give to the citizens an active role in the process by using the Internet.

7. Conclusions

The need to evolve business processes towards *e*Business by introducing innovative information and communication technologies implies the analysis of the current processes of an enterprise and of the supporting software systems. This paper proposed a measurement framework based on the GQM paradigm for assessing *e*Business processes and supporting software systems from both quantitative and qualitative points of view. The proposed framework is generally applicable to any business process and supporting software system after its instantiation to the specific context. The collaborative environment WebEv supports the instantiation and application of the framework.

The measurement framework has been applied to a real context concerning a small local public administration. The research described addresses relevant issues for the evolution of business processes in the *e*Business context.

Nowadays, it is important to react to the central role that Internet is playing in the definition of new business relationships within and among enterprises. Therefore, the process innovation cannot be performed without a careful evaluation of the impact of process changes on existing legacy systems and the introduction of innovative technologies. The paper proposes a measurement framework to support the cited assessment and to facilitate the identification of the evolution requirements in the context of the process innovation. Some of the lessons learned are referred to the necessity to formalize as rigorous as possible the

instruments used in the assessment activities. In fact, the rough definition implies a continuous interaction with the process owners that can misunderstand the proposed questions. Additionally, the interviewed employees should not have any possibility to provide information deriving from their ideas and feeling. This can be avoided by formulating the questions in such a way to suggest the possible metric values and facilitating the objective interviewed employees' answers.

Furthermore, the continuous meetings necessary to answer to the measurement framework can be avoided if a software tool exists facilitating the interaction and the data entry through the Internet. The paper presents such a tool that will be further extended for the next applications of the measurement framework. In fact, actually the results are almost quantitative. Then, though they are objective they are still numbers that require an interpretation by the business analysts. The future work of the authors will include in the framework critiquing techniques that will allow to reports to the analysts qualitative results from the analysis on processes and systems. The goal is that managers immediately can understand the points where an organization needs interventions without analyzing detailed aspects. On the basis of these considerations, WebEv will be integrated with a modeling tool and include functionalities for graphically depicting qualitative results.

Finally, an instrument to be successfully used with early adopters of BPR and assessment methodologies needs to be tested in various contexts in order to be refined on the basis of various needs and received feedbacks. For this reason, the proposed measurement framework will continue to be experimented. With this in mind, the authors are interested to share information and experiences with whoever interested. Both the measurement framework and software environment will be transferable to researchers that want to apply them in various contexts.

8. References

[1] L. Aversano and M. Tortorella, "Evolving legacy system toward eLegacy system in eBusiness context", *Proc. of 7th European Conference on Software Maintenance and Reengineering*, IEEE CS Press, Benevento, Italy, March 26-28, 2003, pp. 201-210.

[2] V. R. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach", *Encyclopedia of Software Engineering*, Wiley, 1994.

[3] V.R. Basili and D.M. Weiss, "A Methodology for Collecting Valid Software Engineering Data", *IEEE Transanctions on Software Engineering*, IEEE CS Press, vol. 10, no. 6, November 1984, pp. 728-738.

[4] Becker S.A. and Bostelmann M.L., "Aligning Strategic and Project Measurement Systems", *IEEE Software*, IEEE CS Press, vol. 16, no. 3, May/June 1999, pp. 46-51.

[5] K. H. Bennett, M. Ramage, and M. Munro, "Decision Model for Legacy Systems", *IEE Proceedings Software*, vol. 146, no. 3, June 1999, pp. 153-159.

[6] L. Buglione and A. Abran, "Balanced Scorecard and GQM: what are the differences?", *FESMA-AEMES Software Measurement Conferences*, 2000.

[7] Presidenza del Consiglio dei Ministri, Dipartimento per l'Innovazione e le Tecnologie, Allegato n. 1 "Front office e servizi di e-government per cittadini e imprese", http://www.pianoegov.it/.

[8] M. Hammer and J. Champy, *Reengineering the Corporation: a Manifesto for Business Revolution*, NY: HarperCollins, New York, 1993.

[9] ISO, *Information Technology - Software Product Quality*, *ISO/IEC*, 1999.

[10] I. Jacobson, M. Ericsson, and A. Jacobson, *The Object Advantage: Business Process Reengineering with Object Technology*, ACM Press, Addison-Wesley, 1995.

[11] J. Nielsen, *Usability Engineering*, Morgan Kaufmann Publishers, October 1994.

[12] D. Norton and R. Kaplan, "The balanced scorecard: measures that drive performance", Harvard Business review, vol. 70, no. 1, 1992.

[13] G. Verdugo, "Portfolio Analysis- Managing Software as an Asset", *Proc. of International Conference Software Maintenance Management*, Software Maintenance assoc., New York, 1988, pp. 17- 24.

[14] V.A. Zeithaml, A. Parasuraman, and L.L. Berry, *Delivering Quality Service: Balancing Customer Perceptions and Expectations*, Free Press, March 1990.

[15] S. Rugaber and S. Doddapaneni, The Transition of Application Programs from COBOL to a Fourth Generation Language, *Proc. of Conference on Software Maintenance CSM'93*, Montreal, Canada, IEEE CS Press, 1993, pp. 61-70.

[16] Sneed H.M, "Risks involved in reengineering projects", *Proc. of 6th Working Conference on Reverse Engineering*, Atlanta, GA, IEEE CS, 1999, pp. 204-211.

Acknowledgments

The authors would like to thank the anonymous referees, whose constructive suggestions have enabled to improve the paper.