THE ANALYSIS OF INFORMATICS SECURITY COSTS IN CITIZEN ORIENTED APPLICATIONS

Dragos Palaghita[1]
Bogdan Vintila[2]

Abstract

The paper highlights the analysis of informatics security costs for the citizen oriented applications. The citizen oriented informatics applications are defined. The differences brought by these when compared with the traditional applications are described. Structures of citizen oriented informatics applications are presented. A few common citizen oriented applications are discussed. The special security requirements of the citizen oriented applications are discussed. Ways of increasing the security of the applications are given.

Key words: security, cost, estimation, citizen orientation, distributed applications.

1. Citizen oriented informatics applications

In the context of the knowledge based society and of higher citizen requirements the appearance of a new category of informatics applications is necessary. The citizen orientated applications bring a new orientation as the citizen is considered to be the central element. These are different by the classic applications through:

- these are developed to solve the problems of the citizens, not the problems of the organization for which are developed;
- the target group is very large and very divers being formed by all the citizens;
- the applications are always available online;
- the citizen oriented applications aren't dependent on the hardware or software platform;
- the cost of use is very low or null;
- the quality requirements are much more strict than for traditional applications;
- localization assumes having the dialog with the user in his own language;
- the use of the applications doesn't assume previous training of the users [1];
- are very often updated to reflect the changes in the environment;
- adaptation to offer the citizens a greater degree of satisfaction.

The structure of the citizen oriented informatics applications differ on the offered functionality and the domain they are created for. The citizen oriented informatics applications are with:

- simple linear structure; these are applications that, for problem solving, assume the following of a number of steps, in a preset order, without the possibility to go back to a previous step; the first step of the sequence starts up the processing and the last one returns the results for another application; for an informing

[1] Dragos Palaghita, University of Economics, Bucharest, Romania, email: mail@dragospalaghita.ro

[2] Bogdan Vintila, University of Economics, Bucharest, Romania, email: vb@vintilabogdan.ro

application, the information is structured in a logic sequence of the steps that must be followed;
- linear structure and simple links between components; these assume the possibility of going back to the previous steps; these are applications for which the possibility of modifying data from the previous steps or repeating them is a must; for the applications with a high number of steps is inacceptable for the user to redo all the steps just because he made something wrong in the end;
- linear structure and multiple links; assumes the existence of links between components and the navigation is made between any of the connected components respecting limitations imposed for the correct functioning of the application; the navigation towards a step is not allowed without the fulfillment of the prerequisites; for the informing applications of this type, the navigation has no restrictions beside the logical links;
- tree structure and simple links; these are applications for which from a step the user can move in many directions; the simple links between modules allow the advancement only on vertical as the user is going away from the tree root; these type of applications is suitable for showing information on the basis of selection criteria;
- tree structure and double links; these assume the existence of bidirectional links between the components to browse the tree structure both top-bottom and bottom-top; double links ensure the possibility to go back to previous steps;
- tree structure and multiple links; the pass from a component to another is made only in the limit of the good functioning given by the logic of the processing which the applications make; the tree structure with multiple links is the most complex of them all; this allows the development of complex citizen oriented applications.

The applications for virtual campus training must satisfy the requirements of the persons that access the educational system. For this, these must be flexible, maintenance free, secure, accessible, platform free, without additional costs, always available, adaptable.

Considering the very dynamic character of the virtual campus training domain, the requirements of the users quickly change and the applications, in order to be competitive, must evolve to fulfill them.

E-commerce is a very popular form of commerce as it has some clear advantages on the traditional commerce:
- no more stocks; virtual stores don’t have stocks, or, if they do, these are very small [2];
- geographic borders are eliminated;
- very low running costs;
- users’ comfort;
- automation.

The e-governing applications are used in the relation of the state with the citizens for solving different situations in which they are partners [3]. Issue of certificates and forms is made automatically. The e-voting applications are also very used in the e-governing process. These must be accessible from as many geographical points as possible.

Informing applications are those that guide the users to obtain information regarding a certain domain, state, process, object, phenomena. This type of application must be characterized by a clear structure allowing the user to reach, in as few steps as possible, to the desired result. The informing applications must not have using costs.

2. Security requirements for the citizen oriented applications

Citizen oriented informatics applications differ from the other informatics applications, determining a series of particularities because [4]:

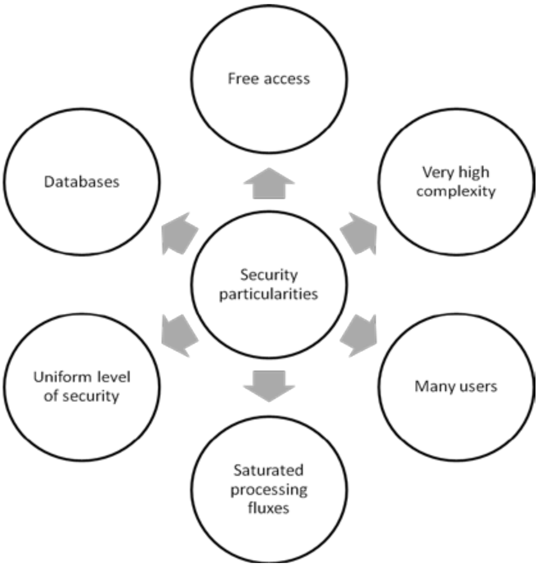

Figure 1 – Security particularities for citizen oriented informatics applications

- the applications are freely accessed because there are no more geographical borders and the users must be able to access the applications anytime without additional costs;
- the complexity of the citizen oriented applications is very high as these are distributed applications and their development overwhelms the classic approach;
- the users are many and diverse because the citizen oriented informatics applications reach their aim only if they are used by many persons;
- the processing fluxes are saturated given the large number of users and the complexity of the applications;
- the security level is uniform for all the branches of the tree associated to the application and there are no security breaches;
- the management of the database and the files of the application is made through procedures that exclude processing incidents.

The security of the application is tightly connected with its quality. The quality characteristics are those that influence in a decisive way the security. The most important quality characteristics from the security's point of view are:

- reliability;

- maintainability;
- portability;
- integration.

If, for classic applications these properties are not very important, for the citizen oriented informatics applications these have a special importance.
To ensure security, during the development cycle, for each component a special step is made. During this step the implementation of security techniques and technologies that are suitable for the considered component is made [5].

The increase in security is made through:

- inserting components that eliminate vulnerabilities;
- inserting authentications; is made to give rights only to the users that prove they are part of the system;
- data restrictions are used to protect the data the application works with from the bad intended users and from the users without access rights;
- updates only by adding information are made to protect the data from the attacks made by system's users;
- developing software components that cover better the vulnerabilities through the training of the developers and awareness of the most frequent risks.

Increasing the security of the applications above a certain level is justified only if the costs implied by its lack are higher than the costs for implementation. Figure 1 highlights the main security particularities of the citizen oriented informatics applications.

3. Informatics security influence factors

Direct influence factors consist of:

- the target group which is defined as all the individuals that form the collectivity which uses the informatics product; the target group influences security directly through:
    - structural diversity, a collectivity structural analysis is necessary to determine behavioral patterns differenced based on age, sex and education in order for the security system to register user actions and assign a behavioral pattern to application users such that an adaptive security policy system is used to grant or deny privileges to them;
    - dimension such that the security system is correlated to the number of individuals that access the application; this way the security system will work at optimal parameters;
    - the social status in the collectivity, thus if it proves to be true that certain individuals in it are against actions or thoughts that the application owner sees as favorable, a greater amount of effort must be made to ensure an increase in physical and logical security of the application;
- the development process quality has a direct influence on informatics security because:
    - a high level of quality leads to the minimizing the number of defects which in turn reduces the informatics security risk;

  - a low level of quality increases the number of vulnerabilities in the application thus increasing the informatics security risk [ALHA08];
- in the development cycle of the security system fixed quality objectives should be followed:
  - homogeneity of source code by developing modules and procedures which integrate totally in the security system;
  - intelligibility of implemented procedures in order to minimize testing, optimization and maintenance time of the security system associated to the informatics application;
  - flexibility of network communication and reporting systems in n order to function with an extended set of report formats thus assuring a high compatibility degree with intrusion detection systems;
  - scalability of components in order to easily increase the adaptability of the security system;
- used development technologies represent an important aspect because they influence the level of informatics security by:
  - quality transfer, if the instruments used in the development stage have a high quality level then by using them the developed security system will benefit of a high quality level;
  - the degree to which the development assistance tools help the developer make good decisions by providing useful observations at development time;
  - the novelty degree of used instruments and tools and their coverage level of the newest informatics attacks thus allowing the developer to bring the performances of the security system to the highest standards;
- the environment in which the informatics product is used and in which the security system activates influences the level of security by the degree of provided physical security;
- hardware elements have a direct influence on the security system by their wear resistance and reliability considering they have to work continuously; performance is another key issue for hardware equipment being necessary to ensure a small response time for each event in the security system;
- dynamic elements of the problem that the informatics application needs to solve, this implies an increased flexibility level to handle new and unforeseen events generated by structural or logical changes in informational transfers required by modifications in the problem structure.

The indirect factors that determine security are:

- complexity which has an important effect over informatics security, according to as the software product's complexity grows so does the number of defects thus decreasing the level of informatics security. Complexity is defined using the following models:
  - Halstead which is characterized according to by the following equations:
    - the length of code *N* which is represented by the sum of the operator number N_1 and operands number N_2*:*

$$N = n_1 * \log_2 n_1 + n_2 * \log_2 n_2$$

where:
n1 represents the number of distinct operators;
n2 represents the number of distinct operands;

- volume is the product of the code length with the minimum number of bits needed to store operators and operands:

  $$V = N * \log_2 n$$

  where:

  $N = N_1 + N_2$

  $n = n_1 + n_2$

- difficulty is defined as:

  $$D = \frac{n_1 * N_2}{2n_2}$$

- the effort for implementing the program is computed using:

  $E = D * V$

o cyclomatic complexity is defined by:
C = m – n + 2
where:
m is the number of arcs in the graph associated to the program;
n is the number of nodes of the graph associated to the program;

- the application affects the target group through the content displayed as part of some individuals who are inclined to cause damage to corporate computer application for their own benefit, if the application contains sensible files;
- IT application developer experience helps decrease or increase the quality of source texts thus making a decisive contribution to the quality of information security;
- how to install and implement computer application is important because it ensures smooth operation and security of the system test parameters, if the installation procedures are carried out incorrectly then decreases quality information security behavior leading to unexpected results of computer system security.

Figure 2 presents the graphical representation of direct and indirect factors over informatics security.

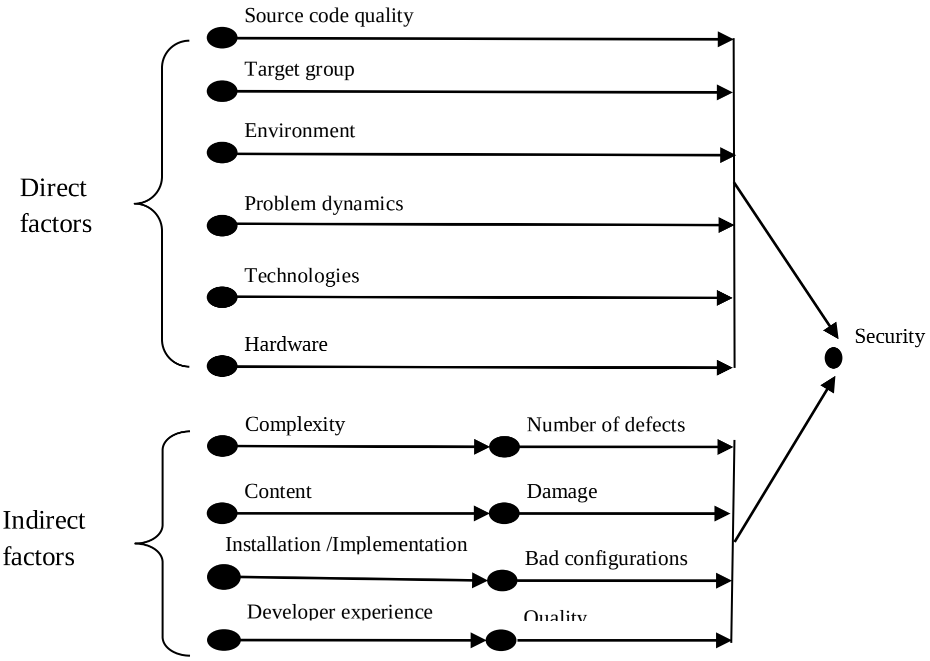

Figure 2 - Graphical representation of influence factors

Influence factors are an important element in information security analysis and the weighting of cost models. Figure 3 is the connection between influence factors and quality characteristics of the source texts that are used for the computer security systems.

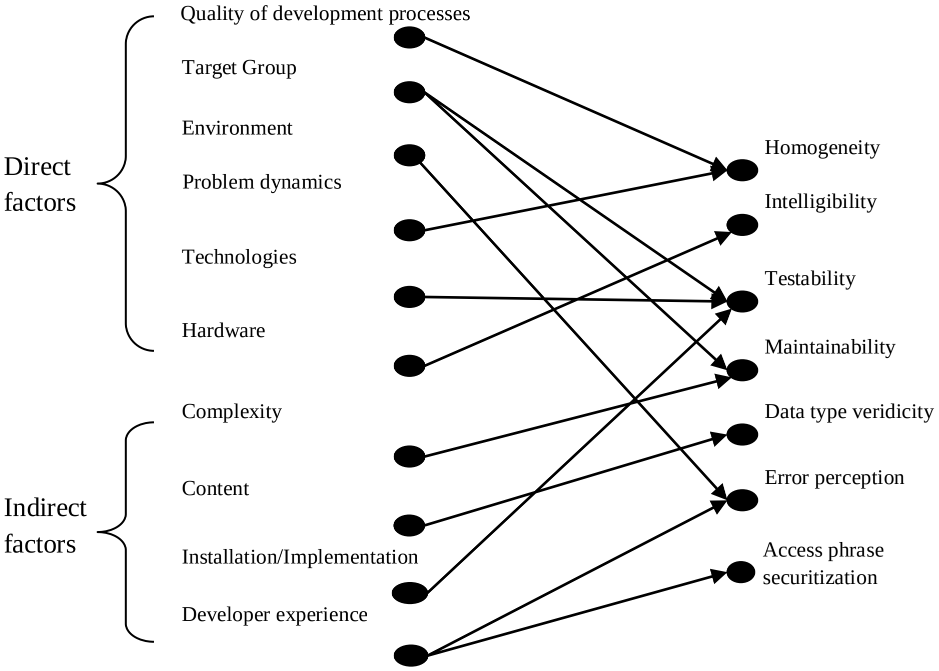

Figure 3 - Influence factor related to quality characteristics

In Figure 4 the influence on quality characteristics of interaction processes associated with computer software is presented.

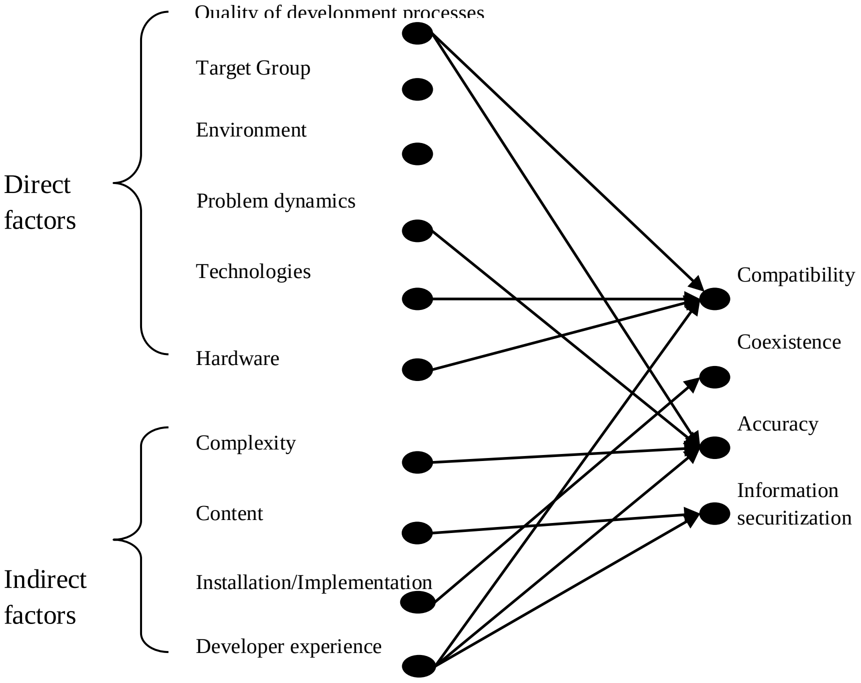

Figure 4 - Factor influence on quality characteristics related to the interaction with the software product

In Figure 5 the influence on quality characteristics of interaction with the user computer system security is presented.

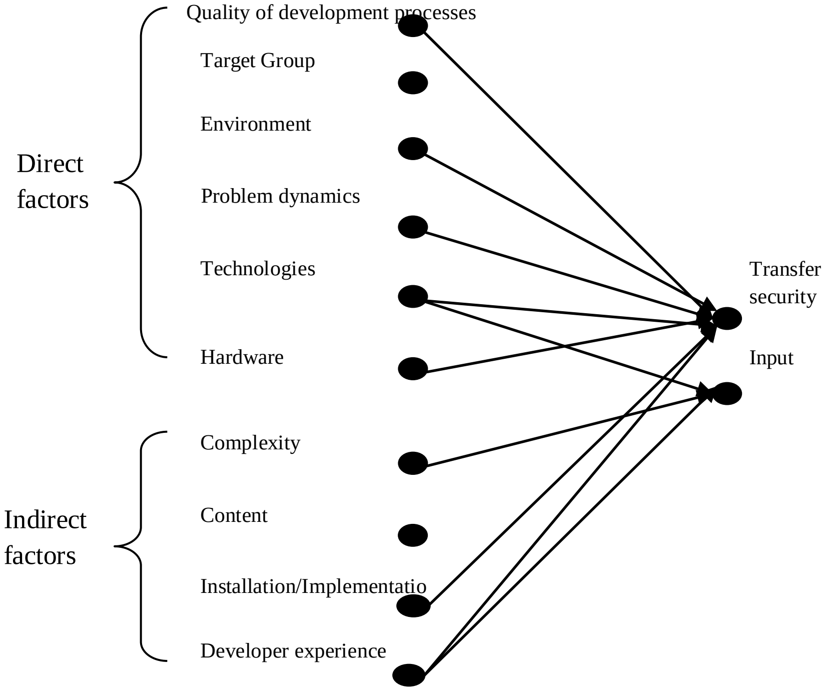

Figure 5 - Factor influence on quality characteristics related to user interaction

Quality characteristics of the security system are influenced positively or negatively depending on the influence factors. There is a direct proportional relationship between the influence factors and quality characteristics of the security system.

4. Using risk estimation towards cost modeling

In order to determine the risk involved in distributed application's security the costs caused vulnerabilities must be defined.

Errors are the root causes of software defects in order to obtain a thorough analysis of vulnerabilities error prediction must be analyzed.

Considering the error set SE = $\{E_1, E_2, \ldots, E_{NE}\}$ the compensations are quantified in Table 1.

Table 1 - Data structure for computing compensation expenses.

Ch_{DESP}	Error	Error frequency	ChE
Ch_{DESP1}	E_1	f_1	ChE_1
Ch_{DESP2}	E_2	f_2	ChE_2
…	…	…	…
Ch_{DESPi}	E_i	f_i	ChE_i
…	…	…	…
Ch_{DESPn}	E_{NE}	f_{NE}	ChE_{NE}

where:

ChE_i – costs caused by error E_i

E_i – error i;

f_i – frequency of error i;
NE –maximum error number.

The appearance rate of an error is estimated using:

$$pe_i = \frac{fi}{\sum_{j=1}^{NE} f_j}$$

The model for computing expenses is:

$$Ch_{DESP} = pe_1 * ChE_1 + pe_2 * ChE_2 + ... + pe_{NE} * ChE_{NE}$$

where:

pe_i – the probability of error E_i appearance;

Risk analysis is based on vulnerability estimation and mitigation, according to [6] the risk estimation is a five stage process. In **stage 1** there are two models:

A. The vulnerability model, formed from the set of vulnerabilities $SV = \{V_1, V_2, …, V_n\}$ which has associated the set of probabilities $PSV = \{PV_1, PV_2, …, PV_n\}$. Where V_i represents the vulnerability *i*, and PV_i is probability that the vulnerability was being exploited.

B. The threat model is composed of a set of threats $SA = \{A_1, A_2, ..., A_m\}$, with has associated a set of probabilities= $\{PA_1, PA_2, …,PA_m\}$. Where A_i is the threat *i*, and PA_i represents the probability of the threat to quantify.

Stage 2 has been materialized in the development of the goods model which is represented by all the goods used by the application $SB = \{EB_1, E B_2, …, EB_k\}$.

Stage 3 represents the analysis of risks occurrence probability. It is considered that a threat exploits a set of vulnerabilities, each vulnerability of the set has an exploit probability PV_i, thus a matrix of perceived risk is developed by adapting the model described by [3] adding the consideration of threats at the vulnerability collectivity level given the vulnerability set $SV=\{SV_1, SV_2, … SV_v\}$. The exploit probability of one or more vulnerabilities *i* from set SV in order to get access to asset EB*j* is determined:

$$PSV_i EB_j = \prod_{k=1}^{v} PV_{ik} * EB_j$$

Where: PV_{ik} is the exploit probability of vulnerability *k* from vulnerability set *i* by a threat A_l. **Stage 4,** combating the risk, consists in the use of countermeasures to minimize the threat. Thus each threat A_i from the set *SA*, is associated with a set of reduction factors $SFD = \{FD_1, FD_2, …, FD_k\}$.

The probability of risk quantification through a set of vulnerabilities is determined using the formula below:

$$R = \sum_{i=1}^{m} \left[PA_i * \frac{\left(\sum_{j=1}^{k} \mathrm{PSV_i EB_j} \right)}{k} * FD_i \right] * \frac{1}{m}$$

Using the above formula a clearer picture is given over the quantitative aspects of risk. In Figure 1 the risk model analysis model is presented. **Stage 5** is needed for future risk analysis as this will speed up the process in the future based on gathered documentation.

Considering the vulnerability set SV = {V_1, V_2, ..., V_{NV}} the data structure presented in table 2 is used to determine the effective vulnerability costs encountered in a distributed application.

Table 2 - Data structure for analyzing the costs associated with vulnerabilities

Ch_{VUL}	Vulnerability	Vulnerability frequency	ChV	CH_pV
Ch_{VUL1}	V_1	fV_1	ChV_1	CH_pV_1
Ch_{VUL2}	V_2	fV_2	ChV_2	CH_pV_2
Ch_{VUL3}	V_3	fV_3	ChV_3	CH_pV_3
...	...	...	...	...
Ch_{VULi-1}	V_{i-1}	fV_{i-1}	ChV_{i-1}	CH_pV_{i-1}
Ch_{VULi}	V_i	fV_i	ChV_i	CH_pV_i
Ch_{VULi+1}	V_{i+1}	fV_{i+1}	ChV_{i+1}	CH_pV_{i+1}
...	...	...	...	...
$Ch_{VULNV-2}$	V_{NV-2}	fV_{NV-2}	ChV_{NV-2}	CH_pV_{NV-2}
$Ch_{VULNV-1}$	V_{NV-1}	fV_{NV-1}	ChV_{NV-1}	CH_pV_{NV-1}
Ch_{VULNV}	V_{NV}	fV_{NV}	ChV_{NV}	CH_pV_{NV}

where:
fV_i –the exploit frequency of vulnerability V_i
ChV_i – costs caused by the exploit of vulnerability V_i;
Ch_PV_i – programming effort for removing the vulnerability;
NV – maximum number of vulnerabilities.

In order to compute the rate pV_i of vulnerability V_i exploit the following formula is used:

$$pV_i = \frac{fV_i}{\sum_{j=1}^{NV} fV_j}$$

The formula for cost estimation considering vulnerabilities is defined as:

$$Ch_{VUL} = pV_1 * (ChV + Ch_PV_1) + pV_2 * (ChV_2 + Ch_PV_2) + ... + pV_{NV} * (ChV_{NV} + Ch_PV_{NV})$$

The security cost computation as described in [7] is defined by the following measure:

$$Ch_{T\sec} = Ch_P + Ch_{RBD} + Ch_D + CH_{PSS} + Ch_T + Ch_{DESP} + Ch_{OPT} + Ch_{VUL}$$

where:
Ch_P – expenses with the development team;
Ch_{RBD} – expenses caused by restoring the data base;
Ch_D – security system maintenance caused expenses;
Ch_{PSS} – expenses with designing the security system;

Ch_T – testing activities expenses;
Ch_{DESP} – compensation related expenses;
Ch_{OPT} – optimization related expenses;
Ch_{VUL} – vulnerability minimization expenses.

In order to improve the cost computation model the risk estimation must be considered in order to determine an aggregated measure.

In order to compute the security cost considering the risk factor the risk measure in step 4 of the risk model is used in conjunction with the security computation formula defined in [4] thus the integration of the risk estimation model into the security computation model is defined in the following measure:

$$GCh_{T\sec} = Ch_P + Ch_{RBD} + Ch_D * R + CH_{PSS} + Ch_T + Ch_{DESP} + Ch_{OPT} + Ch_{VUL} * R$$

Where GCh_{Tsec} is the global security cost computation including risk assessment for key areas like maintenance and vulnerabilities and R the risk estimation measure from Stage 4 of the risk model.

By including the risk estimation measure in the security cost computation the degree to which the measure of security cost meets the demands of a distributed application increase and provides more accuracy.

5. Conclusions

The citizen oriented informatics applications appear to satisfy the users' need of high performance problem solving tools and ease of use. These must have high quality characteristics to ensure flawless processing and also must ensure the access to the features for all users, regardless of their previous training. The citizen oriented informatics applications have many new requirements compared with the traditional applications. These came as the users evolve and need additional features. As the problems the citizens confront with are of great diversity, the citizen oriented informatics applications also have a wide range of structures. Each of these structures is suitable for a certain type of citizen oriented application. As these applications are accessed by a large number of users, the security requirements differ from the ones for the usual applications. For the citizen oriented informatics applications the security requirements are fulfilled by many methods. The integration of open source components is one of the most used methods to increase security. Security cost computation including risk elements provides a better standing model offering more insight into the effects of vulnerabilities on cost estimates of citizen oriented applications. Using these measures to improve cost computation is needed to provide a quantification of informatics security risks in citizen oriented applications.

6. Acknowledgements

This article is a result of the project „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 Program for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.

7. Bibliography

[1] C.Y. Yoon, "Measures of perceived end-user computing competency in an organizational computing environment," *Knowledge-Based Systems*, vol. 22, no. 6, pp. 471-476, Aug. 2009.

[2] L.J. Harrison-Walker, "The measurement of a market orientation and its impact on business performance ," *Journal of Quality Management*, vol. 6, no. 2, pp. 139-172, 2001.

[3] C.S. Ong and S.-W. Wang, "Managing citizen-initiated email contacts," *Government Information Quarterly*, vol. 26, no. 3, pp. 498-504, Jul. 2009.

[4] D. Mellado, E. Fernández-Medina, M. Piattini, "Towards security requirements management for software product lines: A security domain requirements engineering process ," *Computer Standards & Interfaces*, vol. 30, no. 6, pp. 361-371, Aug. 2008.

[5] C. Wolter, M. Menzel, A. Schaad, P. Miseldine, P. Miseldine, "Model-driven business process security requirement specification ," *Journal of Systems Architecture*, vol. 55, no. 4, pp. 211-223, Apr. 2009.

[6] D. Palaghita, B. Vintila, “Security Risk Analysis and the Security Need in Citizen Oriented Applications”, *Economy Informatics*, vol. 9, no. 1, pp. 79-86, Sept. 2009.

[7] I. Ivan, D. Palaghita, “The Informatics Security Cost of Distributed Applications”, *Theoretical and Applied Economics*, vol. 17, no. 1, pp 49-68, Jan. 2010.