Collaborative Systems Evaluation

Dragos PALAGHITĂ
Economic Informatics Department, ASE, Bucharest, Romania,
mail@dragospalaghita.ro

Abstract: *Types of collaborative systems are analyzed. A quality system is established. Quality characteristics used in evaluation are discussed. Domain specific systems are analyzed. Real world examples of working collaborative systems are mentioned. Overall evaluation of collaborative systems is performed.*

Key words: *collaborative system, evaluation, quality, control*

1. Electronic business systems

According to [15] regardless if a companies' electronic business are oriented towards individual clients or towards other companies, the success depends on solid web fundaments. The key elements of an efficient electronic business infrastructure are related in an equal extent to the management and technological strategy. In order to create a competitive company capable to introduce and to quickly apply business initiatives the management must rethink the role of information technology and to modernize the technological infrastructure, A standardized and flexible architecture, well managed content and advanced user management systems, which will help more then obstruct business processes are within the numbers of the essential components o a successful web strategy.

Building solid and scalable web fundamentals needs cooperation and vision. The following aspects make developing a web strategy difficult:

- a lot of organizations look at costs associated with information technology only as indirect costs that must be reduced; IT budgets tend to be centralized and rarely are associated with initiatives of individual business units;
- decisions regarding infrastructure have a tactical nature and are often taken at a department level; this fact leads to a fragmented technological architecture, to the proliferation of data centers and the absence of standards considering the whole company; adding new applications and capacities implies high costs and is slow;
- companies have manual procedures which need readjustments between the web sites and the internal systems like planning the enterprise resources and chain management systems; this process is slow, error prone and needs a high work load to complete;
- the creation instruments and web publishing are hard to use, such that companies must delegate these tasks to professional full time authors;
- professionals in the security field tend to focus on the specific objective of protecting company assets, frequently enough by sacrificing the business dynamics; frustrated users avoid security procedures and mechanisms in order to complete their tasks.

Finding solutions to these problems is of a critical importance in order to introduce a ne web strategy. Following are some suggestions for it to succeed:

- the IT component should be considered as a strategic element in order to make a business more competitive; presently the technological infrastructure is the one that is being centralized while as the main initiatives that include technology are promoted and financed bi individual business units;
- the progressive companies presently take strategic decisions regarding the infrastructure and apply standards at a company wide level; companies that have technological infrastructures and business processes that have developed step by step wisely proceed to building a new base for electronic initiatives; the first step is implementing architecture standards like Java, Hypertext Transfer Protocol and Extensible Markup Language company wide;
- important companies integrate their web sites with applications essential for the company, either owned or owned by partners; this allows the rapid unfold of complete processes; for example when a client consults the web site to see if a certain product is available, a message is automatically sent to the internal corresponding company systems such that the client will receive correct and up to date information; in the middle of these integration applications B2B is XML, a language based on an open standard which offers a common method of describing data allowing web pages to function as recordings in a data base;
- progressive companies introduce advanced content management systems with instruments which are intuitive enough to offer the possibility even to occasional authors to create and publish content;
- the security professionals help to quickly launch the new business initiatives, protecting the company assets at the same time; presently, their objective is to facilitate the establishing of new relations for companies and authorized persons to complete their tasks without compromising the assts that hold a major importance in the company.

Companies create and implement web strategies solidly planned and enjoy extended advantages. Important companies try to align their IT departments with different business units giving birth to partnerships which will identify new possibilities and apply technologies to obtain success. When accent is placed on business competitiveness, discretionary infrastructure and application development budgets are decentralized, such that IT resources will be dedicated to these initiatives which bring the most important competitiveness degree to the company. At the same time The main objective for IT is to create an unique, flexible and complete web platform which will permit business units to rapidly implement and continuously develop their web initiatives.

A strategic designed architecture will be developed around the companies long term objectives, instead of short term objectives of business units considering the need for flexibility. This type of design will unify technological systems of the company and will facilitate the introduction of new applications and business initiatives.

The electronic integration technology will uphold business processes that include two or more applications within the company and between companies. For example, when an electronic integration system exists all requests made through the company web site will automatically go through the request administration system without the need for an employee to rewrite the data. The same, a query about the availability of a product on the web site will generate a message that will be transmitted in real time to the internal corresponding system such that the client will have correct and up to date information. Companies are able to extend this integration beyond their organizational limits in order to automate transactions with commercial partners.

The request of web content is no longer a privilege of professionals but a process close to the user, a lot of the instruments for development have a large scale availability. When creation and publishing activities are made by persons in the organization without much specialty training, companies use content management systems to store information in a warehouse where they are easily accessible for reuse or reorientation with different goals. This is important especially for maintaining a solid and consequent brand to the level of the entire organization which unfolds electronic business.

As company departments have become integrated and clients turned into users of organization electronic business systems, the traditional security mechanisms have made room for authorization programs – modern security systems which administer and store intelligently user data and correlate them with the organization access rules. These rules define authorization in business terms, according to the role each person posses in different business processes placing at the disposal of the company, the necessary means to identify users and to offer them in a secure manner data and functions they need, considering the business necessities and policies.

2. Quality characteristics for collaborative systems evaluation

When evaluating collaborative systems there are certain very important quality characteristics that must be considered in this process. These characteristics, if in high levels, add real value to the system according to [14].

According to [18] a quality system must be developed in order to capture the essential software quality characteristics. Figure 1 presents the main quality characteristics of the collaborative system.

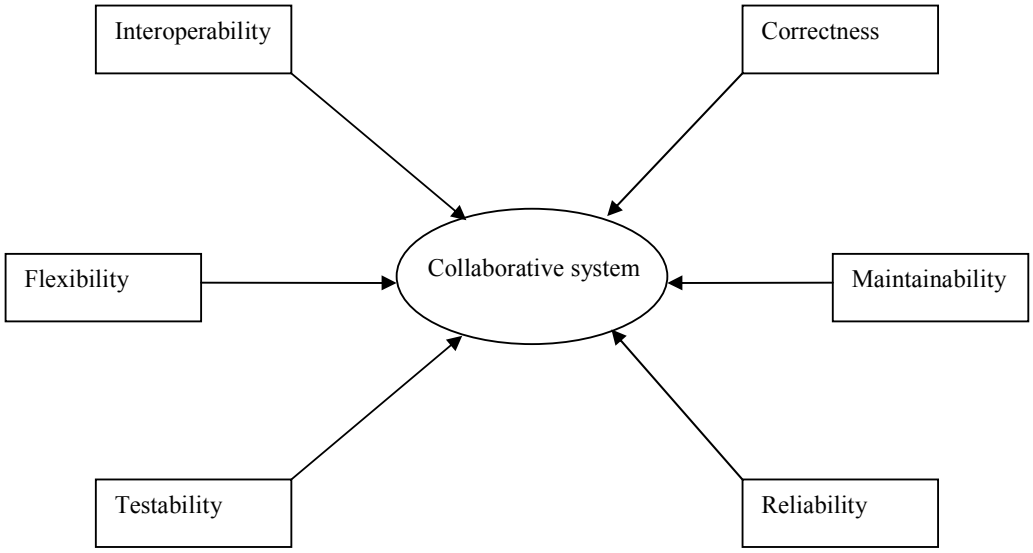

Fig. 1. Quality system

Reliability of the collaborative systems is their condition of existence. Without it the collaborative system structure will fail and customer satisfaction levels will be very low.

According to [9] non-reliability is obtained by:

- using non-adequate content;
- making operations with particular values which position the condition indicators such that further processing is impossible;
- altering the content of control variables which determines referring memory zones with random content.

Because of this programmers who develop collaborative system components apply strict rules that eliminate the conditions for non-reliability. According to [6] all application functional and security bugs are in fact related to code quality which if improved will minimize undesired effects in reliability.

Also for collaborative system components there are elements that solve a great diversity of requirements related initially to non-reliability.

If for the personal identification number there is a unanimously accepted procedure, it will be incorporated in the new applications in which using this code is an issue. If for managing user names and passwords there are standard procedures defined then collaborative system developers will use these procedures in their applications. Concerning the validation of email addresses the issue of incorporating collaborative system is still viable according to [16].

According to [10] developing models to study reliability it's not a special problem because there are:

- software reliability modules;
- automated data collecting procedures regarding the behavior of any software product;
- software that implements algorithms used for module behavior estimation.

The most important aspect is to start a systematical activity to study the reliability oriented to collaborative systems and results must be used to increase the reliability of collaborative systems of this type of IT products.

Maintainability in [3] for classic software products means all the modifications made after an interval in which the software product was used currently. Maintainability is necessary due to:

- changes made in processing algorithms that are developed over time;
- restructuring of input data;
- passing to other ways of presenting the results;
- creating a new positioning of the application with respect to the user convenience of moving the application to the online environment.

In the case of collaborative systems maintainability is a continuous process because the development process includes all specific requirements related to maintainability.

Maintainability in collaborative systems components is more efficient according to [11] compared to maintainability of classical products because of a the fact that the system is being used differently and maintenance will take place on independent modules rather then on the whole product that enables other developers and contributors to examine and improve parts of the component that they consider faulty without this being obvious to the user.

According to [13] the maintenance process has an important impact on the resulting collaborative system software product if specific coding standards and coding rules are used. This is particularly important in increasing the overall quality level of the collaborative system component during maintenance because it will provide the component with superior results during testing and ease future development.

Some users exploit already existing collaborative system components, programmers add developed components while other programmers work on improving existing components.

Correctness in [5] is a quality characteristic which imposes the collaborative systems' in the world of users and programmers. Any component once launched in production is checked thoroughly in means of correctness and very fast any overlooked defects appear. This is the reason that before launching anything in the collaborative system environment it must be checked thoroughly.

Correctness is evaluated through:

- demonstration using a strict mathematical basis and adequate software;
- testing using well structured, well dimensioned and with a high coverage degree data sets data sets.

The results are known and this is the reason why by starting from generated entry data it is verified if the new collaborative system procedure is or is not correct. A programmer, before beginning to develop a program must introduce it in the collaborative system environment first to see if:

- there are similar products and to be certain if there are similar products and if his contribution is better;
- has access to more test data sets in order to be certain that his procedure is completely tested.

When launching a component the programmer must:

- specify the limits;
- make the testing procedure public;
- indicate the risks he removed;
- to describe the advantages his component has over other similar ones.

All of these create a clear context reporting to the ensured correctness of the new component when launching it in the collaborative system environment.

Testability according to [7] is defined as the extent to which a certain component supports testing after it was modified or launched. This definition is transposed to collaborative system software by adding the following:

- each collaborative system developer must test all his modifications;
- after each improvement brought to the component thorough documentation must be done to ease the future testing effort;
- all test data sets, testing methods and test results must be made public such that the whole testing procedure will be redone with ease by any other collaborative system contributor.

The testing process is considered to be successful if and only if the same tests done on the component using the same data sets and the same testing method yield similar results form more then one collaborative system developer according to [4].

Testability is more important as on its results depends the future success of the component, if the testing is poorly done then no user or developer will integrate the component or respectively use or improve it further.

The testing process has several aspects considering the collaborative system state:

- the collaborative module is new, testing is done by the developer and all the data related to the process is posted publically;
- the collaborative module is old and is modified rarely, the testing process is done by each developer that brought modifications, posting all the data related to the testing process publically;
- the collaborative module is old but modified frequently by developers and team of developers, here the testing is scheduled to occur at specific periods of time

considering freezing the development process on the last stable build before the testing thus no further modifications will be brought; the testing process is done by a team and all the test data is posted publically to be available for the public and to be of support for the next testing sessions.

Testability according to [17] is constructed based on the following components:

- controllability of the component consists in the ability to control the state of the collaborative system component under test;
- observability is defined as the ability of the testing process to make the results of the test clear to the tester;
- isolation ability is the trait of the testing process to isolate a part of the component for independent testing, this reveals how separate parts of the component behave during testing and sets effectiveness and quality correlations between the sub-components of the collaborative system product;
- automation is the characteristic of the testing process that allows a certain degree of automation in testing the functionality of the component, as the degree of automation increases the quality of the testability characteristic increases as well;
- intelligibility represents the extent to which a sub-component of the collaborative system product is self explanatory or documented such that the testing process reaches the target and reveals whether the functionality of the product is as expected;
- according to [8] class testability is a crucial component in establishing that all paths in the source code yield expected results.

Non-testability is achieved by:

- lack of source code functionality documentation;
- no documentation of newly added components or improvements;
- non-homogenous source code throughout the collaborative system component;
- error prone source code, the code generates too much errors such that the testing process becomes tedious;
- developers worked anonymously and can't be reached in order to provide clarifications on the intended behavior of the collaborative system component;
- lack of well constructed data sets that can be used specifically to test separate areas of the component functionality;
- badly constructed theoretical hypothesis based on flawed understanding of the collaborative system components' functions.

Non-testability is characterized by:

- long time of establishing the testing procedure;
- tedious work to complete each test stage;
- uncertainty if the test results are correlated with reality;
- questionable product behavior;
- low product trust;
- component avoidance as developers and users alike will avoid improving or respectively using this component as the results that it brings hold no guarantee;
- low performance of the collaborative system component;
- unpredictable behavior of the product.

Testability is one of the most important characteristics of collaborative system software as it provides the developers and users with vital information regarding the overall quality of the collaborative system component.

Flexibility according to [12] is the ability of the software product to easily adapt to modifications. With each new modification there will be an effort to adapt and optimize the communication of the existing code with the new one. Flexibility is measured in man

hours needed to add the modifications and make all reconciliations with the old source code.

If the collaborative system module is badly developed flexibility will be low and each modification made will require more and more time until the component will be abandoned being easier to rewrite the code in order to bring improvements or fix defects.

Flexibility is achieved considering:

- careful examination of the problem that the collaborative system component must solve and try to obtain a high degree of generality for it;
- when developers add their contributions they must keep in mind the following principles:
  - source code generality try as much as possible to use existing classes or functions or develop new generic ones;
  - avoid writing code that already exists in another module;
  - coding rules and standards must be respected;
  - leave room for other improvements and document all new code thoroughly;
- develop dynamic code that is easily adaptable to new conditions of exploitation;
- each developer must carefully examine all the side effects that inserting the modification will have over the components' functionality;
- the modifications must be rapidly integrated in the old code, this is easily done if the components' source code is homogenous and coding rules are respected throughout the source code.

Non-flexibility has the following characteristics:

- inability of the software component to be adapted to changes in the environment without an important work time allocation;
- substantial efforts to make modifications and improvements in the collaborative system component as a response to technology changes;
- impossibility due to high costs to adapt the collaborative system component to new design requirements.

Flexibility is a valued characteristic due to the existence of uncertainty in the collaborative system environment. The dynamicity of this specific domain affects all software components thus unforeseen modifications or improvements arise at any given time and a high level of flexibility is vital for the fast and efficient development of any collaborative system component.

Interoperability is according to *ISO 9126* the ability of the software product to interact with one ore more specified systems. In collaborative system interoperability is achieved considering the following:

- the collaborative system module must use generic communication protocols;
- the collaborative system module must be thoroughly tested in order for the developer that integrates it in a larger system will know its limits;
- the collaborative system module must be well documented such that the source code is properly understood such that interoperability is properly conducted;
- the source code must respect generally approved coding standards in the collaborative system community such that the code has a high readability level and guarantee a high level of compatibility with interacting systems;
- the functionality of the collaborative system must be properly explained, test data sets must be available so must be expected results such that the interactions with other systems perform as expected;
- the collaborative system must have strict logging procedures in place such that all inputs and outputs generated by interactions with other systems are well observed by the developers and testers;

- the collaborative system must be able to receive input and provide output in as many formats as possible in order to guarantee a high level of system interaction.

Interoperability is an important characteristic because it sets the basis of integrating successfully the developed collaborative system modules in other systems and insures proper interaction with them.

Non-interoperability appears because:

- not all institutions or organizations use the same data formats in their reports and files;
- lack of conformity to a certain standard considering network protocols and information structure;
- frequent change in the environment leading to changes in the system functionality;
- complexity of individual modules and increased difficulty of establishing a successful interaction according to [17]
- lack of understanding the functionality of a collaborative system component due to high complexity or lack do documentation and misinterpreting its interaction possibilities with the system this leading to errors and program failures.

In order to obtain high interoperability levels the collaborative systems must implement as many network communication protocols and input-output formats as possible. The desired network protocol or data format is controlled by specific environment variables set in the configuration file belonging to the component such that the interaction with the system or systems goes according to plan and a uninterrupted interaction is obtained.

3. Public administration system

In the field of public administration in Romania there are several collaborative systems as:

- ***ProTAXI***, to manage duties and taxes collection. This system represents a complete solution for duties and tax management. It allows local budget management obtained from duties and taxes and the computation of financial obligations of citizens according to asset category, quantity and in conformity with the standing normative acts, the evidence and unitary pursuit of the fiscal situation of a citizen, managing identification data of citizens and juridical entities and maintaining their modification, editing verbal processes, pay receipt and the citizen requests;
- ***RUSAL***, is an informatics solution in the integrated human resources management and remuneration field . It is addressed to medium and large public administration organizations, institutions which face problems that have a high degree of complexity;
- ***eTax++,*** is an electronic system for receiving local duties and taxes in a mayor office;
- ***ProTel,*** is an integrated telephonic Call Center type system with a data base; it allows data base interrogation by the citizen using analogical phone lines put together in a hunting type group;
- ***ATLAS,*** is an integrated public administration system; it includesduties and taxex modules, public domain, information management, workflow and electronic archive services, financial and accounting functions, marital status, human resources and remuneration, social care, cadastre and urban planning.

4. Production system

In [2] the integration of quality in production control systems has become an actual request with the growth of quality requirements.

The quality integration form with production has evolved along different stages of integration. In turn the complex problem of quality has known different approaches according to pursued objectives, means and instruments used. To this extent the following stages are highlighted:

- *final product inspection*, with the objective to select and grade products, in order to obtain product groups and uniform characteristics, as a result of applying comparison and measurement techniques;
- a stage of *statistical quality control* which introduces process control in order to exclude noise; using instruments and techniques for statistical control is made under the coordination and supervision of a quality control department;
- a superior stage is marked by the implementation of *the quality insurance system* in which a responsibility distribution is made for quality to functional departments, starting from conception and ending with delivery, in order to control prevention, development and evaluation of quality in all production stages.

Integrating the quality system and the production control system is modeled in agreement with the diversity of production systems and fabricated products.

In discrete production the following distinct types of production systems are defined:

- *process oriented production systems* in which complex and divers products are fabricated in small or average sets; in this case statistic quality control has as its objective operation validation in each stage of production; due to the reduced set size, the resulting conclusions from the evolution of values associated with sizes are much to less used for correcting upstream processes and more for resulting operations; this type of system is presented in figure 1;

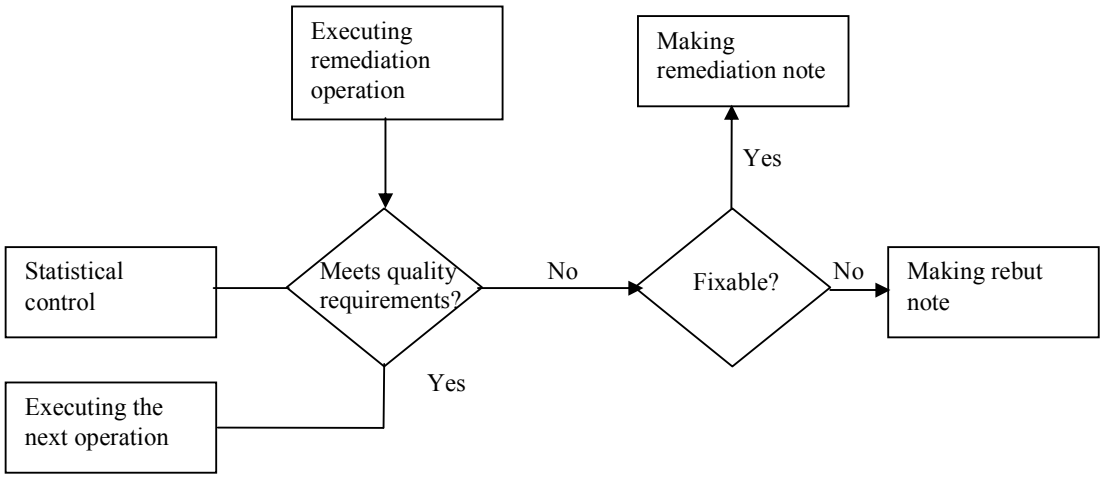

Fig. 2. Decision process regarding production validation and quality in process oriented systems

- *product oriented production systems*, in which lower complexity or diversity products are fabricated in large series; in this case statistical quality control has as an objective maintaining process quality parameters and products in accepted limits by the control technology; for these the evolution of selected attribute deviation value

sets analysis is of great importance; in this case the systems ha a larger set of procedures for processing and representation of the value evolution for each different set and the data base includes in its structure the measured sets for the availability interval of the set or command in work; such a system is presented in figure 2;

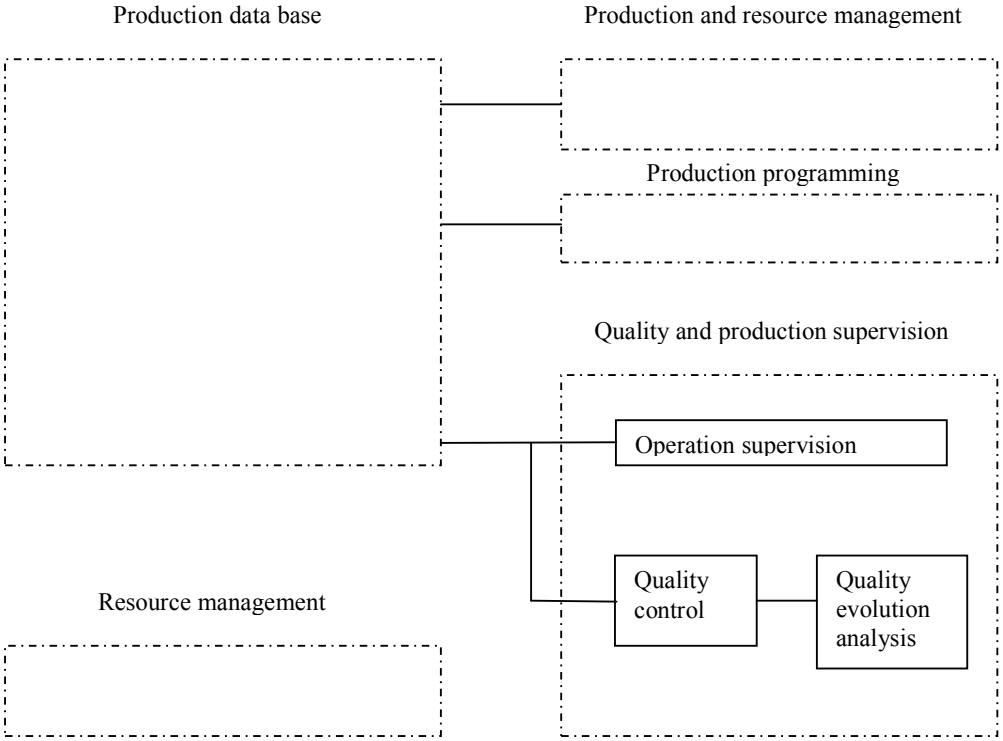

Fig. 3. Quality and production integrated control system for product oriented systems

The problem of integrating production with quality is complex and reflects the diversity of adopted solutions for developing control systems. All of this said, the basic criteria are essential for structuring the system, and the set of procedures for analysis and representation for the evolution of measured values are an effective instrument for accomplishing the integration objectives.

5. Education oriented system

The educational system, respectively the education oriented system are collaborative by definition. Collaboration exists in permanence between teachers and between students or between professors and students.

The e-Learning platform of a collaborative system oriented towards education allows the distribution of courses in text or graphical and multimedia formats and contains synchronous and asynchronous communication elements and advanced testing systems.

Education oriented systems offer, also, facilities in the domain of student evaluation and assistance for the management and supervision of:

- performances and activities at teacher and school unit level;
- curricula elaboration and the methodological norm of validation for integrating computer assisted training in the school curricula; the evolution of school curricula

and the teaching methodology is a continuous process, improvements being added in each stage;

- training at all levels; training is a process addressed in immediate terms to the teaching staff, in order to be transmitted to students, being oriented towards using the whole ensured infrastructure; it begins from basic elements as typing exercises, obtaining generic computer knowledge and IT in general; the means and support which will be used for training are among others preparation courses, at first of short duration, didactic materials for different levels, Internet access, equipping schools with educational software available also to the teaching staff;
- creation and enrichment of the didactic material base.

Education system are widely spread nowadays but not all of them follow the quality characteristics that this type of system needs to have in order to be correct and competitive. This type of education approach must be correct to all students participating in the course, must offer up to date information and well structured digital content, the teaching staff must have full access to teaching material and have the obligation to keep it up to date at all times and correct any errors in the curricula's associated digital content if any.

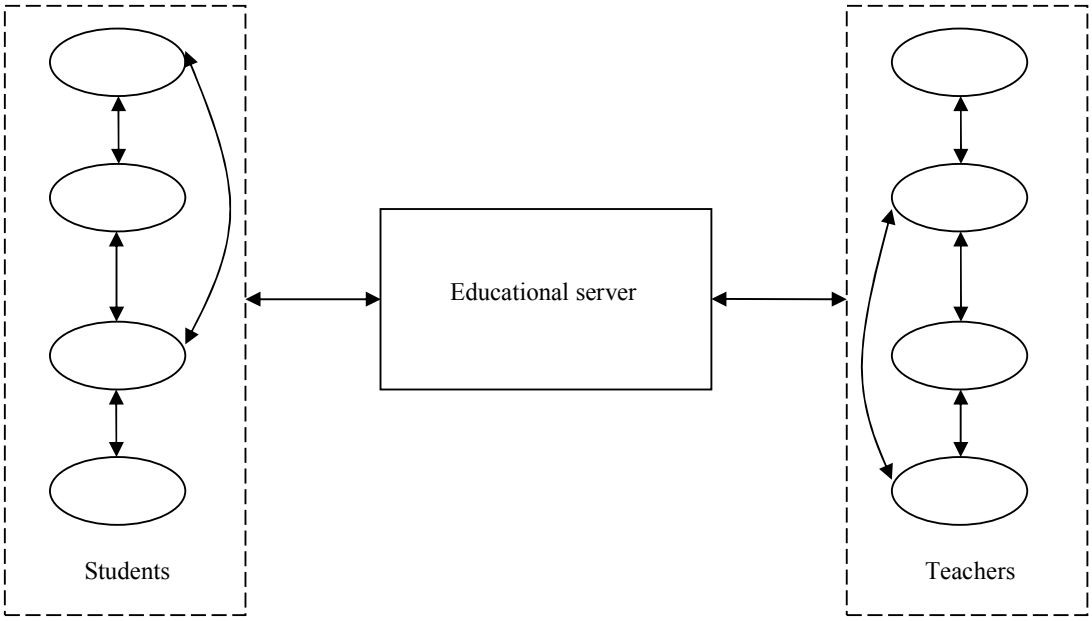

Fig. 4. Education oriented system

Education oriented systems offer grater possibilities compared to classic ones and provide important cost reduction associated with travelling and with the use of digital content instead of the classical printed documents. E-testing is also an important aspect of education oriented systems as it provides real time evaluation of students indifferent of their geographical location.

6. Media application development system

The development of media applications was indirectly caused by the increase of electronic products in current use, These systems include commutation stations for cellular infrastructures, terrestrial, satellite or cable. In all of these applications data processing ins considerably more sophisticated then the classical filters and transformations that characterized these systems during 1960-1970. Generally the performances are more important due to the increase of data rate and complexity of algorithms. Paticularly the performances are more important because they allow the development of more interactive and sophisticated systems. A grater accent is placed on flexibility and functionality diversity based on operating conditions. The electrical consumptions and cost are also important because they affect the overall system cost and its performance.

Media systems are a key factor because:

- they are used in business presentation to better represent data and reports to stake holders;
- business processes are better presented using state of the art media systems;
- using complex media systems gives the users a more realistic feel of the companies products;
- the development of the technological infrastructure permits complex media analysis which leads to advances in medical science as diagnostic tools are being developed based on x-ray photos or electrocardiograms in order to determine an accurate diagnostic.

A media system for business process representation is presented in figure 5.

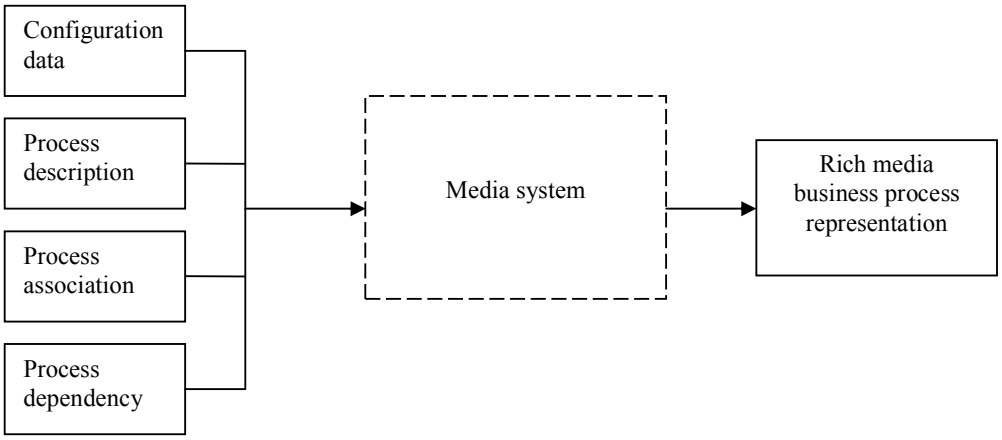

Fig. 5. Business process representation media system

Media systems today are a part of every day life, they revolutionized the market and business processes alike by implementing complex advertising techniques and improving the understanding of complex business process.

7. Conclusions

Respecting simple though effective standardization throughout the company will grant full effectiveness to the collaborative system. Considering the most important quality characteristics in the system and taking steps to improve them will result in a better, more competitive and easier to use system. The integration of all business processes in one single web based application gives more power to the users and provides them with updated information on the companies' products. This is the next step in business evolution and keeping technology up to date is critical in not loosing the competitive edge.

References

[1] I. Ivan, C. Boja and C. Ciurea, *Collaborative Systems Metrics*. Bucharest: ASE Publishing House, 2007.

[2] D. Popescu and M. Popescu "Criterii de integrarea a producţiei şi calităţii în sistemul de control al producţiei şi calităţii", *Revista Română de Informatică şi Automatică*, Bucureşti, 1998.

[3] I. Ivan and C. Boja, *Metode Statistice in analiza software*, Editura ASE, Bucuresti, 2004.

[4] I. Ivan, P. Sinioros, M. Popescu and F. Simion, *Metrici software*, Editura INFOREC, Bucuresti, 1997.

[5] I. Ivan and G. Nosca, S. Capisizu, *Auditul sistemelor informatice*, Editura ASE, Bucuresti, 2005.

[6] M. Howard and S. Lipner, *The Security Development Lifecycle*, Microsoft Press, Redmond Washington, USA 2006.

[7] P. Pocatilu, *Costurile testarii software* Editura ASE, Bucuresti, 2004.

[8] M. Bruntink and A. Deursen, "An empirical study into class testability" *Journal of Systems and Software* vol. 79, Issue 9, New York, NY, USA 2006, pp. 1219 – 1232.

[9] I. Ivan and C. Ciurea "Quality Characteristics of Collaborative Systems" in *Proc. The Second International Conferences on Advances in Computer-Human Interactions*, vol. 1, Cancun, Mexico, 2009, pp. 164 - 168

[10] I. Ivan, A. Visoiu, C. Ciurea and D. Palaghiţă "Model bases and software quality metrics refinement" in *The 4th International Conference "Economy and Transformation Management"*, Timisoara, 2008

[11] M. Popa, I. Ivan, D. Palaghiţă and B. Vintilă, "Quality metrics of citizen oriented informatics applications" in *The Forth International Conference on Applied Statistics – ICAS4*, Bucharest – ROMANIA, November 20-22, 2008

[12] I. Ivan, A. Visoiu and D. Palaghiță, "IT Project metrics" *Journal of applied quantitative methods*, vol. 2, nr. 3, 2008, pp. 302 – 319 [Online] Available: http://jaqm.ro/issues/volume-2,issue-3/pdfs/ivan_visoiu_palaghita.pdf

[13] www.wikipedia.com accessed 29.07.09

[14] I. Ivan and C. Amancei, *Stabilirea coeficientilor modelului global de calitate software,* Editura ASE, Bucuresti 2006.

[15] I. Ivan, L. Săcuiu and D. Milodin, "The development of computer science oriented towards the citizen" in *The Proceedings of Journal ISOM* vol. 2 No.2, 2008

[16] I. Ivan and C. Ciurea, "Entry data validation in citizen oriented applications" *4th International Conference on Applied Statistics*, Bucharest, Romania, 2008.

[17] I. Ivan and I. Rădulescu, *Analiza comparată a complexită□ii entită□ilor text generate prin tehnici de programre*, Editura ASE, Bucuresti, 2006

[18] D. Palaghiță, "Quality characteristics of open source components", *Open source science journal*, pp 38-57 [Online]. Available: http://www.opensourcejournal.ro/

Author

Dragos PALAGHITĂ graduated from the Academy of Economic Studies of Bucharest, Cybernetics Statistics and Economic Informatics faculty, Economic Informatics section in 2008. He is programming in C++ and C# and his main areas of interest are Informatics Security and Software Quality Management. Currently he is undergoing PhD studies at the Academy of Economic Studies of Bucharest, Cybernetics Statistics and Economic Informatics. He published 14 articles in JAQM, Informatica Economică Journal, Economie Teoretică □i Aplicată Journal, Revista Română de Automatică □i Informatică.