Certification of Distributed Informatics Applications

Ion IVAN ionivan@ase.ro
Florentin SCHIOPU florentinschiopu@gmail.com
Dragoş PALAGHIŢĂ dpalaghita@gmail.com

The distributed application concept is defined. The efficiency of distributed applications is strictly dependent on their level of quality. The quality requirements and the way to point out the effective levels that the traits distributed applications have are stated. The certification of distributed applications is validated by passing through stages with detailed and well-defined tasks.
Keywords: *certification, distributed application, software quality*

1 Distributed informatics applications

The informational society's base trait is citizen access to the resources of the informatics applications, in order to solve daily matters.
It means that all organizations that interact with citizens must:
- define issues that the citizen may resolve interactively;
- build the vocabulary with the help of which requests are defined by the citizen, and results are given back to him;
- order, buy and implement an applications which, through an interface, takes commands, commutes databases, processes data, demands option selection from decision makers and offers a solution to the demander;
- assure the confidentiality of the request and of the offered solution;
- guarantee data, processing and result quality, to avoid wrong decisions and solutions that don't reflect reality;
- create good security conditions while specific processes are running.

Distributed informatics applications presume the existence of processing functions strictly dependent in terms of number and complexity to the typology of personnel that interacts to unroll the processes that they were developed for, figure 1.

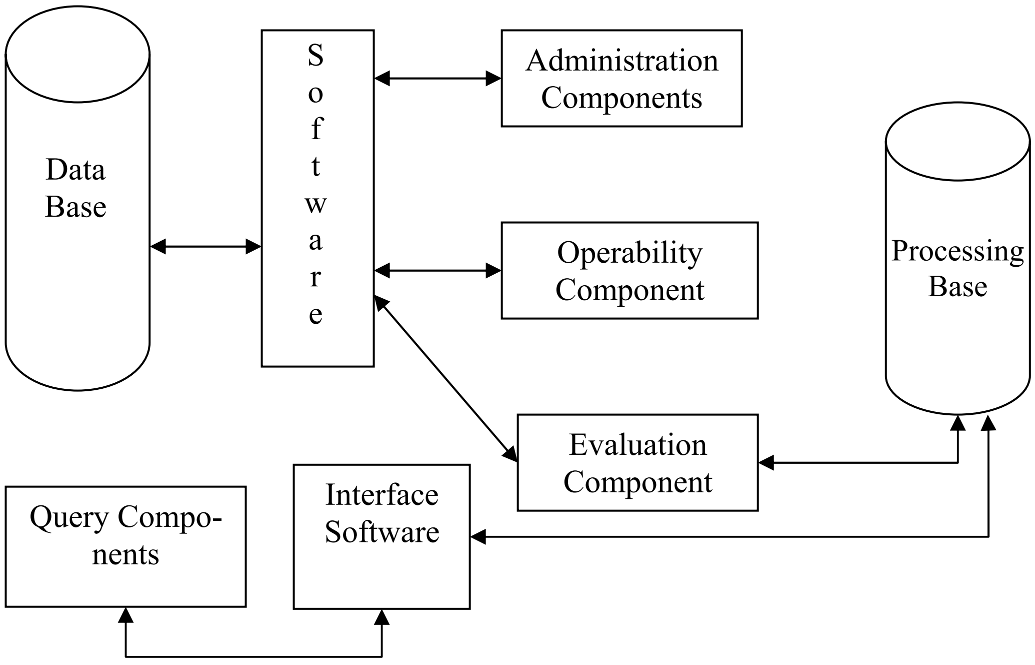

Fig.1. The functional structure of the distributed informatics application

In assisted learning application, the following interfere:
- the subjects that are the object of the training and testing;

- experts that develop the theorems on which the training processes are based: concepts, experiments, classifications, problems, demonstrations;
- independent experts, who develop the testing schedules and evaluate the subjects, offering certificates which show the level of knowledge obtained;
- the application administrators who define the context in which the training processes are evolving and the access to the application's resources;
- application developers who take the specifications and after the application is implemented, insure maintenance;
- people who want to receive information about the functionality of the application and how its resources are accessed; only for this category of users, authentication is not necessary.

Distributes informatics applications, which operate payments and returns in the banking system, include:
- software developers which are keeping sight of the fluidity of structures and processing functions, have an active role after implementation, maintenance being a permanent trait;
- database administrators who secure access to application resources and take on server running, database, and software components tasks;
- system administrators who by combining options secure the execution initially undefined operations who come up on the way;
- operators that enter data into the databases;
- operators that take data to make reports;
- the managerial component that consults the databases;
- clients that consult, make transfers, load accounts and make extractions from the database;
- persons who consult the facilities of services offered by the organization;

Compared to any other type of informatics application, for distributed informatics applications this type of structure must be developed such that the pursued objective will be fully completed. Distributed application certification, in this context, becomes a necessity. Only a certified distributed application is launched in execution, making its resources available to all kinds of users.

The certified distributed application offers the guarantee that serious processing errors will not occur and all the streams defined in the specifications will execute completely and correctly.

2. **Distributed application structures**

Distributed applications with a tree-like structure, which are meant to inform are developed for only one category of users.

There are application administrators and developing operators who maintain the application's content actual.

This type of application is used for training and information on processes, technical fields, organization structures, figure 2.

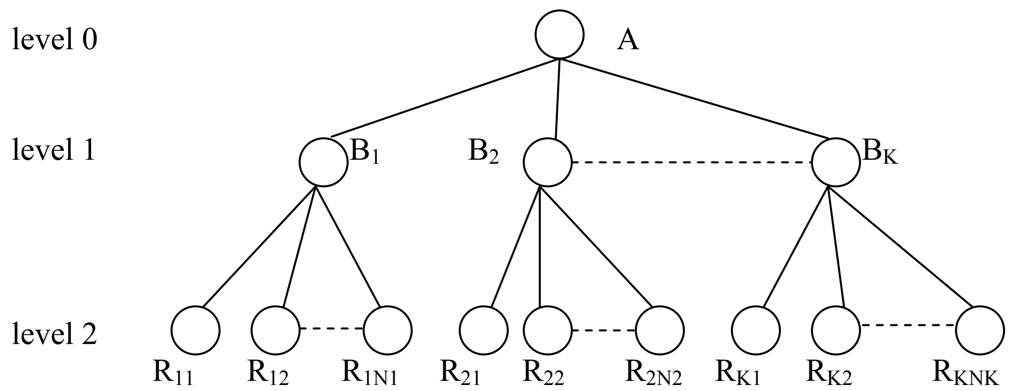

Fig.2. Tree-like structure organized on three levels of information.

The linear distributed application demands a traversing stream, figure 3.

To secure flexibility, forward and backward traversing is provided, figure 4.

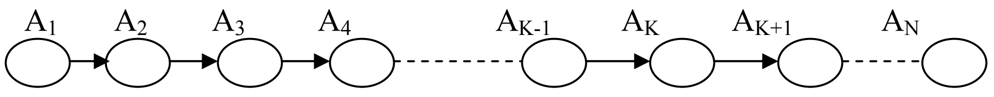

Fig.3. Rigid linear structure

The pure linear structure is rigid; the application user does not have options for retrieving information.

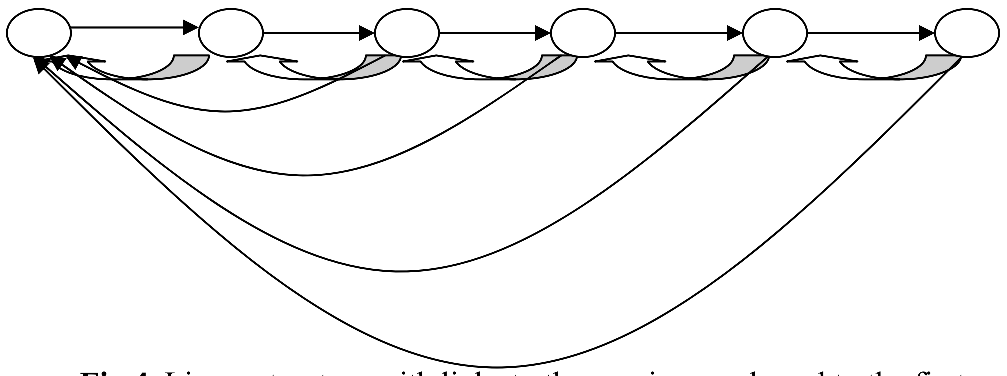

Fig.4. Linear structure with links to the previous node and to the first one

The structure of distributed applications becomes more complex as new links appear between:
- components positioned on different levels;
- components positioned on the same level;
- different parts, which make up a component.

The lack of homogeneity in the structure is given by the differences that come up between the number of component arcs incident to the inside, figure 5.

In addition, the number of arcs incident to the outside, figure 6, influences the complexity and the lack of homogeneity.

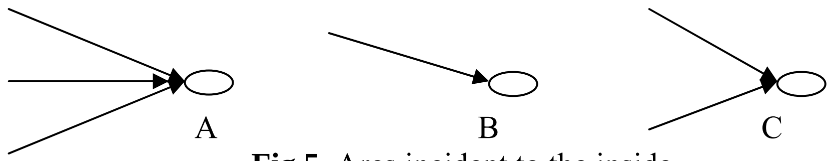

Fig.5. Arcs incident to the inside

The components A, B and C have significant different number of arcs oriented inside.

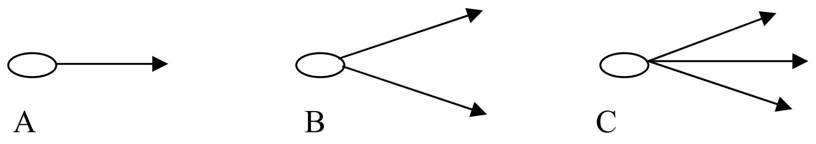

Fig.6. Arcs incident to the outside.

The A, B and C components are different due to number of arcs incident to the outside.

Resource allocation in an informatics application presumes components for:
– resources presentation;
– resources selection;
– processing date regarding the petitioner;
– making the allocation;
– certifying the allocation

These applications are found in ticket reservation, hotel room reservation, road selection, tax payments.

These applications presume option selecting and data input from the user necessary for the allocation.

There are complex supervised training and testing applications through which the succession of applications presumes:
– authentication;
– allocation, followed by access to resources;
– message exchange regarding the fashion in which the user's interaction with the application unfolds.

The structures of distributed informatics applications are influenced by design elements and by the diversity of the means of communication.

3. Distributed informatics applications development specifications

Specifications are texts that are elaborated in the definition, analysis and establishing the target group phases.

The specifications are structured in such a way that:

- the problems to be solved have a clear definition;
- the target group is completely defined and its structure is established, the requirements in report to the application and the means of interacting;
- they clarify what the degree of satisfaction among the users of the application is;
- they are building the vocabulary, message types, and user words, in order to respect the fundamental requests regarding repeatability and text validation given by the user;
- the input data, formulas for calculations and the results that are given to the user are valid;
- there are measures and levels of authentication;
- prefigure how available will de application be for future extensive and deep modifications;
- they contain all the information for the application that must be developed;

The designers, programmers, testers and consultants find all the data to continue their execution operations, such that finally the result is what beneficiaries need to solve their problems.

The objective of the application is developed with the help of the future owner of the application.

The problem that needs a solution is defined through:

- establishing the target group;
- identifying the subprograms specific to each subset from the target group;
- enumerating the access types;
- the sketching of message streams specific to each sub-problem;
- sizing the data that users supply such that defining the stream is exact, direct and concise;
- the structure of the results which must correspond to the requirements of the users;
- the total coverage of the requirements of users with the current manual processing mode available;

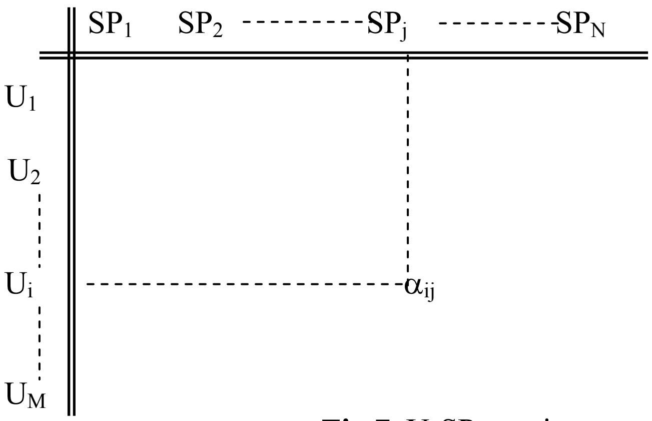

Fig.7. U-SP matrix

The specifications represent the “Bible” for those who develop the distributed informatics application and leave no room for interpretation, being exact, rigorous and with unique solutions. The creativity elements that come from designers, programmers, testers and consultants are strictly related to the resources that they interact with as specialists.

There are already predefined procedures, and the elaborators of specifications highlight them. The designers and programmers know the procedures, which are referred to and take them as they are. The testers and the users know already if the function is performing, as it should, with respect to the given specifications.

The user types $U_1\ U_2\ ...\ U_M$ and the sub-problems $SP_2\ ...\ SP_N$ are established.
Links are made between the users and the sub-problems, thus obtaining the U-SP matrix in figure 7, where $\alpha_{ij} = 1$ if the SP_j sub-problem must be solved for user U_i and 0 in the contrary.
When defining specifications, the following must be accounted for:
- the length of the messages must be minimum;
- the text clarity must be maximum;
- the volume of data entered by the user must be minimum;
- the message exchange must show the transparency of the processing;
- the user must receive all the data to have the certitude that the processing was correct, with respect to the inputted data ant the existing context;
- the reentering of data, and the repeating of random steps must be excluded;
- the reason for every step in the processing stream must be based on clear and explicit aspects, as the legal or procedural base should be.

In this context, a correct basis is developed to develop an application that will lead to maximizing the satisfaction level of the beneficiary, user, client, operator or even administrator of the application.

4. Quality traits of the distributed informatics application

The quality traits like: reliability, maintainability, portability, usability, correctness, implemented in classical applications, are used in distributed applications, with some adaptations that regard:
- multi-user platform;
- working with multiple computers;
- simultaneous processing;
- the risk of unauthorized access;
- lack of user homogeneousness;
- the diversity of user requests;
- real time non-stop usage;

Added to the quality traits above, new ones specific to distributed applications appear.
Continuity permits the use of knowledge accumulated by the user, through access to other well-known informatics applications. Key words and actions that are triggered are taken from different other applications, such that the change to a new distributed application is natural, without failed attempts to reach the defined objective and for which the application is activated.
Confidentiality supplies the measure in which the data, the operations and the results of the interrogation made by a user remain known only to him. Confidentiality is obtained through the data access right given to the user and to the user data management officer in conditions of strict simultaneity over the fields regarding essential operations.
Security is the trait through which a guarantee that the databases are a subject only for the operations describes in the specifications is made. The application has powerful components that ensure protection against all operations except the ones described in specifications. Security can be considered as non-vulnerability.
Homogeneity consists of ensuring a way to traverse the level that structures the application. All nodes have the same number of arcs incident to the inside and the same number of arcs incident to the outside.
It also regards the modalities of communicating with the use of text. A message is displayed. The answer of the user must be given every time using the same modality. If the use of value lists everywhere has been decided on, the user will request his options or he will enter values by selecting list elements. Any other working method will avoid, as much as possible, the combining of data input modes and receiving values from the keyboard would be used only as a last resort, because it can cause errors. In this case, the validation components must be very strong.
Simplicity must be taken into consideration because a tendency to complicate and to maximum generality exists and distributed application developers lean to it. The best options are the simple ones. Breaking the problem into sub-problems, using a three-step method to get to the desired result are ways of keeping things simple.
If a correct regrouping of situations is made,

the objective will be reached. The task of finding key words in a non-homogenous set is very difficult.
The types of users are enumerated. After that, the operations are enumerated.
If the operations are essential, they are combined with operands, with the owners of the transactions. Simplicity is foreseen when the information is diffused, fragmentations are made and even comebacks.
Attractivity is a quality trait, which is obtained through the advantages that it presents: eliminating waiting in line at the counter, eliminating traveling, insurance that the operation will be completed, and quality of the service.
An application is attractive if it lets the user come to it. In the case of obligatorily the rejection phenomenon will appear, it will increase in intensity if the application is hard to use or badly developed.
If the user is given the application as another means of completing the desired transaction, allocation or information process, coexisting with the old ways, systematically the application will substitute the traditional means, without excluding them.
Completeness presumes completing the processing with the help of streams for the entire transaction. The so-called virtual stores that impose the client to make the payments to the bank, and not on-line, already represent mixed forms, hard to accept.
Completeness presumes taking control of operations that are already being executed non-interactively human-computer, eliminating waiting and evaluating new facilities, such that the user will perceive the application as a solution to his benefit. An on-line training course without testing and without certification is exactly an incomplete and unnecessary solution.
Efficiency must lead to obtaining advantages both for the user and for the supplier.
If the application creates problems for the user caused by the incorrect way the database was created, the discomfort leads to avoiding the application. Under these conditions, to maintain the standard, the service suppliers loose money when compensatory payments are grater then the risk limits assumed in the conditions of guaranteeing a profit margin.
The specifications highlight, for every trait the concrete modalities for it to be noticed in the testing process.

5. Informatics applications metrics

It is an error to believe that metrics are built exclusively to measure software quality or distributed informatics applications quality, which includes besides software, data and communication and transmission processes.
To build a metric means:
- to write formulas for quality traits indicators, process activity, transactions, behavior, satisfaction, necessity, efficiency, risks and durations;
- to verify the measure in which the indicators are non-compensatory , metaphysical and representative sensors;
- to validate indicators by controlling the measure in which they put sub-intervals into correspondence with the quality levels very good, good and unsatisfactory;
- to elaborate software for automated collection of data an value calculation;
- to build up trust in using quantitative methods for estimation, prognosis and mostly to measure al the sides of the distributed informatics application’s life cycle for decision making;
- to accept quality standards for resources, processes and products as the only way to guarantee that finally the obtained result is the desired one.

For the same elements that concur to the development of an informatics distributed application a number of metrics is defined, each with its advantages and disadvantages, which are incorporated objectively.
The indicators that measure sides of the behaviour play an important role. They take the following form:

$$I_1 = \frac{NS}{NT}$$

where:
I_1 – behavior indicator;
NS – number of successful operations;
NT – the number of all operations;

Or:

$$I_2 = \frac{\sum_{i=1}^{n} d_i}{\sum (d_i+w_i)}$$

where:
n – number of completed operations
d_i – duration of operation i;
w_i – duration of a result in operation i
Or through complexity indicators C, defined as:

$$I_3 = C = m - n + 2$$

where:
m – number of arcs (operators);
n – number of nodes (operands);
or the complexity given by:
$I_4 = C = (n \log_2 n + m \log_2 m) / (m+n) \lg_2 (m+n)$
The following relation gives the degree of customer satisfaction:

$$GS = \frac{\sum_{i=1}^{k} f_i}{NT}$$

where:
k – the number of leafs from the tree-like structure associated with the application;
f_i – the number of users that have finished processing in the terminal node of associated with the leaf i;
NT – total number of users
It's noticed that:

$$NS = \sum_{i=1}^{k} f_i$$

when the GS indicator is of type I_1.
If the structure from Figure 8 is considered, where the number of nodes is n = 7, and the number of arcs m = 20.

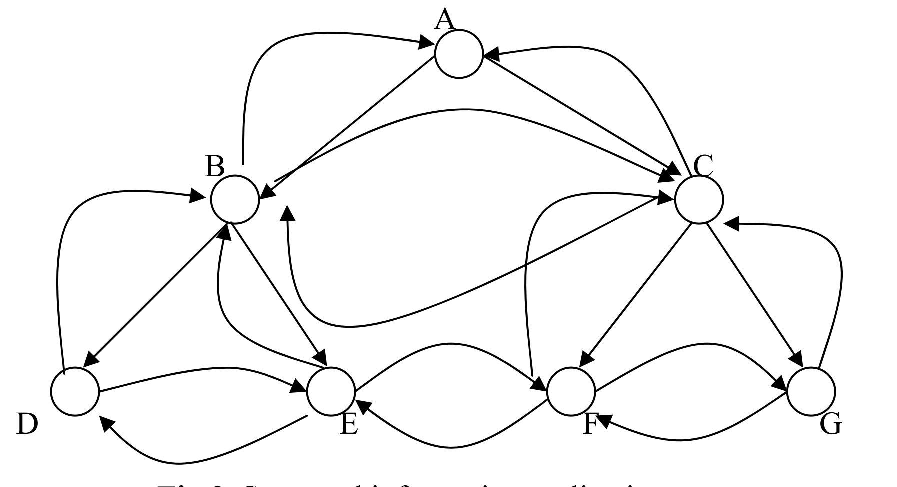

Fig.8. Saturated informatics application structure

$I_3 = C = 20 - 7 + 2 = 15$
Indicator

$$I_4 = \frac{20 \log_2 20 + 7 \log_2 7}{27 \log_2 27}$$

shows that through its value equal to 0.826 that the application structure has a high level of complexity.
It is the job of development organizations to decide if they want to implement metrics and prove the level of performance of the products that they are creating.

6. Certification processes

At the beginning of certification there are:
- the developing specifications that answer the question "What must be realized?"
- the software product, the database, the results that show what was effectively realized;
- the metrics that measure the quality and behaviour traits, consumes, durations, expenses and other quantitative traits;
- certification procedures;
- independent organization with specialists who can run objective certification processes;

If, by analysis, the measure in which between the informatics application and the specifications there is concordance is computed, in case of the certification:
- it is established that the distributed informatics application executes processing according to specifications;

- the tests from the specifications are executed and supplementary ones also to see the measure in which risk processing, restricted processing and processing that is not according to specifications is made;
- it is a guarantee that in normal conditions, defined in the specifications, by entering the correct data the results will also be correct;
- the levels of quality traits are measured and of al other traits to see if they are identical or do not differ significantly from the effective levels measured by the developer;

Distributed informatics application certification includes processes through which the behaviour is highlighted:
- at central level, server or database level;
- locally, for the maximum number of workstations, in simultaneous working conditions.

In the certification process, tests linked to access and application exploitation security are made. The measure in which the application is vulnerable and is resistant to resources access is computed.

For certification, the following steps are followed:
- the objectives of certification are planned;
- the resources used in this process, the costs and the durations are established;
- the application is taken into custody;
- the documentation is taken into custody;
- the documentation is analyzed and an intermediary report is made;
- behaviour measurements are made and an intermediary report is built;
- the application is tested using the sets from the specifications;
- supplementary tests are made by the certification team;
- the intermediary reports are analyzed;
- the final certification report is completed.

The final report is accompanied by a report, which concludes that the distributed informatics application executes what is stated in the specifications, and is a guarantee that when correct and compete data is inputted, the results will also be correct and complete.

The certification is a complex process that presumes:
- that the unfolding procedures exist and are respected;
- stages, tasks and specialized staff in operation execution;
- total independence from the developer of the product;
- professional probity;
- inputs and outputs for every operation;
- absolute reproduction ability for each operation;
- the capacity to verify the way each stage has unfolded;
- absolute transparency to offer the developer the requested information to show that the certification process went according to standards;
- the position consolidation of the certification company through the fact that the certified product works well indeed, and there are no questions that will spread doubt upon the certification;
- the rising of the level of objectivity by including more elements based on observations that are not questioned, in any moment it can prove that procedures were respected;
- the acceptance of the measuring procedures and results obtained from the measurements, by the distributed informatics application developer;
- making an agreement between the products an standards used and the existing application typology, such that the certification will cover all aspects linked with processing, interfaces, security and user qualification level;

The positive outcome of the certification process is accompanied by the release of a document – certificate in which the certification type is specified and in which domain the certification is guaranteed.

The final report contains: a text regarding the product under certification, the stages of the certification process and the certification team; annexes regarding the certification; the released certificate if the product is corresponding and a guarantee that it operates like stated in the specifications. Certification is the team activity that is independent from the application developer. If the application is altered after every development process, there must be a certification. The certification is obtained for a limited period.

7. Conclusions

For distributed informatics applications, certification is different from other tips of applications:

The objective of certification is to guarantee that:

- the processing is according to specifications;
- the access to resources is obtained only in given conditions;
- unauthorized access risk is limited;
- there is a strict delimitation of attributes such that the forbidden operations don't run no matter of the direction from where the attempts are made;
- the executed operations contain complete information regarding the moment, type, operator and computer on which they were deployed;

Certification remains the only way through which the guarantee that the distributed informatics application launched in execution is operational fulfilling the requests defined in the objective.

Only a certified distributed informatics application creates the premises of a high level of satisfaction to a very large set of users.

Bibliography

[BALO99] Balog Al. - *Evaluarea proceselor software*, Editura I.C.I., Bucureşti, 1999

[BRAK98] Wilhelm Brakhahn, Ulrike Vogt – *ISO 9000 pentru servicii*, Editura Tehnică, Bucureşti 1998

[BURA01] Sabin Corneliu Buraga – *Tehnologii Web*, Editura Marix Rom, Bucureşti 2000

[COST99] Costache D., Avram G., Moise M., Sonea E. ş.a., - *Manualul calităţii* AISTEDA, Editura AISTEDA, Bucureşti, 1999

[FREN94] Militon Frenţiu, Basil Pârv – *Elaborarea programelor. Metode şi tehnici moderne*, Editura Promedia, Cluj 1994

[GHIL01] Bogdan Ghilic-Micu - *Aspecte ale impactului trecerii la societatea informaţională*, revista "Informatica economică" editată de Catedra de Informatică Economică şi asociaţia INFOREC, cu sprijinul Ministerului Educaţiei si Cercetării volumul V, Nr. 3 (19)/2001 (pag. 10 - 15)

[IVAN99] Ion Ivan, Paul Pocatilu – *Testarea software Orientat obiect*, Editura Inforec, Bucureşti 1999

[IVAN99] Ion Ivan, Gheorghe Noşca, Sebastian Tcaciuc, Otilia Pârlog, Răzvan Căciulă – *Calitatea datelor*, Editura Inforec, Bucureşti 1999

[IVAN01] Ion Ivan, L. Teodorescu – *Managementul calităţii software*, Editura Inforec, Bucureşti 2001

[IVAN01] Ion Ivan, Paul Pocatilu, Mihai Amitroaie - *Metrici ale societăţii informationale*, revista "Informatica economică" editată de Catedra de Informatică Economică şi asociaţia INFOREC, cu sprijinul Ministerului Educaţiei si Cercetării volumul V, Nr. 4 (20)/2001 (pag. 33 - 40)

[Meye88] B. Meyer – *Object Oriented Software Construction*, Editura Prentice Hall, Englewood Cliffs 1988

[McCa77] J. McCall – *Factors in Software Quality*, Editura General Electric Co., 1997

[MOŢO00] Radu Moţoiu – Ghid pentru managementul calităţii şi calitatea managementului anilor 2000, Editura FiaTest, Bucureşti 2000

[ROSC01] Ion Gh. Roşca, Floarea Năstase, Victor Patriciu, Octavian Paiu, Gabriel Şutac, Carmen Stanciu, Ion Bica – Cryptographic methods and techniques for authentication through digital signature in e-comerce. Juridical controversies and national approach, Information Society, The proceedings of the fifth international symposium of economic informatics may 2001 Bucharest 10-13 may 2001, Editura Economică, Bucureşti 2001 (pag. 345-352)

[SABĂ98] Gheorghe Sabău, Cristina Ioniţă, Georgeta Bădescu, Vasile Avram, Petrişor Oprea, Claudia Cârstea - *Baze de date relaţionale - Aplicaţii în turism*, editura CISON, Bucureşti, 1998

[SOAR96] Elena Soare, Al. D. Colceru – *Certificare calităţii*, Editura Tribuna Economică, Bucureşti 1996

[TRICK99] Ray Tricker – *ISO 9000 pentru întreprinderi mici şi mijlocii*, Editura All Beck, Bucureşti 1999

[VASI92] Gheorghe Vasilescu, Alexandru Balog – *Sistemul caracteristicilor de calitate ale produselor program - referat*, Academia de Studii Economice, Facultatea de Cibernetică, Statistică şi Informatică Economică, Bucureşti, noiembrie 1992

[IEEE94] IEEE Standards Collection, *Software Engineering*, Published by the Institute of Electrical and Electronics Engineers, Inc., 1994 edition;

[****96] Key Computer Consultants, Inc., *How to Write a Software Specification,* http://www.worthy.com/~kcc/kccspec.html Key Computer Consultants, Inc., 1996

[******] Functional Specification http://www.pcwebopedia.com/term/f/functional_specification.html,

[******] Requirements definition/specification http://www.crab.rutgers.edu/~ssykes/ch4/tsl006.html

[******] Software requirements Specification http://www.esi.es/Help/Dictionary/Definitions/Software_Requirements_Specification.html

[******] Software Engineering Principles http://www.eecs.wsu.edu/~sheldom/mms/seslides/

[****96] Key Computer Consultants, Inc., *How to Write a Software Specification,* http://www.worthy.com/~kcc/kccspec.html Key Computer Consultants, Inc., 1996

[******] Requirements definition/specification http://www.crab.rutgers.edu/~ssykes/ch4/tsl006.html

[******] Software requirements Specification http://www.esi.es/Help/Dictionary/Definitions/Software_Requirements_Specification.html

[******] Software Engineering Principles http://www.eecs.wsu.edu/~sheldom/mms/seslides/