Journal of Applied Business Information Systems

http://www.jabis.ro

Informatics Security Cost Factors in Web Applications

Dragos PALAGHITA*

* Academy of Economic Studies

Abstract: *Types of web applications are presented. User oriented and process oriented applications are presented stating characteristics and advantages of use. Informatics security is analyzed at source code, user interaction and interaction with the informatics application level. Influence factors related to informatics security are analyzed and organized considering direct influence factors and indirect influence ones. The AVIO application operation environment is analyzed and security influence factors are determined. The primary security cost generating factors are determined.*
Keywords: *security, cost, model, users, web applications*

1. Types of web applications

Web applications are becoming frequent in daily use due to their high availability degree and their complex processing capabilities. There are various types of web applications that are operational [1]:

- information application that operate through news feeds received from selected sources by using applications that aggregate them according to user needs; these applications are common on the internet allowing the user to receive information from a vast array of sources;
- image sorting using color palettes for identifying the images that containing certain shades or shades combinations; these applications are useful for extracting only a type of image or pattern from a collection of images;
- medical diagnostic applications that implement decision trees in order to diagnose users according to the answers they provide to the given questions; these applications base their diagnostics on prior knowledge from diagnosed patients in medical institutions by certified staff;
- e-government that enable an effective collaboration between state agencies and citizens through the implementation of online platforms for taxes payment or the management of state and citizen problems and responsibilities.

User oriented applications are applications that are centered on resolving user needs these applications have the following characteristics [2]:

- are easy to use implementing accessible interfaces that are explicit by nature; the most important feature of an easy to use interface is its intuitiveness meaning the degree to which the user understands its principles based in his prior experience with informatics applications;
- provide useful results in a timely manner by implementing efficient algorithms for data processing; this characteristic is important because information provided in the right amount of time to the user is valuable;
- the results that are obtained after data processing are a correct interpretation of the input data and the user can rely on them in decision making; result correctness is established by automatic testing and user feedback;

- reliability is important for this type of applications because it enables the application to have a high availability level for its users; availability is achieved using database clusters, backup power generators, a pre-established number of spare parts for replacing defective ones in computing machines and auxiliary hardware dependencies;
- interaction capability is important in such applications because it provides scenarios for improving user experience while interacting with the software product; this implies using client-side development more in order to ensure a high degree of fluidity to the application bringing it one step closer to the capabilities of desktop based applications in terms of interaction;
- operation procedure simplicity is related to how individuals make use of the software application and its functionalities; the simpler the operation procedure the easier it is for user to comprehend the logical structure and the operation diagram implemented by engineers.

Figure 1 presents a user oriented architecture which connects servers and databases with users. The architecture is composed of a set of server groups SG = {$Servers_1$, $Servers_2$, …, $Servers_{i-1}$, $Servers_i$, $Servers_{i+1}$, $Servers_{i+2}$, … $Servers_n$} a set of data stores DS = {Data A, Data B, Data C, Data D, …, Data Z} a set of work stations WSS = {$Workstation_1$, $Workstation_2$, …, $Workstation_{k-1}$, $Workstation_k$, $Workstation_{k+1}$, …, $Workstation_p$} a set of individual laptop computers LPS = {$Laptop_1$, $Laptop_2$, …, $Laptop_j$, $Laptop_{j+1}$, …, $Laptop_m$} a mainframe computer and the end users that have access to the application using a graphical interface.

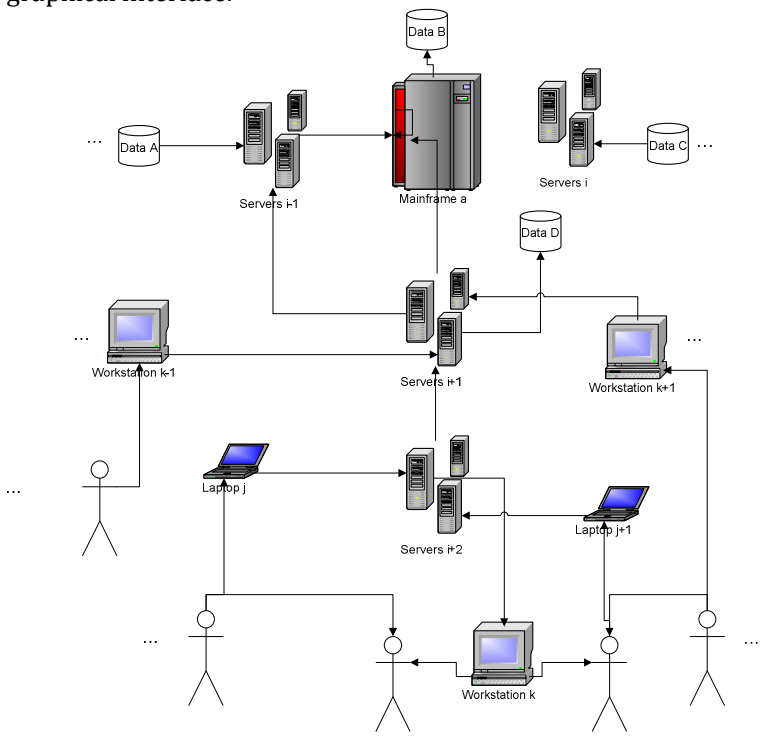

Figure 1. User oriented architecture

User-oriented online applications have the following advantages, according to [3] and [4]:

- give access to the desired resources through online databases that store information of interest to users;
- reduce waiting times for solving problems or for operations execution desired by the user;
- increase the efficiency of operations performed by rapid processing of the required operations and delivering results in a much shorter time;
- achieve the link between customers and suppliers by providing a collaborative environment for problems solving, services, procurement of services, provision of goods and their acquisition;

- improve companies efficiency by increasing sales and providing access to a much greater range of customers locally and internationally;
- give users access to a much greater range of products in an online space where the price quality ratio is high;
- make available to the public financial management systems to record individual income and expenses by eliminating the risk of mistakes and omissions made in calculations;
- users have access to online banking systems that allow checking account, online payments, management of bank deposits and transfers management.

Process oriented applications are online applications that focus on providing distributed functionality to users but without making use of a specific interface to allow graphical user interaction. These applications have the following characteristics:

- robustness by offering data services to developers and individual users according to their specific purpose; the service must use a robust architecture in order to comply with user requests efficiently;
- homogeneity in data transmissions in order to maintain a constant behavior in supplying information to users; homogeneity is achieved by using well formatted outputs;
- correctness is ensured by automatic testing on all methods that aim on obtaining data from the process oriented application; each data stream is analyzed and tested against the formatting pattern and information in the database;
- operation procedure standardization by employing the use of extensive documentation in order to clearly state methods and arguments, return values and format requirements; this is necessary in order not to induce confusion about offered services and communication protocols;
- reliability is important for guaranteeing a high level of non faulty operation time for the process oriented application.

Figure 2 presents a process oriented architecture which is focused on processing information from databases and establishing network connections between servers to distribute the results.

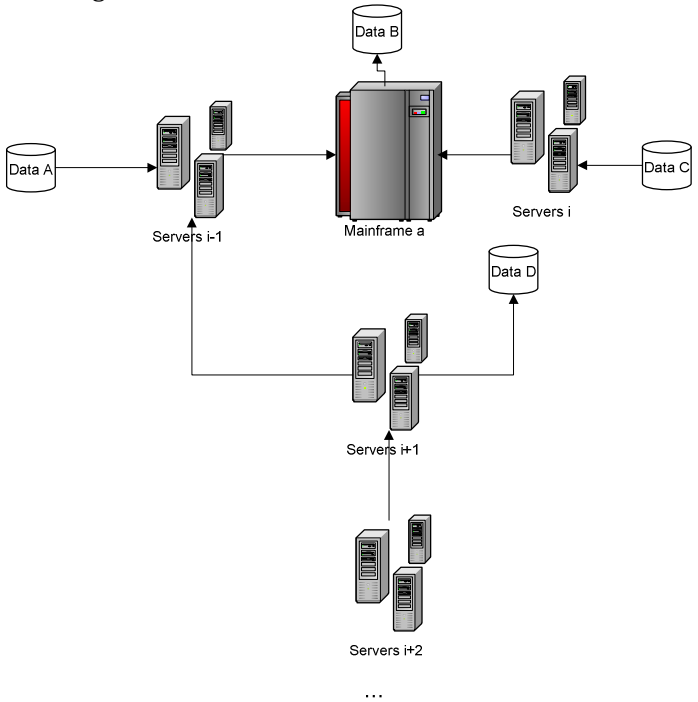

Figure 2. Process oriented architecture

Process oriented applications have the following advantages:

- fast data access by using efficient algorithms that process the requests and queue them up for response;
- improve user information flow by implementing standardized data formatters which have a high portability level;
- provide well structured information to the public through exposing methods that return the formatted information in a standardized form selected by a parameter in the method call;
- increase the efficiency in information gathering for third party applications that use its services;
- provide data services with no third party resource cost enabling increased data volume processing without affecting the user experience.

User oriented and process oriented applications serve the same purpose that of providing useful information in a correct way to different types of users located in a distributed geographical environment.

2. Informatics security

Information security is defined in [5] as the protection of information and informatics systems from unauthorized access, use, divulgation, interference, modification and destruction in order to ensure:

- integrity , defined in [5] as protection against incorrect information modification or destruction and includes ensuring non-repudiation and information authenticity;
- authenticity according to [5] is necessary to ensure that data, information or transactions are original; non-repudiation in [6] implies the fact that no one is able to deny sending or receiving a transaction; authenticity and non repudiation are applied in electronic commerce by using digital signatures [5];
- confidentiality is defined in [6] as keeping authorized restrictions regarding access and publication, including means to protect private life and personal information;
- availability is represented in [5] as insuring timely and trustworthy access to information.

In [7] and [8] informatics security addresses problems related to:

- informatics security risk by analyzing general concepts related to risk, vulnerabilities and threats;
- access control highlighting different authentication modalities and user required protocols; the vulnerabilities of control systems are analyzed considering the main types of attack they are subject to;
- cryptography aspects of security systems pursuing network key management, the most efficient encryption methods are identified and the encryption algorithms are described; advantages and disadvantages for each encryption method are presented;
- application security by detailing the newest security methods and presenting the role of application quality in this process;
- Internet security by describing vulnerabilities and threats in the online environment;
- network security by identifying vulnerabilities of transfer protocols and analyzing the threats of network communication;
- wireless network security, presenting aspects relating to vulnerabilities, threats and recommended security policies;
- cellular network security detailing security protocols for radio transmissions, attack modalities of radio networks and defense strategies applied to radio networks;
- ways of increasing the security level by improving code quality, by identifying and eliminating system physical vulnerabilities, user training, by increasing access control systems quality, by improving password management techniques, by increasing the

quality of security policies and clearly differentiating user roles, by developing a quality characteristic system associated to the security system and proceeding to improving the individual characteristic quality level in order to obtain a global quality increase of the informatics security system;
- management aspects of information security by clearly detailing the main components of a informatics security management system;
- intrusion detection systems and vulnerability evaluation techniques;
- the identification of legal aspects of informatics security by studying laws and regulations imposed globally for better practices in this domain.

Within informatics security systems quality plays an important role, being a major influence factor in the well being of an informatics application. The quality characteristics are placed in a hierarchy based on:

- source code:
  - homogeneity which is represented by the nature of source code to have the same characteristics and properties in all of the modules belonging to the security system; the use of operators and operands in a similar fashion is desired as using the same kind of formatting in each of the system modules;
  - intelligibility is defined as the characteristic of source code to be perceived easily by developers that did not have any prior encounter with it before; source code intelligibility in security systems is useful due the resource economy it produces by allowing easy understanding of the implementation logic thus rapidly making improvements or modifications to existing code;
  - testability is the capacity of the source code to undertake the testing process easily by covering all logical paths; a high testability level in security systems ensures a minimization of defect numbers and omissions of the system thus improving the global quality level; testability is ensured by the homogenous development of the security system and using logical internal reporting systems for all operations and events encountered in the security framework;
  - maintainability which if present in a high level minimizes defect fixing application improvement costs, this characteristic is in tightly related to homogeneity and intelligibility of source code;
  - data type veridicity thus pursuing the elimination of buffer overflow which leads to the security system corruption; lower and upper limiting of buffers is needed in order to ensure the system is protected from this type failure; enforcing buffer limitation as a standard for all input received by the application is a proven technique for avoiding memory corruption;
  - error perception is a characteristic which oversees the level on which the security system reports errors and interprets them correctly; a high level of perception in the security system reduces the costs relating to restoring the informatics application, paying compensations and the cost of reengineering the security system altogether;
  - using secret phrases in source code is a known security issue mostly when the phrases take the form of access tokens and are hardcoded for future use; this practice is not recommended as it raises several security hazards that are hard to control;

- interaction with the informatics application:
  - compatibility meaning using common communication protocols thus maintaining communication between the two entities in the best conditions possible;
  - coexistence defined as security system's ability to work at optimal parameters within the informatics application; a high level of coexistence increases the reliability degree of the informatics application and minimizes the maintenance costs associated to the security system;
  - accuracy is represented by the exact nature with which the signals emitted by the informatics application are perceived by the security system;
  - securing information is characterized by the security system's ability to protect and ensure the confidentiality of data used and processed in the informatics application;

- user interaction:
  - transfer security in the informatics application by implementing efficient authentication and authorization systems thus ensuring a correct user authentication effort, minimizing identity theft cases and costs with:
    - restarting the security system of the informatics application after a breach;
    - damage evaluation provoked by the unauthorized access in the informatics application;
    - paying compensations due to compromising protected assets;
  - the validity of data inputted by the user; by ensuring a high level of which the attack opportunities are minimized and human-machine interaction is improved; data validity is ensured by implementing validation controls and procedures to prevent the most common and dangerous informatics attacks to which the application is exposed.

Software vulnerabilities are discovered in all stages of interaction. In order to determine vulnerability density in a module the following formula is used:

$$DENV_j - \frac{NRV}{NRLM_j}$$

where:
NRV – number of vulnerabilities;
$NRLM_i$ –number of source code lines of the analyzed module.

Table 1 presents the DENV indicator values corresponding to vulnerabilities found in the AVIO software.

Table 1. Vulnerability distribution in the AVIO software

AVIO module	Vulnerability type	No. Vulnerabilities	NRLM	DENV
TextOrthogonality	Input Validation	1	186	0,00537634
Images	Input Validation	1	9	0,11111111
Validator	Input Validation	1	248	0,00403226
ImageValidator	Input Validation	1	21	0,04761905
OrtoImage	Authorization	4	61	0,06557377
NameOrthogonality	Type equality	1	26	0,03846154
Log	Viewstate management	1	10	0,1
LogoInput	Upload	4	113	0,03539823
ImageLoad	Upload	2	309	0,00647249
NewUser	Authorization	1	10	0,1
Login	Extend authentication protection	1	10	0,1
Handler1	Method access privilege validation	1	20	0,05
UserAdmin	Stack exposure	1	86	0,01162791

A low value, closer to 0, of the DENV indicator shows that there is a low degree of vulnerabilities per line of code. A high value closer or greater than 1 presents a high concentration of vulnerabilities per line of code which shows that the specific module has a poor quality level.

Informatics security is a necessary requirement for large distributed systems according to [9]. It is imperative to develop secure systems in the conditions of an increasing number of threats and threat agents against known vulnerabilities in the system. The vulnerability density indicator is important for determining vulnerability prone modules in the source code and determining a fixing priority based upon it.

3. Informatics security influence factors

Informatics applications are complex constructions used in defined social and economical contexts. The influence factors are numerous and have diverse effects.

According to [10] 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 [11]

  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.

According to [10] the indirect factors that determine security are:

- complexity which has an important effect over informatics security, according to [12] as the software product's complexity grows so does the number of defects thus decreasing the level of informatics security. Complexity in the AVIO software is defined using the cyclomatic metric which 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;

Table 2 presents the cyclomatic complexity measures recorded in the AVIO application.

Table 2. Cyclomatic complexity of AVIO classes

Class	Cyclomatic Complexity
TextOrthogonality	72
Image	4
Organization	9
NameArray<T>	9
ComplexArray	1
Complex	46
ImageMetrics	42
BitmapAlredyLoaded	4
Validator	97
ImageValidator	17
UnmanagedImage	43
RGBL	3
RGB	1
Histogram	12
ColorSetLocations	17
ColorLocationList	12
ColorLocation	8
BmpStatisticsHelper	23
BmpHelper	60

StatisticsHelper	17
LogHelper	21
ColorPair	1
OrtoImage	60
NameOrthogonality	25
Handler1	30
LogoInput	10
Login	10
NewUser	15
ImageLoad	20

In Figure 3 the graphical representation of the influence of direct and indirect factors over informatics security is presented.

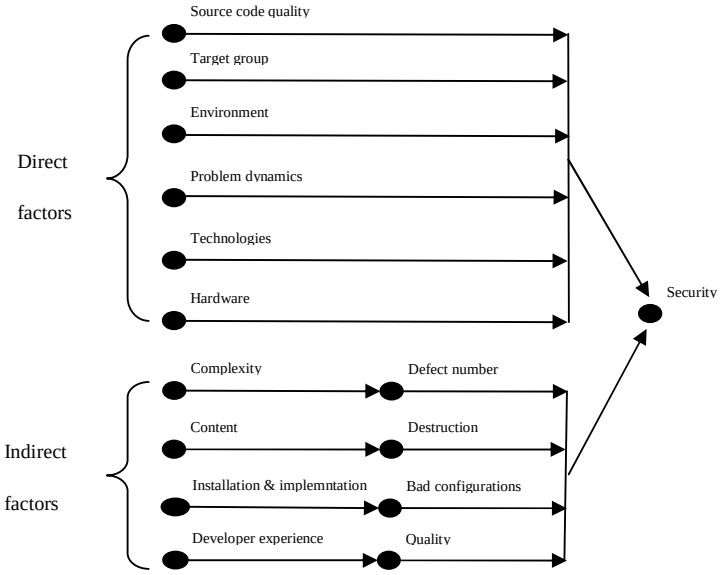

Figure 3. Graphical representation of the security influence factors [1]

The influence factors are an important element in informatics security analysis and in establishing cost model coefficients.

4. Security costs factors in the development of the AVIO application

Informatics security cost analysis is done in [14] by analyzing the security costs involved in the development in distributed applications. Other publications like [15], [16] and [17] analyze security costs and effects in distributed computing. In order to determine the security cost generating entities a development security costs analysis is done on the AVIO product to reveal the elements included in this part of the project. Figure 4 presents the operation environment of the AVIO application [18].

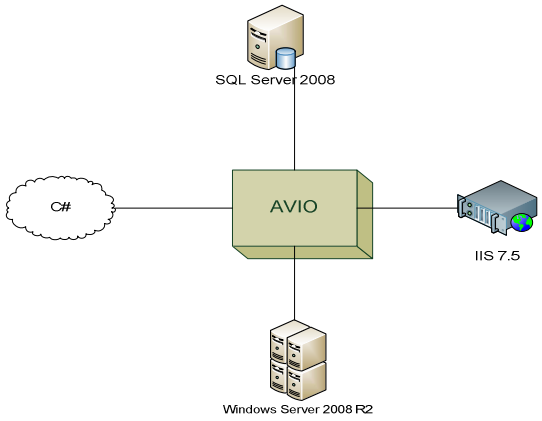

Figure 4. the AVIO operation environment [18]

Development costs are influenced by:

- system analysis which is represented by the analysis of operating environment as the work undertaken to determine the factors influencing operating in the present application environment for AVIO; The operating environment for AVIO is influenced by the following categories of factors:
    - factors involved in the development strategy of AVIO are:
        - the programming language C# used both for developing functional libraries within AVIO and developing the security system;
        - the database management system in SQL Server 2008 which is used to store the organizational identifiers and monitoring application behavior;
    - factors involved in the operation of AVIO:
        - access modality is represented by the online environment the informatics application is available at https://www.dragospalaghita.ro;
        - the characteristics of the server the application is hosted on:
            - type: Microsoft – IIS/7.5;
            - OS: Windows Server 2008 R2 x64;
            - technology: ASP .NET 3.5;
            - data base server: SQL Server 2008 Web Edition x64;
    - factors involved in the diversity of user groups of the application as:
        - non-homogeneity which affects the operation of the application because it makes user actions unpredictable;
        - user motivation which is represented by the intentions of users when using the application; this factor leads in two directions one which is using the application functionality and the other being the will to inflict damage upon software components which leads to loss of data integrity;
- system planning and design which is done according to use case and misuse case analysis; use case analysis represents the activities undertaken to determine according to regular use cases attack possibilities and representing them as misuse cases for the AVIO product; the developed analysis identified paths that are open to cybernetic attacks initiated by inside or outside entities; the result of use case analysis activities is the identification of possible attacks on operations allowed within AVIO by determining the situations that are favorable to an attacker and through which gains access to confidential information or provokes damage to the software system; in the use case analysis protection method identification is done through finding means and measures that handle the unwanted effects of a cybernetic attack or directly prevent it from happening; to this extent the use case diagram is altered by adding methods that aim to improve the security level of AVIO; by determining the countermeasures aimed and stopping or minimizing the effects of informatics attack necessary information is obtained for risk analysis and management; the structural analysis of AVIO is represented by planned

activities in order to determine the methods that produce attacks by taking advantage of structural deficiencies in AVIO;

- vulnerability analysis Vulnerability identification represents the set of activities undertaken to determine the exiting vulnerabilities in the AVIO software. Vulnerability analysis is elaborated by studying:
    - use cases and determining vulnerable functional areas in the AVIO software;
    - the operating environment of the AVIO software identifying vulnerabilities in operating systems, data base management systems and web servers used to ensure the operability of the software;
    - the functioning mechanisms involved in the operation of the AVIO software by testing existing functionalities in order to determine which modules of the software product present functioning problems considering informatics security;
    - the existing source code to determine the code sequences which due to implementation logic pose security threats through the opened vulnerabilities.
- threat identification represents the group of activities that are undertaken to determine threats for the AVIO software; a software attack is the action undertaken by an individual towards a software product targeting a weak entry point, vulnerability, a defect or common software functionality to get unauthorized software access, to cause destructions or alter the behavior of the application;
- asset identification is represented by the process of identifying entities in the application they must be protected; the identified assets in the AVIO software are:
    - authentication details of registered user accounts;
    - the integrity of the main database that stores user accounts, organizational identifiers and web monitoring records;
    - the integrity of organizational identifiers processing algorithms;
    - the operability of the AVIO software;
- risk assessment is represented by the set of activities that result in the development of risk quantification measures that are used in risk management for the distributed application AVIO;
- effective development costs that are based on information from previous activities and are represented by the effective costs of source code development.

Security system maintenance costs are related to operations made in the post-release lifecycle of the application and directly focus on defect fixing. An important factor in the immediate effort required by resolving a defect is severity. Severity will differ according to different factors like:

- affected functionality which is of high importance for severity as it marks the software functionalities made unusable, faulty or incompatible with established quality standards;
- number of replications refers to the number of defect or problem reports listed for this issue, these defects are all marked as duplicates except one which is to be solved;
- workarounds means methods or ways to get around the specified defect; these include avoiding the incorrect functionality, using a different input such that the defect won't replicate or any other way that does not trigger the faulty behavior; if indeed a workaround is found then the defect will be marked as lower severity due to this, a defect with no available workarounds is suitable for a higher severity;
- impact is related to the effect of the defect on the software product as some defects are specific only to a restricted functionality or code are where as some are have an effect on more modules thus affecting more software functions; the impact of the defect is higher as the source code error that introduced the defect is in a more critical and used area of the source code;
- severity lowering cost means what will the organization loose in credibility, compensations and effort if the defect is given a lover severity by the QA manager; reasons for doing this include a high number of more urgent defects, no developers available to solve the issue in question or no interest for the organization to resolve the issue due to lack of usage of the affected functionality in the upcoming version and assume the risk of leaving it open.

Security system operation has costs regarding the time required by security procedures when analyzing and validating user input. After simulating 4096 user access in the AVIO system an average cost for security procedures was established for structured entities validation equal to 3 milliseconds.

5. Conclusions

Security cost analysis is based on determining the main cost generating factors in the software product. These factors are related to the development operations that are undertaken to complete the security system for the informatics product. The nature of the factors is variable according to the developed application and it is dependent on how the software product operates. Also the study showed that there are a multitude of factors in the operation environment that each has an influence on how the security system operates.
Security costs are determined at source code development level based on the influence factors that were highlighted in section 3. The customization of those factors to the AVIO application shows that the impact is different for each of them depending on the characteristics of the AVIO software product.
Insecurity is an important issue for costs and it generates reputation losses and compensations to affected parties. The cost of insecurity is usually much higher than the development of an average security system that can handle most of the security threats present today. Insecurity is generated due to the lack of a security system or the existence of a deprecated one that doesn't handle all the present security threats and is easily bypassed by malicious users.

Acknowledgements
The article *Informatics security cost factors in web applications* 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.

Bibliography

[1] Ion Ivan, Bogdan Vintila, Dragos Palaghita - *Types Of Citizen Oriented Informatics Applications* - Open Education Journal, Russia, ISSN 1818-4243, No.6, 2009
[2] Bogdan Vintila, Dragos Palaghita, Sorin Pavel - A Qualitative Model For Managing The Development Cycle Of Citizen Oriented Applications, Al 34-lea Congres ARA: Cercetare Stiintifica – Securitate – Dezvoltare Durabila Conexiuni, 18–23 mai, 2010 Bucuresti, Romania
[3] Ion IVAN, Bogdan VINTILA, Cristian CIUREA, Mihai DOINEA - *The Modern Development Cycle of Citizen Oriented Applications* , Studies in Informatics and Control, Vol. 18, No. 3, 2009, ISSN 1220-1766
[4] Ion IVAN, Bogdan VINTILA, Cristian CIUREA, Mihai DOINEA - *Citizen Oriented Informatics Applications development Cycle* , Ekonomika, Statistika, Informatica, MESI, no.4, 2009, pg.139 - 145, ISSN 1994-7844
[5] http://csrc.nist.gov/publications/nistir/NISTIR-7298_Glossary_Key _Infor_Security_Terms.pdf
[6] http://en.wikipedia.org/wiki/Information_security
[7] J. Vacca - *Computer and Information Security Handbook* , 928 pages, ISBN: 978-0123743541, Morgan Kaufmann, 2009
[8] H. Tipton, M. Krause - *Information Security Management Handbook, Sixth Edition*, 456 pages, ISBN: 978-1420067088, Auerbach Publications, 2008
[9] E. B. Dudin, I. A. Zhlyabinkova, E. G. Zakharova And Yu. G. Smetanin "Information security in distributed computing systems. A review", Automatic Documentation and Mathematical Linguistics, vol. 43, no. 4, pp 234-240, ISSN: 1934-8371, Springer New York, NY, USA, 2009
[10] I. Ivan, D. Palaghita, "The Informatics Security Cost of Distributed Applications", *Theoretical and Applied Economics*, vol. 17, no. 1, pp 49-68, Jan. 2010.
[11] Omar H. ALHAZMI, Yashwant K. MALAIYA - *Application of vulnerability discovery models to major operating systems*, IEEE Transactions on Reliability, vol. 57, issue 1, pp 14-22, ISSN: 0018-9529, IEEE-Inst Electrical Electronics Engineers Inc, 445 Hoes Lane, Piscataway, NJ 08855 USA, 2008
[12] Paul POCATILU – Costurile testarii software, ASE Publishing House,252 pages, ISBN: 973-

594-549-5, Bucharest, Romania, 2004
[13] Maurice HALSTEAD - *Elements of Software Science, Operating, and Programming Systems Series*, Volume 7, New York, NY: Elsevier, 1977.
[14] Ion Ivan, Dragos Palaghita, Bogdan Vintila – 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, ISBN 978-973-1747-32-1
[15] Dragos Palaghita, Bogdan Vintila - The Analysis Of Informatics Security Costs In Citizen Oriented Applications, Journal of Information Systems & Operations Management, ISSN: 1843-4711, pp. 54-66, 2010
[16] Dragos Palaghita - Informatics Security System Planning and Development Using Open Source Components, Open Source Journal, vol. 2, no. 3, 2010, pg. 65-84, ISSN 2066-740X
[17] Ion Ivan, Adrian Visoiu, Silvia Trif, Bogdan Vintila**,** Dragos Palaghita – The security of the Mobile Citizen Oriented Applications, Economy Informatics, vol. 10, no. 1, pp.22-33, 2010
[18] Ion IVAN, Dragos PALAGHITA, Sorin VINTURIS, Mihai DOINEA -*Vulnerability Minimization Model in Web Distributed Applications*, Proceedings of the 3rd International Conference Security for Information Technology and Communications, 18 - 19 November, 2010, Bucharest, Romania, ASE Publishing House, pp.39 - 48, ISBN 978-606-505-385-4