Security in IT Application in the Store

Liviu Adrian STOICA
Bucharest Academy of Economic Studies
stoica.liviu@csie.ase.ro

The objective of this paper work is to secure IT Application in the store. It applies Informatics Security Master knowledge to make the software better and more stable. The major improvement is the Licensing module which limits the software use to the persons with rights and doesn't let it to be used by persons without rights. The need for this approach is given by the supply of correctness in the long term transaction in correlation of its operators. It describes the architectural model of a store application, the implementation of the security modules and the licensing system.
***Keywords**: Licensing, Copyright, Security, Costs of Security, Store*

1 Introduction

The main benefit is to encourage creation by making it financially viable in order to make a living, and its main purpose is to increase the profit for the author of the IT Application in the store. This is achieved by selling a lot of product copies for money. Without a proper licensing system the software would be used for free by everyone, so the time spent and the money invested by the author would be for nothing or just a bad investment. To survive in this world and in the business you need to carefully protect everything you make against exploits.
Copyright is the legal concept that gives the exclusive rights of a product to its creator. It gives the creator the right to be credited for his work which he will financially benefit from.
The means used are:

- analyzing business economy which means : collecting data about the companies based on product selling and the papers needed to develop a fast and reliable business system that is easy to use, protecting confidential data, limiting persons the access to the company's secrets, helping managers to reduce costs and invest in future and profitable products, taking risk decisions easier and better, reducing the human error risks;
- Visual Basic .NET - VB.NET which is an object-oriented computer programming language that is viewed as an evolution of the classic Visual Basic - VB, which is implemented on the .NET Framework by Microsoft, by two main editions of IDEs for developing : Microsoft Visual Studio 2010 – currently used to develop the application – and Visual Basic Express Edition [14]; I used this because it provides a quick functionality with auto-complete syntax and has the most functions that are already integrated;
- cryptographic algorithms such as Triple DES which is the common name for the Triple Data Encryption Algorithm - TDEA or Triple DEA - block cipher, which applies the Data Encryption Standard - DES - cipher algorithm three times to each data block. The original DES cipher's key size of 56 bits was generally sufficient when that algorithm was designed, but the availability of increasing computational power made brute-force attacks stronger. Triple DES provides a relatively simple method of increasing the key size of DES to protect against suck attacks, without the need to design a completely new block cipher algorithm [15]; this algorithm is used because it is more stable than DES and easier to integrate, more reliable and secure;
- MySQL database is the world's most used open source relational database management system that runs as a server providing multi-user access to a number of databases; it is written in C and C++ and it works on many different system

platforms [16]; it is used because it is a free platform, with a huge tutorial, very friendly forum and rapid support; indeed it is not as stable and performing as the SQL database or Oracle but considering that it is used on a transactional software for a market with under one million transactions it is more than enough and it has the best performance versus costs.

2 IT Application in the Store

Selecting a target group is the basis of any business and the only way that effectively increases sales and decreases marketing efforts. It is clear that income and success depend largely on the attitude of the target group and how you respond to it. If the wrong target is chosen in the beginning, it becomes a loss of time and money and a failure to reach any results.

For selecting optimal segment the following criteria has been established:

- customers are facing a problem; optimum management of a business is solved by creating something that manages all the documents of the business, input rights to employees, creating detailed reports, keeping logs and tracks of everything that happened in the business, lowering the risk of human errors occurring, fraud, access to classified documents;
- customers need a solution to their problem; lack of software that controls input-output relationships in a company generate high maintenance of the business and allow lots of human errors that are destroying the company; all these make the need of a software to be a must for a successful company;
- how the clients are found; they are part of the same market and economical segment, distributed uniform along the country in every city;
- customers paid in the past to solve the current problem; most of the potential customers already own a software but are not pleased about it; it is not sufficiently powerful and easy to use -which slows the business management instead of making it faster and reliable;
- customers don't want to pay again; it matters less if you are providing a great product if those who are in need of it cannot afford to buy it; the sale price of the product is a low one comparative with the software capabilities; the profit is made by the rule 'cheap and more' rather than 'expensive and few';
- how many potential customers are there and if they are enough to develop the business; there are over 200.000 companies with economic profile in Romania, and more than 75.000 are based on sales;
- how is the collaboration with the potential clients; given that the target group is exclusively for legal persons it is assumed that they are good prepared and well trained which means they understand the need of a software, its functionality, implementation and all the benefits that it offers; this means an easy-to-maintain collaboration with the customers;
- customers schedule; most companies have normal working hours between 8 am and 8 pm, that leads to providing a rapid support for them.

Table 1. Customers repartition [20]

C.N.	Business	2008	2009
1	Share of enterprises that have used PC in total enterprise assets (%)	80.4	82.2
2	Share of enterprises with Internet connection in all active enterprises (%)	72.2	78.6
3	Persons that use a PC out of the general population (%)	26.9	29.3
4	Persons that use an Internet-connected PC out of the general population (%)	22.1	24.8
5	Investment in hardware (Millions RON)	799	702
6	Investment spent on information technology products and services (Millions RON)	2969.9	3713.6
7	Percentage of enterprises that use broadband connection (%)	40.7	49.1
8	Percentage of enterprises that have their own website (%)	28.0	34.6
9	Percentage of the turnover achieved via the Internet out of all the enterprises with economic activity (%)	2.3	3.6
10	Percentage of the turnover achieved via the Internet out of all businesses that sell online (%)	28.0	28.3

As observed in Table 1, the potential customers are increasing from year to year. In this table is described the evolution of the investment in the IT Technology for every business in Romania. As a prevision, it is clear that all the business will need computers; internet and the employees will be trained well enough in order to use a system that will change the manual work with computer work.

Based on these criteria, the target group is to be given to companies with objects of economic activity involving the sale of products. Examples of this kind of companies are as follows: supermarkets, mini markets, fast foods, bakeries, clothing, footwear, auto parts, drugstores, pet shops and almost any business from the smallest ones to the big shops with hundreds of thousands of products or chain shops.

The receipt cost RT is calculated using the formula:

$$RT = \sum_{i=1}^{N} Q_i * P_i$$

where:

Q_i - Quantity of product i;

P_i - Price per 1 piece of product i;

N - number of products;

i - product;

Price can be represented as 0.00 or 0.0000 or 0.000000 and kept as reference.

In the case where:

P_1=120.7357 referred on 0.0000 format (suppliers);

P_2=120.74 referred on 0.00 format (fiscal printer);

P_3= 120.735714 referred on 0.000000 format (CNAS);

where i=1,2,3 = price of 1 unit of Cetrotide 0,25 Mg 0,25mg

Q_1=Q_2= Q_3=30;

N=1;

We have:

T_1 = 3622.071

T_2 = 3622.2

T_3 = 3622.07142

Due to some software products that keep control of the stock on 4 or 6 decimal places and the Romanian Economical Law that says you must submit invoices on 2 decimals, the receipt and invoice totals are different. That is happening because of software errors like this one: the stock is on 4 decimals, means the sales prices is something like 3.2354 ($P_{example}$=3.2354), now when they make the

invoice of a quantity, let's presume is 30,($Q_{1,2}$=30) they do T_1=30 * 3.2354 and round it to 2 decimals, means T_1=97.06 which is a big mistake because T_2=30 * 3.23=96.90, which means a difference of 0.16 . Due to these things the pay system is irregular with errors, the basic accounting cannot be held well, so our solution is to make a legal note of this difference and keep a record for each supplier. This maintains an accurate accounting system. It is considered a plus of the software that attracts new customers as a magnet and increases the sales.

- introduction of utility bills;
- evidence for invoices, utility bills which allows further editing (adding or deleting items on the bill already introduced), delete the invoice, payment record to the provider : introduction or deleting payment (with its automatic introduction on the cash book or bank register according to its type), header edit of the invoice, invoice and payment centralization;
- evidence of input and outputs with filters for categories of product description, VAT to buy or sell type of the reception, validity, period of reception and report for the filter;
- current stock with detailed product sheet which allows adding or changing products directly in the existing stock, automatically creating the near protocols;
- nomenclatures for: VAT, producer, shape, categories, subcategories, partners;
- primary ledger containing cash books, bank register, purchases journal, sales journal, reports, inventory balance, credit balance and the afferent reports;
- interface that allows issuing sales, receipts, customers promotions;
- record of input-outputs;
- notes for stock correction, expired products, consumed products;
- appointments;
- export to XLS, PDF, DOC;
- logs and records for every operation that was done by a certain user.

To better describe the relation between the clients and the products used in the business, I made a table with the relation between customers and products.

Table 2. Client-Products relation

clients\products	Product 1	...	Product j	…	Product M
Client 1	x	x			x
...		x	x		
Client i		x	CP_{ij}		
...				x	
Client N		x			

Where CP_{ij} represents the relation between Client i and Product j. As an example we have Client = Liviu Stoica which in clients' list is number 12 and an acquired product that is Laptop Fujitsu, which in product list is number 4 and the number of product the client bought is 3, so $CP_{12\,4}$=3.

In Table 2 we observe the real correlation between the clients and the products used in the software, which is: a client buys a product or more products, a product is sold to a client or more. From this we make reports on clients and products.

Based on the same table schema we make the reports based on how much a client spent in the store and the discount marketing schema. All we need to do is to replace the product with the total amount spent which makes the reference like this: Client = Stoica Liviu, Spent = 3400 (3 laptops (1100/unit) + 1 Mouse (100/unit)) which makes $CP_{12\,4}$=3400. To make it more detailed we have a table with the following header : Nr. Crt., Total Spent, Total products bought, Total receipts,

Discount accorded, Classification which give us the report of the best customers ; the same we do for suppliers : Nr.crt, supplier , total value (we buy), total discount (we get).

The circuit of using the software is determined by the commercial order.

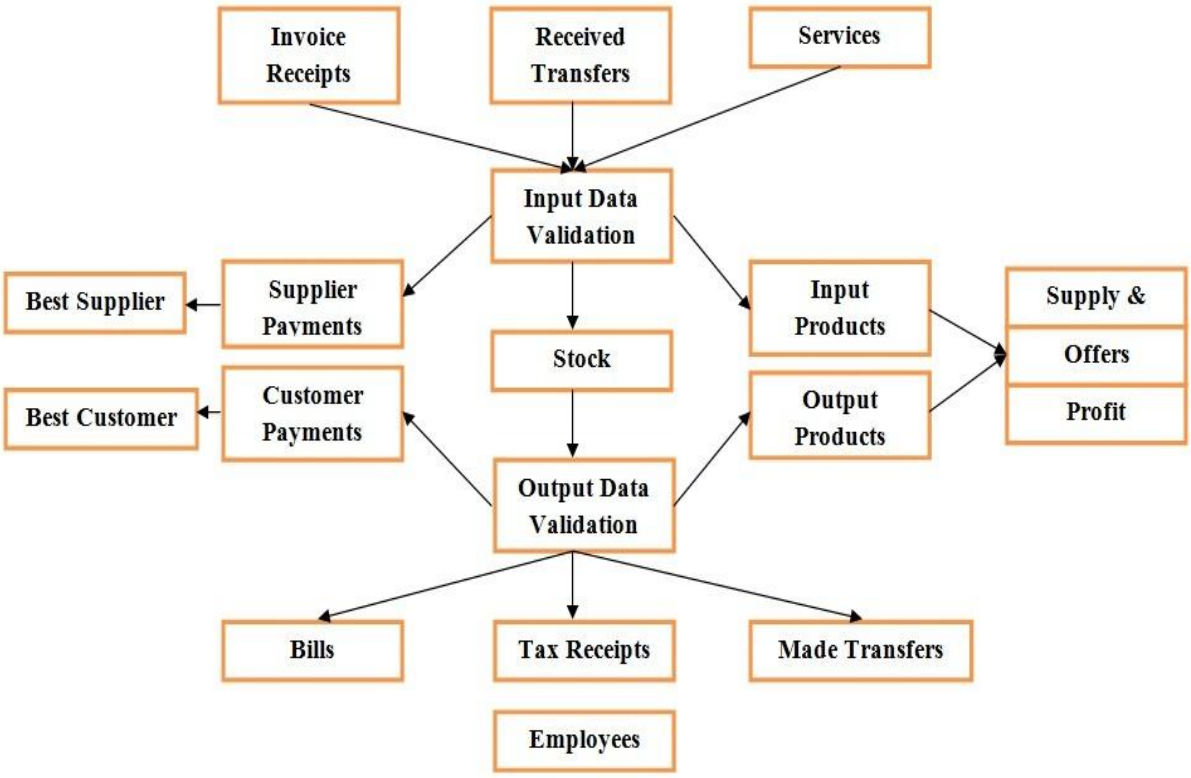

Fig. 1. IT Application Software architecture

In the Figure 1 we see the software diagram that represents the economic model of the economy in the current business. We observe the transaction course in the software, the correlation between the input-output relations. All the software is based on the stock which is the core of the transactions. Almost all are related to the stock flow, need of supply and satisfy the demand. The profit is calculated by the products' flow.

Software Complexity SC_i of a module i is calculated using the formula:

$$SC_i = M_i * \log_2(M_i) + L_i * \log_2 L_i$$

where:

M_i - number of submodules of module i;

L_i - number of links between submodules of module i;

i - module name(software, inputs, outputs, reports, payments, validation, products, etc.);

$SC_{software}$= 139.06

SC_{inputs} = 23.48

$SC_{outputs}$= 23.48

$SC_{reports}$ = 19.6

$SC_{payments}$ = 10

$SC_{validation}$ = 44.96

$SC_{products}$ = 10

As a product, it has several steps in order to make the correlation work between input and output entries which are explained in a few important steps:

- authentication (each employee logs in with his user and password and accesses the functions that he is certified and trained for);
- invoice receipt (the receiver introduces the goods in the system and decides the sale prices and offers; he uses the pay system, too, for keeping record of the money invested in the business and all the costs involved which means the stock and the relation with the suppliers is built) ;

- selling the products (the cashier serves the clients of that business, which means the relation with the customers is starting to build and the generating of profit and the business itself has started);
- analyzing reports which determine the productivity of employee, the total sales, profit, future prices and offers, what client should benefit from promotions, which is the best supplier for each product; this improves the business and the income; it also shows if the current chosen way is a good or a destructive way, make forecasts of the future of the business and its future income; at this step you determine where to increase funds in developing or where to cut off from, because they produce a constant lost.

The IT Application in the store was created to ensure an efficient flow for the business and an accurate register and relations between business, suppliers and clients.

3 Conditions of Security in IT Application in the Store

A good software is the one that is dependable, executes predictably and operates correctly under all conditions (hostile conditions, software comes under attack, runs on a malicious host), trustworthy, contains few if any vulnerabilities or weaknesses that are intentionally exploited, the software dependability (the software does not contain malicious logic that makes it behave in a malicious manner) and survivable, which is strong enough for the most known attacks and it recovers fast after a new attack.

Software is more secure when we take care of the following aspects [11]:

- development principles and practices which means that during the development it is considered and evaluated the security in each phase of the software life cycle;
- development tools used, like the programming language used for development, which if it is well chosen it avoids the security vulnerabilities and supports secure development practices and principles
- testing practices and tools used; software is tested using special tools against security issues or malicious attacks;
- deployment configuration; the installation of the software is minimized to the exposure of any residual vulnerabilities that it may contain;
- execution environment; the third party tools that assist to ensure the software security on the running operating system, like anti-viruses or malicious tool removal;
- programmer knowledge; the analyst, designer, developer, tester and maintenance crew are provided with the necessary information about security faults and exploits, so they manage those incidents and fix them.

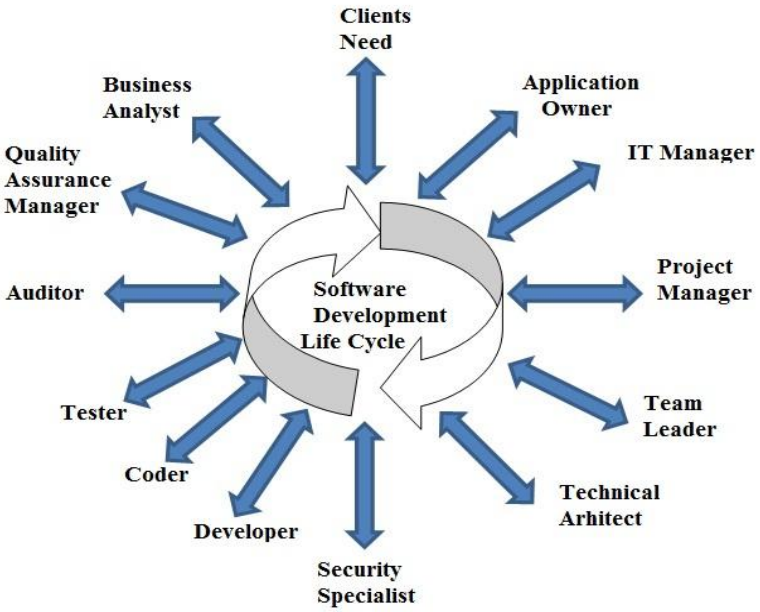

Fig. 2. Software Development Life Cycle

Based on the functionality that is required by the business and the technology requirements that are on a constant development and improvement, it is important to take in consideration the answers for the following questions, which improve the security during development phase [17]:

- the type of users that are using the software that are only internal, only external or both internal and external; in IT Application the software uses only internal users;
- what roles are needed for each users and what system is used to manage the user authentication and role management, which the Rights module is implemented for in order to manage the roles;
- what database is used and what we know about it, like the exploits already known, the protection that it offers, types of authentication and encrypt algorithms that are used, and the reason that MySQL was chosen for, a fast, reliable and secure database;
- data access, which data users have access to, if there is any special protection mechanism that is required by the software where the Rights module was needed and adjusted to maintain both the users and data protection;
- technology and what technological solutions are available, if a web service or a client/server architecture are included; in this case that is only a desktop application, the client/server is maintained by the MySQL server;
- existing threats, what threats already exist on the company that is using the software product that attacks the software itself, the confidentiality about the suppliers and the prices of the products, the customers database which are protected by the cryptography algorithms being used;
- if the design is too permissible and if all the data fields are tested during tests, the fields were adjusted according to the database table management;
- assuring if every function was tested according to its capabilities and exposures and if it was tested against malicious things to ensure that the software won't break and maintain a high defense against SQL Injection procedures;
- implementation, that referes on what system platform is the software implemented on and the other programs that assist for the benefit or for the attack of the product, as well as how safe is the environment and how the software is running on a malicious environment.

Some main methods to ensure the security as spotted in [11] are:

- minimize unsafe function use that refers to buffer overrun vulnerabilities and being aware of the library version and operating system functions used;
- use the latest compiler toolset because it is important that any error generated by the compiler is analyzed and solved;
- use static and dynamic analysis tools and source code and binary analyzing tools;
- manual code review is needed especially for reviewing the high-risk code like the one that interacts with the Internet or parses data from the Internet, web specific vulnerabilities such as cross-site scripting, database specific like SQL injection, common cryptographic errors like poor random generation or weak secret, mathematical errors;
- validate input and output is assured by simply validating the input data which remedies most of the vulnerabilities;
- use anti-cross site cryptographic library, that is a minimal defense, by HTML encoding to all the web-based output that may include untrusted input;
- use canonical data formats means that the canonical representation ensures the forms of an expression which does not bypass any security or filter mechanism;
- avoid string concatenation for database query is done by building statements query with input strings that protect against exposure for malicious injection;
- eliminate weak cryptography by using proven algorithms only;
- use logging and tracing that is very important for security, monitoring and debugging of an application.

Table 3. Adapted from the Saltzer & Scroeder Protection of Information in Computer System [10]

Design Principle	Explanations	Solution
Economy of mechanism	Keeping the design simple and less complex	Modular Code, Shared objects and Centralized services
Fail-Safe defaults	Access denied by default and granted explicitly	User management module
Complete mediation	Checking permission each time a request access to a form or a function is needed	User rights
Open design	Design is not a secret, implementation of safeguard is	Cryptographic and hashing algorithms
Separation of privilege	More than one condition is required to complete a task	Split keys, Compartmentalized functions
Least privilege	Rights are minimum and users granted access explicitly	Non-administrative accounts, Need to know
Least common mechanisms	Common mechanisms to more than one user/ process/role is not shared	Role based dynamic libraries and functions
Psychological acceptability	Security protection mechanism unbeknownst to the end user for ease of use and acceptance	Help dialogs, Visually appealing icons.

For the best security improvements, it is required the use of automated testing tools such as: fuzzing tools, network vulnerability scanners, web application vulnerability tools, packet analyzers, automated penetration testing tools, network/web proxies that

manipulate network data, protocol analysis and anti-malware detection.

4 Security Costs

Adding security to software means implementing new features in the source code. That is considered a cost because implementing security increases the effort required for developing the program. The security in a software is a two-way need because if the product is poorly designed it causes losses for the clients that are using it, which means that they can sue the producer and if the product is not properly secured, the customers can use it for free, which means no profit for the producer.

Major cost items are: use of new tools (hardware or software) that are required for developing secure software, more developing training which is composed of two parts: the time used during training and the cost itself, increase the effort level by hiring new persons, impact on delay of the final product.

To elaborate a cost formula we take in consideration the following elements [12]:

- percentage of source code change;
- complexity of software after the change compared with the base;
- program documentation before and after;
- programming team capability (before and after training);
- tools used to add security as cost of new hardware and time to learn them;
- change in development time;
- overall employees per month and months;
- the average cost per employee per month;
- estimate of reliability requirements;
- losing by delayed software finishing date.

To calculate the total implementation cost TIC is used the formula:

TIC = EC + TRC + CNE + CD

where:

EC - Effort Cost;

TRC - Training Cost;

CNE - Cost of New Equipment;

CD - Cost of Delay;

The formula for effort cost EC is:

EC = ECH / 100 * CE

where:

ECH - Effort Change;

CE - Current Effort;

To calculate effort change is used:

$$ECH = \prod_{i=1}^{n} W_i \frac{A_i}{B_i}$$

where:

$W_i \frac{A_i}{B_i}$ is the indicator of the changes between software (B-Before and A-After) of variable i and n = number of variables

For calculating training cost TRC is used the model:

TRC = ETX * CTE + NEX * CNEH

where:

ETX - number of Employees in Training;

CTE - cost of Training per Employee;

NEX - number of New Employees;

CNEH - cost for New Employees Hired;

Table 4. Costs

Variable	Before (B)	After (A)	$W_i = \frac{A_i}{B_i}$ Indicator
Code lines(i=1)	35 (thousands)	43 (thousands)	1.23
Complexity(i=2)	4 (vl,l,m,h,vh)	6 (vl,l,m,h,vh)	1.5
Documentation(i=3)	4 (vl,l,m,h,vh)	5 (vl,l,m,h,vh)	1.25
Analyst Capability(i=4)	6 (vl,l,m,h,vh)	7 (vl,l,m,h,vh)	1.17
Programmer Capability(i=5)	6 (vl,l,m,h,vh)	7 (vl,l,m,h,vh)	1.17
Time Tools(i=6)	0 (days)	0 (days)	0
Time Before(i=7)	6 (months)	7 (months)	1.17

Reliability Before(i=8)	6 (vl,l,m,h,vh)	6 (vl,l,m,h,vh)	1
		$\prod_{i=1}^{8} W_i \frac{A_i}{B_i}$	3.69
Current Effort	3000 €		
EC			110.7 €
Employees in Training	0 (persons)	1 (persons)	
New Employees	0 (persons)	0 (persons)	
Cost of Training	0 €	2000 €	
TRC			2000 €
Cost of equipment	0 €	300 €	
CNE			300 €
Delay Cost	450 €	450 €	
CD			450 €
TIC = 2860.7 €			

(vl, l, m, h, vh) = very low, low, medium, high, very high, noted from 1 to 10.

It means that to implement security on the current software costs 2860.7 euro, but considering that a license costs 200 euro means 15 new customers are enough to compensate the security costs. Thinking that without the implementation of the security, the potential customers could use the program for free; means the cost to implement is insignificant compared with the benefits obtained.

5 Risks in Utilization of IT Application in the Store

Risk is an event that is waiting to happen. It is described as: [9]:

- threats that exploit eventual system weaknesses.
- combination between the probability of an event and its consequences (ISO Guide 73).
- a vulnerability triggered or exploited by a threat (NIST SP 800-3).

Risk analysis assumes a security risk identification process, determining the amplitude and also identifying the areas with a high degree of risk that need to be secured. Risk analysis is a part of the assembly of measures that are called Risk Management. Risk evaluation is a result of a risk analysis process.

Risk management is defined as the total system of identification, control, and elimination or minimization method of the events that affect the system's resources. This includes [9]:

- risk analysis;
- the benefits cost analysis;
- mechanism selection;
- evaluating the adopted measures security
- risk analysis in general.

Risk evaluation means identifying and classifying the risks that affect the business.

Conducting decision support is identifying and evaluating the control measures and solutions taking into account the cost-benefits report.

Control implementation means implementing and running control measures meant to reduce or eliminate the risks.

Table 5. The levels of IT Application security risk management [9] [14]

Level	Status	Description	Exposure
0	Non-Existent	The company does not have the security policy well documented	Highest
1	Ad-hoc	The company is aware of the risk. The risk management efforts are done in a hurry and chaotic. Policies and processes are not well documented. Risk management projects are chaotic and non- coordinated, and the results cannot be measured and evaluated.	High
2	Repeatable	The company has knowledge about risk management. The risk management process is repeatable but immature. The risk management processes are not sufficiently documented, but the company is taking actions in this sense. There is no formal training or communication regarding risk management, the responsibility being left to the choice of the employee.	Medium
3	Defined	The company adopts a formal decision for implementing the risk management. The objectives and the ways of measuring the results are clearly defined. The employees are formally trained at a base level.	Low
4	Managed	Risk management is well understood in all compartments and levels of the company. There are well defined procedures of control and risk reduction. Efficiency can be measured. The personnel is trained. The allocated resources are enough. The benefits are visible. The risk management team work to permanently improve the processes and the instruments they use. A great deal of the risk evaluation processes, of control identification and of cost-benefits analysis is non-automatic (manual).	Very low
5	Optimized	The organization has committed significant resources to security risk management, and staff members are looking toward the future trying to ascertain what the issues and solutions will be in the months and years ahead. The risk management process is well understood and significantly automated through the use of tools (either developed in-house or acquired from independent software vendors).	Inexistent

Measuring the program's efficiency means analyzing the efficiency of the adopted control measures and checking if the applied controls ensure the established protection level.

Vulnerability means a flaw or weakness in system security procedures, design, implementations or internal controls that could be exercised (accidentally triggered or intentionally exploited) and result in a security branch or a violation of the system's security policy (NIST SP 800-3) [9].

Factors that determine vulnerability are of physical, natural, hardware, software, communication or human nature.

Physical factors that affect the software are the location where the customer has the business. It can be in a wet place, a freezing place or exposed to hot temperatures which leads to hardware malfunctions. Also it can be one of the cases: unlocked or unsecured rooms (where the database server is located);

building design flaws; building construction flaws; insufficient anti-fire system.
Natural environment can be fire, flood, earthquakes, storms, explosions.
Hardware factors are obtained by using improper components without dependencies, or too old components that are supposed to broke anytime.
Software risk is by miss of an anti-virus which leads to a virused and malicious operational system.
Communication risk is obtained by the low network configuration, bad VPN or router security which leads to opened ports that can be exploited, unencrypted communications, active protocols without use, non-filtering the communication between sub-networks.
The human errors are the most common among risks and are due to the fact that operators have lack of knowledge in IT or improper training on using the software; failure to report attacks, weak response to attacks, lack of recovery plans in case of disaster, insufficient procedure testing.

Table 6. Risk Level [9]

Impact/Probability	Low	Medium	High
Low	Low	Low	Medium
Medium	Low	Medium	High
High	Medium	High	High

An Impact/Probability have these values: Low, Medium and High.

Table 7. Exposure [9]

Exposure rate	Consequences	Description	Probability
1	Insignificant	Minor financial losses. No material damages.	40%
2	Minor	Medium financial losses. Low material damages.	20%
3	Moderate	Important financial losses. The activity is carried further.	5%
4	Major	Important financial losses. The production capacity is diminished.	1%
5	Catastrophic	Enormous financial losses. Total loss of the production capacity.	<0.01 %

To ensure a good risk analysis, you have to establish a list of risk level combining the determination of the impact value for goods and estimation of the probability for an event to occur.
The mathematical approach to determine the application risk level RL uses the formula:

$$RL=IR*PR$$

where:
IR = Impact Rate
PR = Probability Rate
As an example we take the Impact of malfunction of a printer with a Low damage caused to the business which has a low probability to appear, thus generating a Low Risk Level.
The Risk level calculated based on the formula from [9] is determined by the Impact Rate = 3 and the Probability Rate = 1 which Result 3 (Low).
$RL_{printer}=3*1=3$

Table 8. Risk Level

Impact – Rate	Probability – Rate	Result	Risk Level
Broke Database – 10	1	10	Low
Broke Report – 4	6	24	Medium
Broke Hardware – 6	6	36	Medium

To adapt our example, we assume that we need a report for a customer and the printer is broke. In this case the Risk Level changes because the Report coefficient with the Printer coefficient together generate a Medium Risk Level:

$$RL_{report} = 4*6=24$$

To calculate the total risk level TRL is used:

$$TRL = \sum_{i=1}^{n} RL_i$$

where i is the risk level of an event (printer, hardware, location, HDD, server etc) and n is the total number of events

$TRL = RL_{printer} + RL_{report} = 27$ which means Medium

To calculate the Annual Loss Expectancy – ALE_t - per Threat, I used the formula:

$$ALE_t = Q_t \sum_{a=0}^{n} V_a$$

where:

ALE_t = Annual Loss Expectancy per threat **t**,

V_a = Value of asset a (0 to n, number of assets),

Q_t = Estimating the number of occurrences of threat t (0 to million threats).

t = threat

Table 9. IT Application Software Risk

Threat	Occurrence times/period	Costs on Server loss in €	Costs on Client loss in €	ALE_S (server) in €	ALE_C (client) in €	ALE (total) in €
Voltage shock high	2 times/year	100	100	200	200	400
Voltage shock low	5 times/year	50	50	250	250	500
Database loss	1 time/2 year	1000	0	500	0	500
HDD malfunction	1 time/year	300	100	300	100	400
Bad Report	2 time/month	10	0	240	0	240
Human Error high	1 time/3 months	50	10	200	40	240
Human Error low	1 time/month	5	1	60	12	72
Total in €						2352

Based on table 11 the total Annual Loss Expectancy is esteemed to 2352 Euro. Some of the Threats are resolved with an initial investment on buying necessary equipment to protect against voltage shocks, replace of old hardware components and more training to employees to avoid the human errors.

6 Conclusions

Every software should have security implemented because without it, the damage that is caused is significantly greater than the cost of implementation.

Secure software is software that enhances trust and respect for the company producing it, is a benefit of image and provides stability both for customers and manufacturer.

To increase the software reliability even more, another approach is to implement a licensing module, which in this case, it secures and improves the stability of the software by creating a serial key and it confuses the potential attackers by making a fake security. By validating the input and output data, it keeps the correctness of the transaction system. The user management implementation protects the software from possible attacks by the employees and avoids potential fraud.

All of these increase the sale of the application and please almost all customers.

A token with the serial on it, with ROM memory, that will make the program run only if it is inserted in the computer will be one of the most protective sources for the application.

References

[1] I. Bica, C. Boja, E. Burtescu, V. Cristea,

M. Muntean. V. V. Patriciu, M. Popa, A. Toma, C. Toma, I. Ivan, *Informatics Security Handbook, 2nd edition*, ASE Publishing house, Bucharest 2009, ISBN 978-606-505-246-8, 663 pg.
[2] I. Ivan, L. Teodorescu, *Software Quality Management*, Inforec Publishing house, Bucharest, 1999.
[3] C. Toma, *Security in Software Distributed Platforms*, ASE Publishing house, Bucharest 2008, ISBN 978-606-505-125-6
[4] A. Freeman, *Programming .NET Security*, O'Reilly Publishing house, 2003
[5] I. Ivan, C. Ciurea, D. Milodin, "Validarea datelor de intrare in aplicatiile informatice orientate spre cetatean," *Revista Romana de Automatica si Informatica*, vol. 18, nr. 4, 2008, pg. 75 – 86
[6] S. Maguire, *Writing solid code*, Microsoft Press, 1993
[7] V. V. Patriciu, I. Bica, M Ene-Pietroseanu, *Securitatea Comertului Electronic*, ALL Publishing house, Bucharest 2011
[8] W. Stalings, *Cryptography and Network Security 3/E*, Prentice Hall Publishing house, 2003
[9] E, Burtescu, *Risk Analysis in Secure Systems*, Course IT&C Security Master. http://www.stonegate.ro/disertation/c01.ppt
[10] I. Smeureanu, *Source Code Programming Security*, Course IT&C Security Master, Available at: http://www.stonegate.ro/disertation/c02.ppt
[11] S Simpson, *Fundamental Practices for Secure Software Development*, http://www.safecode.org/publications/SAFECode_Dev_Practices0211.pdf; http://www.stonegate.ro/dissertation/article_2.pdf
[12] A. Arora, S. Frank, R. Telang, *Estimating Benefits from Investing in Secure Software Development*, https://buildsecurityin.us-cert.gov/bsi/ articles/knowledge/business/267-BSI.html
[13] 15th Annual 2010/2011 Computer Crime and Security Survey https://cours.etsmtl.ca/log619/documents/divers/CSIsurvey2010.pdf
[14] http://en.wikipedia.org/Visual_Basic
[15] http://en.wikipedia.org/wiki/Triple_DES
[16] http://en.wikipedia.org/wiki/MySQL
[17] http://en.wikipedia.org/wiki/ Software_security_assurance
[18] http://en.wikipedia.org/wiki/Obfuscation
[19]http://msdn.microsoft.com/en-us/library/yy6y35y8.aspx
[20]http://www.insse.ro/cms/files/ISI/Societatea%20informationala.pdf

Liviu Adrian STOICA has graduated the Faculty of Economic Cybernetics in 2009. He holds a master diploma in Informatics Security from 2012 is a PhD student at Cybernetics and Statistics at Bucharest Academy of Economic Studies. His work focuses on the analysis of quality and security of software applications.