COSTS OF CITIZEN ORIENTED INFORMATICS APPLICATIONS

Ion IVAN, Daniel MILODIN, Mihai GEORGESCU, Bogdan VINTILĂ, Cristian CIUREA, Constantin Daniel AVRAM
ionivan@ase.ro, daniel.milodin@ase.ro,
mihai.georgescu@gmail.com, vb@vintilabogdan.ro,
cristian.ciurea@ie.ase.ro, costin.avram@gmail.com
Academy of Economic Studies, Bucharest, Romania

Abstract
The characteristics of the citizen oriented informatics applications and their development and life cycles are described. The difference between estimating, calculating and evaluating is highlighted in order to understand exactly the significations of the costs. Expenses structures for the development, use, maintenance, reengineering and removal from use of the COIA are identified. Models of estimation, calculus and evaluation of COIA for the development and life cycles will be built. The quality of the models will be analyzed for their validation.

Keywords: *model, COIA, costs estimation development, lifecycle.*

1. Citizen oriented informatics applications - COIA

COIA must be regarded as investments as, unlike other applications [1], they focus on the user and the processes have no longer the role of supplying information for the management of an organization, but are aiming to decrease the users' resource consumption.
COIA are no longer the property of an organization that uses them for the management of their decisions and to help the decision making process, they are aiming the citizens and solve their problems. The profit is obtained through the increase of the efficiency of the activity for the citizen that uses the application.
The main characteristics of COIA are the followings:

- COIA are always available;
- develop correct and complete activities;
- provide real time resources allocation;
- have a very large number of users, heterogeneous, regarding the level of adaptation to the human-computer interaction [2].

The development of COIA is made through the following steps:
S_1 – establishing the target group;
S_2 – establishing the problems that must be solved by the members of the target group;
S_3 – selecting the problems that have applicability on the market given by the absence of other applications that treat the considered problem or by the differences that are identified opposed to the existing applications;
S_4 – defining the problems in detail;
S_5 – elaborating the specifications;
S_6 – building a diverse set of solutions and the choosing of the most commendable one;

S_7 – the development of the software product;
S_8 – testing and correcting;
S_9 – implementation and assisted use.
These steps correspond to the development cycle. The lifecycle of COIA contains all steps of the development cycle and, additionally, the following steps:
S_{10} – current use of COIA;
S_{11} – current maintenance of COIA;
S_{12} – maintenance generated by changes in data input, data structures and the processing algorithms;
S_{13} – applied reengineering that consists of the redefinition of the problem, reconsideration of the target group and the realization of new solutions as well as the choosing of a suitable solution that contains new technologies, the implementation being made through the re-design of the best components from the old application, according to the requirements of the new technologies;
S_{14} – the use of the application after the reengineering process;
S_{15} – removal from use of the old version of the application due to the costs or reduction of the target group.
The lifecycle, but also the development cycle must be regarded from the point of view of the expenses of the users, developers and citizens.
The development of the citizen oriented informatics applications [3] must be done so that certain levels of the quality characteristic are met. Any other approach will lead to waste of resources if the required quality levels are not obtained. Let *Q* be the set of quality characteristics

$$Q = \{q_1, q_2, \dots, q_n\}$$

and *W* the set of weights associated to them in the calculus of the aggregated quality indicator

$$W = \{w_1, w_2, \dots, w_n\}$$
$$w_i \in (0,1), i = \overline{1,n}$$
$$\sum_{i=1}^{n} w_i = 1$$

Let *L* be the set of the levels associated to the quality characteristics from *Q*

$$L = \{l_1, l_2, \dots, l_n\}$$

The aggregated quality indicator, *AQI*, is given by the formula

$$AQI = \sum_{i=1}^{n} w_i * l_i$$

In the ideal case *AQI* has the value 1. For this to happen the levels of all quality characteristics implied in the process must be maximized.
Let *C* be the set of costs associated to the maximization of the quality characteristics from set *Q*

$$C = \{c_1, c_2, \dots, c_n\}$$

and *M* the set of the minimum accepted levels associated to the quality characteristics from set *Q*

$$M = \{m_1, m_2, \dots, m_n\}$$
$$m_i \in [0,1], i = \overline{1,n}$$

In these conditions, the minimum aggregated quality level *AQIM*, is given by the formula:

$$AQIM = \sum_{i=1}^{n} w_i * m_i$$

and the associated costs, *TC*, are

$$TC = \sum_{i=1}^{n} c_i * m_i$$

In order to obtain the highest quality levels with the lowest costs, resources are allocated to the increase of the levels of those characteristics for which the cost-weight report in the total quality is the greatest. The priority level of the characteristics, *PL*, is given by

$$PL = \max_i \{\frac{c_i}{w_i}\}$$

Generally, the costs are not constant regardless the level of the considered characteristic [4]. The cost generated by the increase with one percent of the level of a characteristic increases with the level of the characteristic. The cost, c_i, is therefore function of the actual level of the considered characteristic:

$$c_i = f(l_i), i = \overline{1,n}$$

In these conditions the maximization of the aggregated quality indicator *AQI* and the minimization of the total cost *TC* are wished and thus, the solving of the equations system

$$\begin{cases} \max AQI \\ \min TC \end{cases}$$

For users the quality of the application must have some minimum levels. These minimum levels are also valid for each characteristic. If the level of the aggregated quality indicator or level of any characteristic doesn't reach the minimum level, the users orient themselves towards other applications that solve the same problem and fulfill the quality requirements.

The costs of the exploitation of the citizen oriented informatics applications have two components: the cost that the citizens pay and the costs supported by the owner of the application [5].

The costs supported by citizens are zero in most cases. These costs are different from zero only in the case when the application has processes that require the purchase of third-party resources.

The costs supported by the owner of the application include the costs associated to the web server the application is running on, the costs of data protection and the maintenance.

The maintenance of the citizen oriented informatics applications is a process that takes place often, as the application must reflect all the modifications of the environment. Even if is a process with a relatively high frequency, the costs are low if the development produced a modular application and the code is well documented.

The reengineering of the citizen oriented informatics applications is made when the application does not correspond in a very high measure to the users' requirements and the costs of the reengineering process are lower than the costs associated to the maintenance process in the last two years and the maintenance cost estimated for the incoming year:

$$CR < CM_{cy-1} + CM_{cy} + CM_{cy+1}$$

where:

CR - the costs implied by the reengineering process;

CM_{cy} - the costs of the maintenance process in the current year;

cy-1 - the previous year of the one the evolution is made in;

cy - the year the evaluation process is done in;
cy+1 - the forecasted level of the maintenance costs for the incoming year on the basis of historical data.

The removal from use of the citizen oriented informatics applications assumes the limitation of access to the application and it's closing when there are no more active users. The users that already started the process of problem solving must be allowed to continue, but no more users will be accepted. When all users with ongoing problem solving processes have finalized the process, or a certain time interval expired, the application is closed and all data are archived.

The total cost of the citizen oriented informatics applications is given by the sum of costs associated to the development, exploitation, maintenance, reengineering and removal from use.

2. Typologies of costs for citizen oriented informatics applications

Deterministic models assume a complex approach with a reduced number of factors. The addition algorithm, the use of a simple device or simple equipment assumes also a simple approach, in which the deterministic elements are essential.

There are phenomena, processes, very complex organisms of which integral knowledge assumes the considering of a very large number of factors, impossible to realize, measure and aggregate to fundament a decision in a deterministic way. That is why work is done with a reduced number of factors, a reduced number of variables, of measured levels, with incomplete information.

Only properties from the generic plain will be known: events with the same probability, linear variation, not linear variation, discreet approach or continuous variables.

In practice, there are many types of costs:

- the estimated cost determined by the complexity of the informatics application that is realized;
- the calculated cost given by the total expenses needed for the development of the application;
- the post-calculus cost is the one that contains all expenses made during the application's development and putting the application in use.

Let us consider the informatics applications AI_1, AI_2, ..., AI_i, ..., AI_n, that have the measured complexities $K_1, K_2, ..., K_i, ..., K_n$ and the effective costs $C_1, C_2, ..., C_i, ..., C_n$. For the cost estimation models are built:

- linear type: $C = a * K+b$;
- exponential type: $C=a * e^K$;
- polynomial type: $C=a * K^2 + b * K + d$;

where a, b, d are the coefficients that must be estimated.

At the design of the informatics applications, these are fit in complexity classes and the estimation of the cost of the application is done. The estimate cost assumes the existing of hypothesis regarding:

- the number of components probable that are used to build a product;
- the typologies of applications;
- the probable durations of realization;
- the number of probable persons that participate to the realization of the product or service;
- the probable cost of the raw materials;
- the hourly wages probable to be paid.

In all cases, the estimated cost gives an insight about the level of expenses that will be incurred in the future, without claiming that they will coincide exactly.

In developing models for estimated costs, the following procedure is applied:

- the problems that need to be solved are given;
- a class of problems that the issue belongs to is identified;

- establishing the computer applications oriented towards the citizen that solved problems in that corresponding class;
- estimating the time periods are selected, that are mandatory for developing those applications and the probable, minimum, maximum or the average duration in which the new application will be ready;
- the estimated structures are identified, for the applications to which the problem *P* belongs and the hypothesis that a certain structure is fitted to the type of the application that will be developed is accepted;
- it is assessed that this new application has a similar degree of complexity to other application that are already in use;
- analyzing the quality levels of the applications and it is agreed whether to maintain the existing level of quality or the quality should be increased.

In these conditions there are enough prerequisites to define:
NL – the number of lines required to finish the project;
NM – the number of modules required to build the structure of the application;
P_i – the weights of the stages during the development cycle with the expected finish duration
$P_i \in [0,1]$, where $\sum_{i=1}^{k} P_i = 1$, k is the number of stages of the development cycle;
D_i – expected duration for finishing the application;
CP – the estimated complexity of COIA;
NP_i – the number of persons required to complete each stage, $NP_i = f(CP, D_i)$.
Little by little the elements necessary to obtain a possible structure of expenditure is constituted:

Table 1. *The structure of expenditure*

Chapter	Year I	Year II	Total	Expenditure Unit	Expenditure Total Chapter
Number of analysts	α_{11}	α_{12}	α_{13}	C_1	T_1
Number of programs	α_{21}	α_{22}	α_{23}	C_2	T_2
Number of computers	α_{31}	α_{32}	α_{33}	C_3	T_3
Number of testers	α_{41}	α_{42}	α_{43}	C_4	T_4
Number of implementations	α_{51}	α_{52}	α_{53}	C_5	T_5
TOTAL	$T_1=\sum_{i=1}^{5}\alpha_{i1}$	$T_2=\sum_{i=1}^{5}\alpha_{i2}$	$T_3=\sum_{i=1}^{5}\alpha_{i3}$	$T_4=\sum_{i=1}^{5}C_i$	$TT=\sum_{i=1}^{4}T_i$

TT represents the estimated cost of the project. The COIA application does not exist physically, it exists only as a schema in which there are pointed out the necessary objectives, to whom it is addressed and whether or not makes a profit. The work assumes the characteristics of information, cost structures and it is based on the optimistic or pessimistic assumption. It must be clearly defined what optimistic and pessimistic assumption means.
The real cost of COIA is given by the sum of expenses made during the development cycle. If *NX* represents a number of documents for which the payments were made, the real cost *CR*, is computed using the formula:

$$CR = \sum_{i=1}^{NX} PL_i$$

The real cost is rigorously estimated. There are situations when:

$$TT = CR$$

In building the estimated cost, there were taking into account feasible hypotheses, possible levels obtained with high accuracy were selected, also the unit expenditure were determined with a good accuracy.

This situation is purely coincidental and if appears should be analyzed in order not to be considered the result of a failure.

The situation in which $CR < TT$ appears when the developer, because of too much prudence:

- overestimates the completion interval;
- overestimates the team;
- overestimates the level of unit expenditure,

The difference $\Delta = TT - CR$ must not be interpreted as a cost saving, the risks assumed must be analyzed when the overestimation occurred.

If $CR > TT$ then the probable levels were underestimated and the difference $\Delta = CR - TT$ must not be treated as a loss. In both situations it is recommended that all estimated and real costs to form data series that must be analyzed. Models are built to make corrections to the estimated costs.

Table 2. *Cost correction*

Application	Real costs	Estimated costs
A_1	CR_1	TT_1
A_2	CR_2	TT_2
...	...	...
A_i	CR_i	TT_i
...	...	...
A_n	CR_n	TT_n

$$\hat{CR} = a*TT + b$$

$$TT_{corrected} = \hat{CR}$$

Also, the expenditure structure will be analyzed and the stages will be completed backwards in order to obtain $\hat{\Gamma}_i$, i = 1, ...; $\hat{\alpha}_1$ and $\hat{\alpha}_2$; $\hat{\beta}_1$ and $\hat{\beta}_2$; $\hat{\gamma}_1$ and $\hat{\gamma}_2$; $\hat{\delta}_1$ and $\hat{\delta}_2$; $\hat{\theta}_1$ and $\hat{\theta}_2$, corrected values.

The expenses involved are estimated using estimation models. The costs regarding resources, implementation, execution as well as the costs regarding the increase in quality of the application are very closely observed.

The estimation models are flexible structures, for each application starting from the achieving principles. The used technique for modeling is oriented towards increasing the applications' efficiency at the users' level.

The total costs of COIA consist of the costs for each stage, respectively:

- cost of achieving is the cost involved in the application development;
- cost of usage consists of the needed resources in order that the application complies from the functional point of view;
- cost of maintenance determines the financial implications required for maintaining and improving the functional qualities of the application;

- cost of reengineering defines the needed resources for reuse of the components in order to update the functions of the application.

The necessary condition for making the costs more efficient, imposed by COIA is that they must be balanced, that is composed of elements which cost is consistent with the importance of the life cycle of the COIA [6]. The importance of this stage is defined and characterized by the users' degree of satisfaction.

Determining the costs necessary for the development of COIA increases the efficiency of the usage of applications and also defines advanced working criteria. Maintaining low costs during construction and usage phases ensures focusing applications towards citizens.

3. The quality of the estimation models used for validation

The following models are built for estimation purposes:

M_1 – estimation in conditions of common implementation, testing and use;
M_2 – cost estimation in an optimistic view, with no unforeseen events;
M_3 – cost estimation in a pessimistic view, with additional expenses;
M_4 – estimation of costs, expenditure and consumption;
M_5 – estimation of resource requirements considering an average level of consumption;
M_6 – estimation of resource requirements considering a high level of consumption;
M_7 – estimation of resource requirements considering a low level of consumption.

Considering the following notations:

$A_1, A_2, ..., A_i, ..., A_n$ the n executed applications that are currently in use;
$CR_1, CR_2, ..., CR_i, ...CR_n$ the real costs of the n applications;
$K_1, K_2, ..., K_i, ..., K_n$ the application costs estimated for normal conditions;
$L_1, L_2, ..., L_i, ..., L_n$ the application costs estimated for optimal hypothesis;
$H_1, H_2, ..., H_i, ..., H_n$ the application costs estimated for pessimistic hypothesis;
$I_1, I_2, ..., I_i, ..., I_n$ the application costs estimated for average levels.

Using the above indicators, we get the following correlation table:

Table 3. *Cost model correlation*

Application	**CR**	**M_1**	**M_2**	**M_3**	**M_4**
A_1	CR_1	K_1	L_1	H_1	I_1
A_2	CR_2	K_2	L_2	H_2	I_2
...	...	...	...	...	...
A_i	CR_i	K_i	L_i	H_i	I_i
A_n	CR_n	K_n	L_n	H_n	I_n

From the above table we obtain the following sums:

$$S_1=\sum_{i=1}^{n}(K_i-CR_i)^2$$

$$S_2=\sum_{i=1}^{n}(L_i-CR_i)^2$$

$$S_3=\sum_{i=1}^{n}(H_i-CR_i)^2$$

$$S_4=\sum_{i=1}^{n}(I_i-CR_i)^2$$

Considering $S_x = min\{S_1, S_2, S_3, S_4\}$, the corresponding M_x is the optimal cost estimation model for the selected applications. If M_x is evaluated by all the involved parties as being the best model, it will be validated and used exclusively. Estimation models are used in order to get a general view of the computer applications currently developed. This way development costs and schedules can be identified.

4. Costs associated with the re-engineering process

The re-engineering process is a component of the life cycle of a computer application. Its purpose is to extend the operating period of the application.
Re-engineering occurs in the following situations:

- There are profound modifications in the problem addressed by the application and the maintenance costs become so high that it is more efficient to extract the performing components from the old application and integrate them in a new computer application that is fundamentally restructured; this application would thus contain the old components that were selected and new processing functions [7];
- New technologies are developed for the problem in question, totally different from the ones used by the initial computer application; the solution offered by the existing application and the solution offered by the newly built applications differ significantly, the losses incurred by the old application become prohibitive and thus a major change of the existing application is required; the invoice creation process for a store's products implies a data input; an old application for example, would require providing data about the clients, about the providers and about the sold products; the functionality of the application is limited to calculus and centralizing the sums obtained; this solution is imposed by the existing software which does not contain a database management system and by the existing hardware which has a very limited storage capacity; with the development of software – database management systems, of hardware – faster processing speeds and increased storage capacity, and the improvement in data acquisition methods, the invoicing problem changes radically and a re-engineering process is required because:
    - The database will store data about the sold products;
    - The products will be identified by reading the barcodes;
    - Data is being sent to the client's bank using electronic signature;
    - Electronic payment is used;

The application is integrated in the ERP solution used by the company. New application means:

- different database from what already exists;
- different information flows from already exists;
- new interfaces totally different from the ones from the old application;
- high acquisition, deployment, utilization costs which are given by the operators training;
- purchase of hardware compatible with the minimum requirements and software acquisition licensed for the operating system and data acquisition.

Reengineering retain elements of the old application that softens the impact suffered by the classic users when the application resulting in the reengineering process brings new items.
Reengineering ensure that new application completes, using new technologies, what the old application effectuated, but also many other new functions.
For the e-learning application old lessons offered solutions already established by the business, leaving for the professors fewer situations to show their personality and take the training lessons to different level from class to class. In these cases, the professor:

- presents the matter;

- uses for exemplification drawings, animation, films from the old e-learning application, on the structure imposed by the developer;
- proposes problems from the implemented the, letting the students to give different values to variables such as lengths, angles, volumes, density and to follow the structure imposed by the developer for the steps of the problem.

The old application offers the professor the opportunity to check the following aspects:

- if the students enter the data;
- if the students go through the steps imposed by the solution proposed by the developer;
- if all pupils have fully realized the requirements.

An e-learning application resulted from reengineering:

- gets the entire functionality of the old applications;
- includes procedures through which the professor defines its own problems and proposes them to students for solving;
- studies the lessons in order to be consistent with the class;
- allows the professor to interact more with the students, by customizing the functions from case to case;
- builds its own lessons using digital images, animation, film, sound, in a standard laboratory, but affordable regarding the cost. The new application also has management components for lessons, which are in the developer's original application form.

The lesson management component available to the professors by the simplicity of the instruments with which are used the animation, the sounds, the images. The management component for building own lessons must be made to take into account that addresses to people with computer training at the level of the users of the products.

The manner of building the structures of their own lessons must use templates, parameterization which facilitates the professor's work - developer. Furthermore, the professor - developer should have access to the lessons' components offered by the application's developers to integrate them into its own lessons.

Re-engineering steps follow the aspects:

- reconfirm the target group taking into consideration the new elements, thus broadening the target group;
- defining the new problem P' that includes the original problem P, adding new features;
- the effort to get the same results must be much smaller than using the old application;
- the period of settlement must be much lower;
- level of flexibility for alternatives solutions must be enlarged;
- must be adapted to the characteristics of each user, the interface being restructured according to user requirements, in the case of a electronic store, each user has a history of purchases and the his interface is dependent on the frequency with which he purchases; it is assigned username and password and generates user-specific interface depending on purchase history.

Reengineering costs are related to:

- the cost of redesigning the solutions following the steps of the application development cycle stages through reengineering;
- costs of implementation;
- costs of adapting to new user features;
- acquisition costs of software and hardware that allow the implementation and improvement of the application's functions.

Reengineering costs must be fully recovered in the using period because the reengineering is seen as an investment, not as product maintenance. It is important that the application's life estimation to be as rigorous as possible. In reengineering case, the new application uses the old application

database, and they are subject to process reengineering, in terms of facilities and not in terms of digital content.

5. Conclusions

Implementation of cost models [8] increases satisfaction for customers that use informatics applications. The cost management and building a balanced cost contribute to an efficient way of working with citizen-oriented computer applications, the citizen being the main beneficiary of the functions implemented.
COIA increase applications efficiency by identifying and correcting existing deficiencies in the facilities provided. The user, being the main recipient of the applications, is defined as the main filter in evaluating applications.
Using re-engineering provides optimization for components that do not meet performance criteria. It is thus ensured cost control by improving the application involved. The entire process of building the COIA intends to improve interaction with the final beneficiary, and building an accessible interface, regardless of their specialization.
In order to study COIA costs are created databases that store data regarding applications and their costs, based on this records are built models with the optimal level of complexity, so that the difference between the estimated costs and actual costs to be bearable.

Acknowledgement
This article is a result of the project POSDRU/6/1.5/S/11 „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co-funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies, project no. 7832, Doctoral Program and PhD Students in the education research and innovation triangle, DOC-ECI. This work was co-financed from the European Social Fund through Sectorial Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213, „Ph.D. for a career in interdisciplinary economic research at the European standards" (DOCCENT).

6. References

[1] TOMOZEI, C., Distributed Application Based On Reengineering For 2sls Regression Analysis, Proceedings of the 34th Annual Congress of the Romanian American Academy of Arts and Sciences ARA, Information Technology Section, pg. 433-437, May 18-23 2010.
[2] IVAN, I., CIUREA, C., MILODIN, D., Collaborative Educational System Analysis and Assessment, Proceedings of The Third International Conferences on Advances in Computer-Human Interactions, ACHI 2010, February 10-16, 2010, Saint Maarten, Netherlands Antilles.
[3] IVAN, I., VINTILA, B., CIUREA, C., DOINEA, M., The Modern Development Cycle of Citizen Oriented Applications, Studies in Informatics and Control, Vol. 18, No. 3, 2009.
[4] PALAGHITA, D., VINTILA, B., The Analysis of Informatics Security Costs in Citizen Oriented Applications, Journal of Information Systems & Operations Management, pp. 54-66, 2010.
[5] IVAN, I., PALAGHITA, D., VINTILA, B., Security Cost Analysis of Citizen Oriented Applications, Proceedings of the 5th International Conference on Business Excellence, 15-16 October 2010, Brasov, Romania, pp. 243-246.
[6] IVAN, I., VINTILA, B., PALAGHITA, D., Types Of Citizen Orientated Informatics Applications, Open Education Journal, Russia, No. 6, 2009.
[7] IVAN, I., SACUIU, L., MILODIN, D., CIUREA, C., Metode de integrare a aplicatiilor informatice distribuite, Revista Română de Informatică şi Automatică, Vol. 20, No. 1, 2010.
[8] IVAN, I., VIŞOIU, A., Baza de modele economice, Editura ASE, Bucureşti, 2005.