JOURNAL OF APPLIED QUANTITATIVE METHODS

Reliability and Quality Control – Practice and Experience

EVALUATING THE EFFECTS OF THE OPTIMIZATION ON THE QUALITY OF DISTRIBUTED APPLICATIONS

Eugen DUMITRASCU[1]

PhD Candidate, University Assistant, Computer Systems and Communications Department,
Faculty of Automation, Computers and Electronics
University of Craiova, Craiova, Romania

E-mail: eugen.dumitrascu@cs.ucv.ro

Marius POPA[2]

PhD, University Lecturer, Economic Informatics Department
Academy of Economic Studies, Bucharest, Romania
PhD Dissertation Title:
Methods and techniques used in the qualitative evaluation of the data (2005)

E-mail: marius.popa@ase.ro

Abstract: *In this paper, we present the characteristic features of distributed applications. We also enumerate the modalities of optimizing them and the factors that influence the quality of distributed applications, as well as the way they are affected by the optimization processes. Moreover, we enumerate the quality characteristics of distributed applications and a series of evaluation's indicators of quality for varied structures of distributed applications.*

Key words: *optimization; quality; distributed applications*

1. Distributed applications

A distributed application or a global application implies the access of data from many nodes of a computers network. The components are being executed on different nodes, on different platforms that are connected to the network.

The term "distributed application" has three aspects:

- the application A, whose functionality is divided in *n* components, A_1, A_2, …, A_n, $n \in N$, $n>1$ that interact and cooperate together; each component is a distributed application or a process;
- the components A_i are autonomous entities that run on different computers;
- the components A_i change information through network.

The distributed applications are those in which many beneficiaries or users that are in different points of territory, access definite resources for computer network to solve a problem. The modern conceptions for banking transactions, inter-banking transactions, realization of e-commerce activities, training activities, informing activities, testing of knowledge activities, concluded the on-line contracts, these are just few of the distributed applications that must characterize the information society. Development philosophies for e-

learning, e-government, e-business, for virtual organizations and the implementation of new work forms are based on the principles of distributed applications.

The particularities of distributed applications are:

- strong interfaces that permit using them by a very diverse number of citizens;
- high generality degree that permits large number of persons to solve their own problems;
- friendly interfaces that permit the elimination of input data errors and the abandon of utilization;
- levels of security which guarantee that the system of transactions is operational;
- levels of access that convenient resolve the problem of security with the problem of transparency;
- high level of correctness and reliability;
- the guarantee for recording sufficient information that give the possibility to reconstitute the information route;
- the components of any distributed application contain two important parts: application part and communication part; some components contain a special part named administrative part with control role and manage role of components;
- high degree of modularity and the possibility of extensibility through addition or elimination of some software or hardware components;
- the possibility of many more users to share the resources;
- a large availability in case of fault of some components;
- fault tolerance.

The importance of distributed applications is bigger and bigger in the information society. In these days almost any application is realized distributely.

2. The optimization of distributed applications

The optimization, in most general meaning, is the process of modifying a product for bringing it in a more important form in relation with one or many estimation criteria and makes that product the best from the technical, economical and social point of view, in comparison with other products that are part of same category.

The aim of optimizing software products is to make them faster, more efficient, more reliable, easy to use, ease to maintain, that occupy little space on disc and consume few resources.

The software optimization is the art or the science to modify a software program, to become efficient, to use one or few computer resources, few memory, to occupy little space on disc and the run time of operations to be short.

The optimization of programs represents one of the directions towards which the implementers of software products orientate in parallel with growing the complexity and the variety of these products. There are many aspects of this process on the programs, among which we mention the following:

- minimizing the time of execution (or run time);
- optimizing the size of operands;
- optimizing the source code;
- minimizing the assignation of resources.

The optimization is the concept according to which, from set of possible solutions to a problem, is chosen that solution that verifies a performance criterion.

There are two important categories of optimizing programs from the point of view of implementing algorithms:

a) using a better algorithm, that means to replace an algorithm with a better one, but that does not prevent the understanding or the ulterior modification of the program;
b) improving the implementation of the algorithm that is already used, to favor the particularities of the framework in what the program runs or in favor of the characteristics of programs' data; the optimization leads to a difficult understanding or to a modification of the program in the future; it also, decreases the portability of the program on different hardware or software architectures.

A rule of optimizing programs that is considered as the "first law of optimization" is not to try optimizing the program, strictly speaking of modifying an existing program, until it works correctly.

The problem of the minimum corresponds to the situation in which the performance criterion leads to the choice of a solution from possible solutions, for which that criterion has the lowest value.

The problem of maximum corresponds to the situation in which the performance criterion leads to the choice of a solution from possible solutions, for which that criterion has the biggest value.

The single-criterion optimization presupposes choosing from a set of criteria a single criterion and extracting from possible solutions, that solution which satisfies best that criterion.

The multi-criterion optimization presupposes extracting that solution that satisfies simultaneously many criteria of performance.

In the case of distributed applications, to optimize means to define a set of distributed applications' structures or a set of variant distributed applications which differ by the:

- disposition of the elements;
- topologies and modalities of connection;
- allocated resources for construction elements;
- levels or layers from architecture;
- implemented functions of processing;
- level of quality of the obtained characteristics;
- cost of realization;
- duration of realization.

and the choice from this finite set of that distributed application that satisfies the considered criterion of performance.

If a new variant will appear, we will go further into evaluating the criterion of performance and setting a new optimal solution. There are situations in which a new variant doesn't change anything, the old optimal solution still remains optimal.

Criteria of optimum:

- the minimization of prices;
- the maximization of the degree of satisfying the users;

- the minimization of the transaction's duration;
- the maximization of the variety of the clients' requests.

3. Quality characteristics of distributed applications

The quality characteristics are emphasized in different stages from life cycle of distributed application. These characteristics are general characteristics and special characteristics. Two special characteristics of distributed applications are synchronization and integration.

The general characteristics of a distributed application are:

- correct functionality of distributed applications' components in a securitize and interoperable mode;
- reliability that maintains the level of application performance in given conditions and for a long period of time. The based attributes of these characteristics are: fault tolerance and recoverability of date affected from diverse errors of application;
- usability from different users;
- efficiency of application given by time behavior and used resource behavior;
- maintainability that refers to the effort needed for certain modifications;
- portability that permits the application to run on different systems;
- interoperability with other distributed applications or systems;
- studied complexity in correlation with other characteristics as reliability, stability or maintainability;
- flexibility in special case of web distributed applications. This flexibility, from the web server for databases point of view, is given by the capacity to incorporate data from accessed databases in web pages;
- security that offers a safer way of information in network, using cryptographic support, read/write rights, protected access by password, etc.

Nowadays, distributed applications refer more and more to the mobility characteristics of users due to exponential progress of mobile communication technologies.

The mobile communications technologies facilitate the interchange of data between users, irrespective of their geographic location. A special importance is given by the form of data represented, form that is determined by the storage capacity and the processing of the mobile devices.

For instance, the exchanged information between the members of a project team has diverse formats of representation. The format of representating the information has an important role for interpreting the data by the members of that team project. In order to offer a content of high quality, the security models and templates were developed taking into consideration the mobile multimedia content [BOJA06][3].

The new forms of communication between users are the result of the technological progress over the last decade. The personal digital assistances, computers with wireless technology help us to realize an improvement in the real time management. The decisions in real time presuppose a good communication between the team members on basis of the represented information in adequate format [BOJA06].

The use of multimedia format is the result of the last evolutions in the information technology domain. The security of the multimedia content implies the progress of security models and templates.

Digital Right Management (DRM) is a specification that designate a set of standards for certain characteristics of business and management models. The use of the media object downloaded from a server is inspected by the providers and by the operators of informational content. The mobile management of projects implies the use of mobile devices. The communication technology is provided by a third organization that has to ensure a high level of quality for paid services by the owner of the project for mobile management [BOJA06].

We define the rules for using the media objects. A single media object has associated with it different rights with different prices. Digital Right Management sells rights to use the media objects and doesn't sell the media object itself.

There are two ways to provided the rights to users, [www1]:

- the delivery with the media object;
- sending rights separately in the media content.

In table 1, we present the media types MIME (Multipurpose Internet Mail Extensions) for objects, regarding their representation of Digital Rights Management message format [BOJA06].

Table 1. MIME media types

DRM Methods	Media MIME types
Forward-lock	application/vnd.oma.drm.message
Combined delivery	application/vnd.oma.drm.message application/vnd.oma.drm.rights+xml
Separated delivery	application/vnd.oma.drm.rights+xml application/vnd.oma.drm.rights+wbxml application/vnd.oma.drm.content

The message Digital Right Management is based on a composite MIME type in which one or many objects are combined into a single one.

The complex system of communication and transfer has the following characteristics:

- processes the important quantities of data;
- takes the decisions in short time;
- communicates the information, the results and the decisions in short time between the users;
- simulates the complex situations;
- estimates the final or partial results.

Alongside with the satisfaction of the quality requirements by ensuring the quality characteristics of the mobile applications mobile, the profit and the time are two factors that emphasize the success of implementation of the distributed applications using the newer technologies of the Information Society.

4. Indicators for estimating the quality of distributed applications

We define a set of quality indicators for distributed applications. These are different from a define structure to another by the specific characteristics that each structure has.

a) Quality Indicators for client-server applications

There is a set of indicators that estimate the quality of these applications, these indicators are determined by computing them by the server application or by the client application, such as:

- the number of servers (I_{NS}) indicates the number of server applications, in case a client-server application uses more servers;
- the number of clients (I_{NC}) connected to server, or the number of different clients that access the server application;
- maximum number of clients (I_{NCmax}) that the server supports, in case a server application has a limit in that meaning;
- the completeness of a server (I_{CS}) is a measure that shows the coverage of connected clients, facing a maximum number of clients

$$I_{CS} = \frac{I_{NC}}{I_{NC\max}}$$

- the completeness of distributed application is a measure given by the sum of the completenesses of each server, in case the application has more servers

$$I_C = \sum_{i=1}^{I_{NS}} I_{CS_i}$$

- the number of logical correlations (I_{CL}) between client and server information;
- the number of changed messages (I_{MCS}) between client and server, in some cases it measures the number of changed messages between clients (I_{MCC});
- the number of client's requests to server (I_{CCS});
- the number of responses given by server to client (I_{RSC});
- average time life cycle of application is the fraction between work length (D_L) of an application and the number of fault of application (*nc*)

$$T_V = \frac{D_L}{nc}$$

- the security level of application is given by the number of attacks (*na*) of different application in time unit (I_{GS})

$$I_{GS} = \frac{na}{\Delta T}$$

- the portability degree (I_{GP}) of application is given by the number of systems that the client-server application runs.

b) Quality indicators for multi-level applications

These indicators are the same with those indicators described for client-server applications and where we only add a few indicators for the others levels of application. Like

the previous model, these indicators are computed by the server application or by the client application. These supplementary indicators are:

- the number of changed messages between client and the intermediate level or levels (I_{MCI});
- the number of changed messages between server and the intermediate level or levels (I_{MSI});
- the number of client requests to intermediate levels (I_{CCI});
- the number of responses given by the server to intermediate levels (I_{RSI});
- the number of transactions (I_T) in time unit realized by the server level with storage data level.

c) Quality indicators for web applications

There are 14 indicators or metrics for this type of application: subject, proportion, depth, cohesiveness, accuracy, source, maintenance, frequency, availability, authority, presentation, information-to-noise ratio (useful information), writing quality, popularity. From this list the following 6 as being widely used:

- frequency – indicates how recent a page was updated, is measured as the time past from last update or modification of the document

$$\nu = T_i - T_{ua}$$

where, T_i = initial time

T_{ua} = the last update time

- availability – indicates the number of incomplete (broken) links on the web page, is calculated like a fraction between these links and the total numbers of links it contains from page

$$Disp = \frac{NLI}{NTL}$$

where, NLI = the broken links

NTL = the total numbers of links

- information-to-noise ratio (useful information) – we measure the useful information content from the web page for a given dimension, and it is computed as the total number of tokens divided by the size of the document

$$IU = \frac{NTS}{Dim}$$

where, NTS = total numbers of tokens after processing

Dim = the size of the document

- authority – indicates the reputation of the organization realized by the web page
- popularity – indicates the number of links of other sites that are mentioned by this page
- cohesiveness – indicates the degree in which the page content is emphasizeded by a certain theme

d) Quality indicators for e-commerce application

The indicators for estimating the quality of e-commerce applications are:

- the authentication grade indicates the number of information requested for authentifying a client: username, password, last name, first name, address, the card number, card type, the data card expiry, etc. For facilitating the authentication, we use for example unique codes for identifying persons like personal numeric code;
- the security degree is given by the number of users that accessed the application without passing through the authentication phase or those who used false authentication information. This indicator must be null for secured application;
- the degree of data's actuality;
- the number of transaction in time unit (hour, day, week, month, year);
- the number of sells in time unit;
- the number of transactions done by a certain person or user in the time unit.

The last three indicators reveal clearly the quality of application because an e-commerce application is qualitative if these indicators are bigger. This means that the application was accessed by many users and the products or commercial services are of good quality. Also, it does comparative statistics on long periods of time and for that it sees if the application improved the quality or not on that period. In this way the indicators must be stored and archived on long periods of time.

e) Quality indicators for mobile applications

The quality of these applications is estimated by the indicators or metrics for distributed applications on multi-levels. A mobile application is considered a client-server application on three levels: on first level it is the client application that works on portable devices and that is a graphic interface, a many times web interface, based on WAP protocol, the second level or middleware level contains applications that run on different servers for instance web servers, or a mobile application server, the third level contains the storage data level or diverse databases.

We enumerate some indicators for estimating a quality of mobile applications:

- the graphic quality of applications that refers to the displayed mode of a user interface of a mobile application. In this domain a series of standards, protocols and even virtual machines for mobile applications have appeared;
- the degree of communicating with a remote server is given by the number of accesses in the time unit of mobile client application to server. The mobile applications didn't have to be connected permanently to a network and that is why we measure only the number of the accesses to the network;
- the access time at the wireless network;
- the synchronizational degree with different and varied mobile applications.

5. Experimental results

We will present some experimental results on a distributed web application, used in identifying persons. The application searches a person in many distributed databases and on the same server or on the different remote servers.

The identification of the person is done by the first and last names or by Personal Numeric Code (PNC). In the cases where there are many persons with same first and last names the identification is not unique and for that it is efficient to search by PNC, that is a unique code and represents the primary key in any table from a database with persons.

Dealing with a web application, the client is a web interface that connects to a database for searching persons.

For a good understanding of the usefulness of this application, let us give a real example. Let's suppose you want to search a person into a database from a city or a county. If that person is not in this county or city the search won't have any results. In this case, it searches in the databases from others counties. So, it is clear that we need a distributed application which a client, for instance a web browser, can access, by web server, other databases from other servers located remotely. In this case, all databases must contain at least a table with the same structure or similar with that table with persons from the local database, to permit the client to send the data that he asked for, for instance the personal data of a person like: PNC, First name, Last name, Sex, Birth data, Address, Email, Phone.

The structure of the application is represented in figure Fig.1.

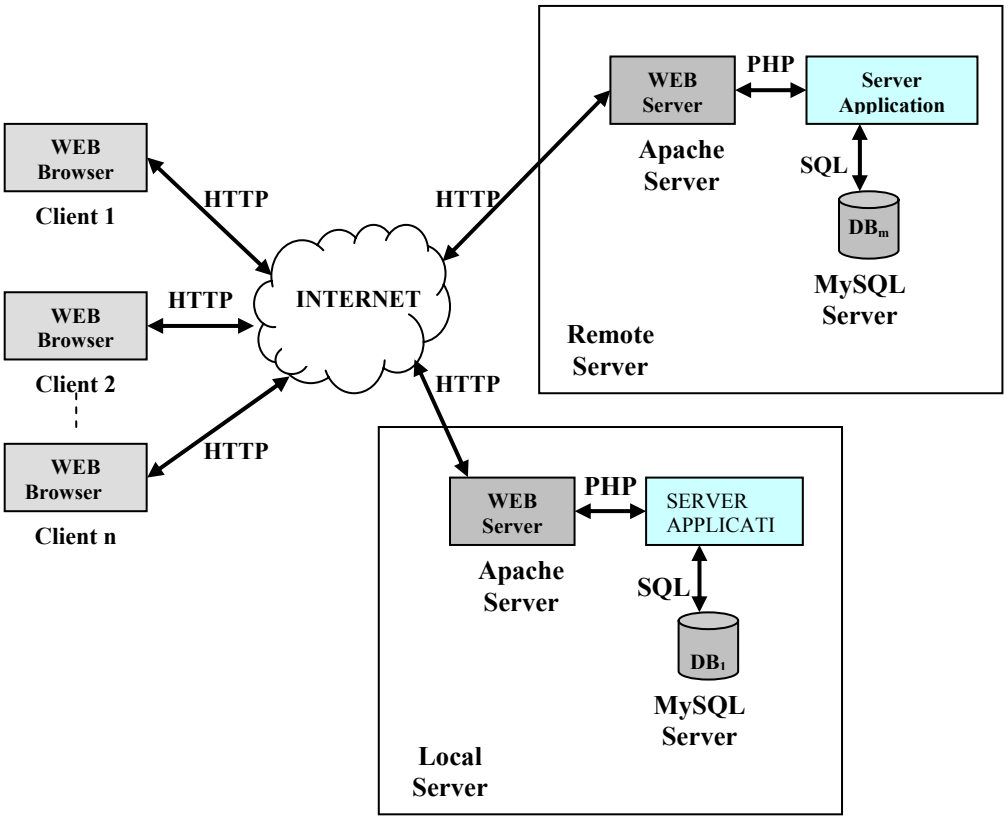

Figure 1. The structure of a web application for identifying a person

We measure the afferent indicators for this type of application. This application is a multi-level application.

$I_{NS} = m$

$I_{NC} = n$

I_{NCmax} = maximum number of clients that the server supports

The application is tested on Windows and Linux operating systems using diverse browsers as Internet Explorer, Netscape or Mozilla, so, the portability degree for server is $I_{GPserver} = 2$ and for client is $I_{GPclient} = 2$.

The tests are realized for 4 servers therefore m=4 and for 10 client stations so n=10.

$I_{NS} = 4$

$I_{NC} = 10$

It is considered for each server $I_{NCmax} = 10$.

In the maximum case of having connected all clients:

$$I_{CS\max} = \frac{10}{10} = 1$$

In the minimum case of having connected a single client:

$$I_{CS\min} = \frac{1}{10} = 0.1$$

a) The single-criterion optimization

We consider a single quality characteristic for measuring an indicator namely the number of transactions realized on a time unit. From here, we derive the transactions volumes on that time unit for each server. The servers have different configurations. We note those with S_1, S_2, S_3, S_4 the used servers.

The servers will be ordered by the volumes of transactions performed in a time unit noted UT. The server with maximum level will be chosen for identifying the optimal server, from the realized transactions point of view.

	S_1	S_2	S_3	S_4
Number of transactions in time unit UT	100	345	215	300

After the ascending ordering, we will obtain:

	S_1	S_3	S_4	S_2
Number of transactions in time unit UT	100	215	300	345

It is observed that the most performed server is S_2 who realized the greatest number of transactions on the time unit UT.

b) The multi-criterion optimization

We test the identification of a person using many data for selecting him/her from the databases. The selection criteria are by PNC, by last name, by last name and first name. We identify the optimal searching technique with a minimum number of results. The tests

will be done by many clients, each of them wanting to find the same person using different criteria.

The client C_1 searches the person by PNC, the client C_2 searches the person by the last name, and the client C_3 searches the person by the last name and the first name. In the table with the results, it will be written the number of results found in the databases. If a criterion is not used, we write -.

Server	S_1			S_2			S_3			S_4			Total no. of results
Sel.crit. / Client	PNC	Last Name	Last Name and First Name	PNC	Last Name	Last Name and First Name	PNC	Last Name	Last Name and First Name	PNC	Last Name	Last Name and First Name	
C_1	1	-	-	0	-	-	0	-	-	0	-	-	1
C_2	-	20	-	-	10	-	-	2	-	-	7	-	39
C_3	-	-	10	-	-	1	-	-	0	-	-	1	12
Min	0	0	0	0	0	0	0	0	0	0	0	0	1
Max	1	20	10	0	10	1	0	2	0	0	7	1	39

We observe that for finding a person in the databases, we need to search by the field that identifies that person as unique, for instance the PNC. In this case, if the value PNC is valid and existing, then there will be a single result. In the case of searching fields that do not identify in unique mode a person, it is better to choose more fields of selection simultaneously to have a minimum of results.

6. Conclusions

The Information Society is characterized by implementing the applications that address to the public at large. All these applications come to support the citizen as users for equipments and services from ICT (Information and Communication Technology).

In most cases, the applications from the information society domain are applications that require using the computer, the software through communication network, therefore through telephones or computers networks.

Having at our disposition a computers network, the most important thing is the communication of application in this network through changing information. In all industrial, economical, finance-banking or telecommunication branches, the used applications are in large parts distributed applications.

In software industry, a series of methods, techniques and solutions are being developed for optimizing products. An important role in the optimization of software products is played by the computer architecture and performances that run the program, especially the own processor, and operating system that administrates the resources of the entire computer.

The optimization of distributed applications means to define some structure of applications with their own characteristics, where we apply different criteria for optimization and determination some minimum or maximum values for certain metrics or for certain quality indicators.

Nowadays, the processors have evolved and are being designed using pipeline technologies that presuppose the superposition of some phases from executing instructions. This technique goes to an optimal execution of run time of programs.

Although the key role is played by the processor of the computer in optimizing the applications, the compiler of the programming language in which the program is written plays a very important role also. A part of the modern compilers contain in their optimazing structure levels of code optimization and levels for obtaining generated code in an optimal form.

However, the programmer has the most important role in the program's optimization, through him work experience with that language. The choice of optimal algorithm that reduces the execution time is an important necessity for the optimization process.

In developing any software product, there must not be neglected the optimization phase, that must by applied always after developing the program and to ensure that this one works without errors. The necessity of optimization is determined after the testing, estimating and analyzing process of the software product.

References

1. Boja, C., Popa, M. **Managementul mobil de proiecte,** Informatica Economica, 2006
2. **ISO8402 International Standard – Quality, Vocabulary,** Genève, Switzerland, 1986
3. **ISO/IEC 9126 International Standard – Information Technology – Software product evaluation – Quality characteristics and guidelines for their use,** Genève, Switzerland, 1991
4. **ISO 9000 Part 3 - Quality management and quality assurance standards – Guidelines for the application of ISO 9001 to the development, supply and maintenance of software,** Genève, Switzerland, 1991
5. Ivan, I., Pocatilu, P., Cazan, D. **Practica dezvoltării software în limbaje de asamblare,** Editura Economica, Bucharest, 2002
6. Ivan, I., Boja, C., **Metode Statistice în Analiza Software,** Editura ASE, Bucharest, 2004
7. Ivan, I., Sinioros, P., Popescu, M., Simion, F. **Metrici software,** Editura inforec, Bucharest, 1997
8. Teodorescu, L., Ivan, I. **Managementul Calitatii Software,** Editura INFOREC, Bucharest, 2001
9. www.forum.nokia.com

[1] Eugen Dumitrascu graduated in July 2000the Faculty of Automation, Computers and Electronics at the University of Craiova.
From 2001 he is PhD student at the Academy of Economic Studies, Faculty of Cybernetics, Statistics and Economic Informatics on optimization of distributed applications.
He is author and coauthor of more than 20 journal articles and scientific presentations at conferences, and coauthor of three books.
He is CISCO instructor at Regional Academy OAO RoEduNet Craiova in Java and IT Essentials domains.
His work focuses on the computer architecture, logical design of the digital computers, assembler, programming techniques, high level programming languages, databases and software testing.
He participated in many research projects (TEMPUS, MINERVA, Leonardo da Vinci, CNCSIS) with Romanian and abroad universities. He collaborate with many IT&C Companies for developing some software projects.

[2] Marius Popa is lecturer in Economic Informatics Department, Academy of Economic Studies of Bucharest. He published over 30 articles in journals and magazines in computer science, informatics and statistics fields, over 35 papers presented at national and international conferences, symposiums and workshops. He is the author of one book and he is coauthor of three books. In November 2005, he finished the doctoral stage, and his PhD thesis has the title Methods and techniques used in the qualitative evaluation of the data. His interest domains are: data and software quality, software engineering and project management.

[3] Codifications of references:

[BOJA06]	Boja, C., Popa, M. **Managementul mobil de proiecte,** Informatica Economica, 2006
[ISO86]	**ISO8402 International Standard – Quality, Vocabulary,** Genève, Switzerland, 1986

[ISO91a]	**ISO/IEC 9126 International Standard – Information Technology – Software product evaluation – Quality characteristics and guidelines for their use,** Genève, Switzerland, 1991
[ISO91b]	**ISO 9000 Part 3 - Quality management and quality assurance standards – Guidelines for the application of ISO 9001 to the development, supply and maintenance of software,** Genève, Switzerland, 1991
[IVAN02]	Ivan, I., Pocatilu, P., Cazan, D. **Practica dezvoltării software în limbaje de asamblare,** Editura Economica, Bucharest, 2002
[IVAN04]	Ivan, I., Boja, C., **Metode Statistice în Analiza Software,** Editura ASE, Bucharest, 2004
[IVAN97]	Ivan, I., Sinioros, P., Popescu, M., Simion, F. **Metrici software,** Editura inforec, Bucharest, 1997
[TEOD01]	Teodorescu, L., Ivan, I. **Managementul Calitatii Software,** Editura INFOREC, Bucharest, 2001
[www1]	www.forum.nokia.com