Journal of Applied Business Information Systems

http://www.jabis.ro

Quality Metrics, Indicators and Hierarchies for Citizen Oriented Applications

Bogdan VINTILA[1], Iulian RESMERITA[2]

[1]University of Economics, Bucharest, Romania,
[2]University of Birmingham, Birmingham, Great Britain

Abstract: *The citizen oriented applications are defined. A comparison with traditional applications is made. The properties of the indicators are discussed. Procedures for building aggregated indicators are created. For the citizen oriented informatics applications quality indicators are created. Indicators for the quality of input data, results, software and users are built. For each indicator the computing formulas are explained. Different values obtained for the indicators are explained. The process of creating hierarchies of applications based on the quality measured through an aggregated indicator is described. The importance of creating hierarchies is analyzed. Directions for future work are presented.*
Keywords: *citizen oriented applications, quality, metrics, indicators, hierarchies.*

1. Citizen oriented applications

In the context of the knowledge based society and of higher citizen requirements the appearance of a new category of informatics applications is necessary. The citizen orientated applications bring a new orientation as the citizen is considered to be the central element. These differ from the classic applications through [1]:

- these are developed to solve the problems of the citizens, not the problems of the organization for which are developed;
- the target group is very large and very divers being formed by all the citizens;
- the applications are always available online;
- the citizen oriented applications aren't dependent on the hardware or software platform;
- the cost of use is very low or null;
- certain quality requirements are much more strict than for traditional applications; usability is a quality characteristic that, in time, determines the acceptance of the software product; the product must be designed such way that people of various ages, various education to be able to interact with the application and obtain desired results, otherwise it will lose its purpose of being used by all citizens; security is also an important quality characteristic that represents the ability of the application to protect sensitive personal information as users divulge such information when filling request forms or apply to services; complexity is quality characteristic that it is important to be as low as possible at interface level, and high at logic level as it is correlated to a high level of functionality obtained with minimum input interaction;
- localization assumes having the dialog with the user in his own language;
- the use of the applications doesn't assume previous training of the users [2];

- are very often updated to reflect the changes in the environment;
- adaptation to offer the citizens a greater degree of satisfaction.

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

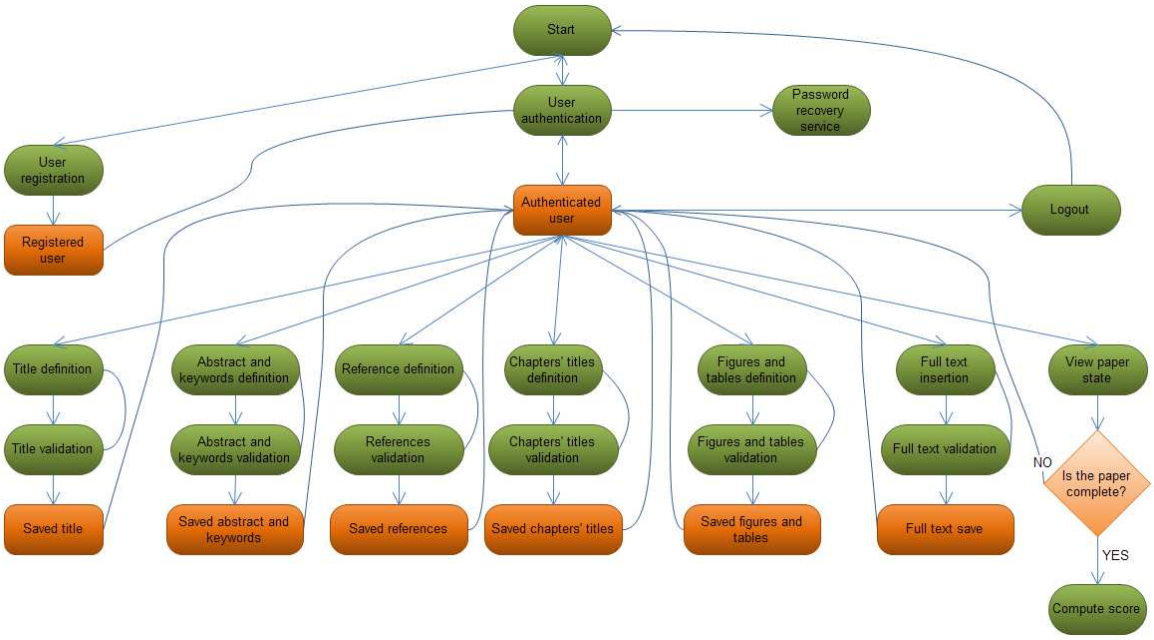

Figure 1. Structure of application for Assisted Scientific Article Development - ASAD

- simple linear structure; these are applications that, for problem solving, assume the following of a number of steps, in a preset order, without the possibility to go back to a previous step;
- linear structure and simple links between components; these assume the possibility of going back to the previous steps;
- linear structure and multiple links; assumes the existence of links between components and the navigation is made between any of the connected components respecting limitations imposed for the correct functioning of the application; the navigation towards a step is not allowed without the fulfillment of the prerequisites;
- tree structure and simple links; these are applications for which from a step the user can move in many directions;
- tree structure and double links; these assume the existence of bidirectional links between the components to browse the tree structure both top-bottom and bottom-top;
- tree structure and multiple links; the pass from a component to another is made only in the limit of the good functioning given by the logic of the processing which the applications make; the structure is presented in Figure 1 for a particular application.

The distributed informatics applications are the result of a complex process that includes steps characterized by [3]: specific objectives in each of them, input elements, activities, resources [4], techniques, methods, technologies, results.

The steps of the development cycle of the citizen oriented informatics applications differ when compared with the same steps of the classical applications because the focus is on the citizen orientation as the application must reflect the needs and preferences of the citizens.

Problem definition step assumes specialists with high experience in the knowledge of the theoretical aspects and also, very important, of the practical ones, from the domains with which the application interacts.

Establishing the target group is the step in which the categories of persons that interact with the application for problem solving is set and the number of individuals is estimated.
Specifications elaboration is the step that assumes the existence of specialists of high performance with rich experience because the specifications must be [5] exact, complete, correct, deterministic.
Project building is the step in which experienced specialists in results interpreting and code elaboration start identifying processing functions, setting modules and their links, defining data structures and storing format, building matrixes of initial data – modules and final results – modules, establishing software and hardware necessary resources, estimating the workforce, elaborating a calendar specific to the development of the application.
Code elaboration is an activity through which a project is materialized and competed by developers with great experience in the chosen technologies, which possess the capacity of taking information from specifications so that the written lines of code represent standalone products whose quality is measured exactly.
Server loading is preceded by an analysis made by a small group of specialists of the developing team.
Technical testing is preceded by autotesting. This means that the product already fulfills the requirements from the specifications [6].
Sample testing assumes the establishment of the persons that, with specialized assistance, solve real problems using the distributed application.
Documentation elaboration is an activity that is, erroneous, left for the end of the development cycle by the majority of the development team members. The documentation must be built simultaneously with the development process even if that is not the final form of it [7].
Implementation assumes the installation of the application on the client's server and the configuring for optimum functioning.
Maintenance is the process of updating the citizen oriented applications to reflect the changes from the economic, social and legislative environment and also fixing the defects that were discovered after the release.
Software reengineering assumes the redefining of the application's objective so that it leads to increases of quality and performance, but the new objective is not totally different from the original one.
Removal from use is a rare process as many applications don't pass the maintenance and reengineering processes.

2. Properties of statistical indicators for evaluating the performance of COIA

To develop a COIA hierarchy defining indicators that measure the quality characteristics is required. Determining the quality characteristics levels by means of indicators has several purposes:

- developing a comparison between applications;
- increasing application quality by correcting characteristics that have non-corresponding levels in regard to the quality requirements;
- deterring the level in which the informatics application answers the user needs;
- developing a price/quality report.

Indicators are primary, they measure the level of a characteristic at application level, or aggregated indicators which measure the level obtained by aggregating several characteristics.
The quality characteristics C_1, C_2, ..., C_n are considered for a COIA. For these quality characteristics indicators are defined in several ways:

- I_1, I_2, ..., I_n; an indicator is defined to measure the level of each quality characteristic;
- I_1, I_2, ..., I_{j-1}, I_j, I_{j+1+x}, ..., I_n; an indicator is defined to measure the level of each characteristic; for C_j, ..., C_{j+x} only one indicator is defined that aggregates the levels of the x quality characteristics C_j, ..., C_{j+x};

- I_1; one indicator is defined for the whole application that aggregates al the quality characteristics.

As the number of defined indicators increases the levels of characteristics are better measured, but it is hard to create another classification. If there is only one indicator, the classification is easy to develop, but the computation effort of the indicator by aggregating the quality characteristics levels is high.

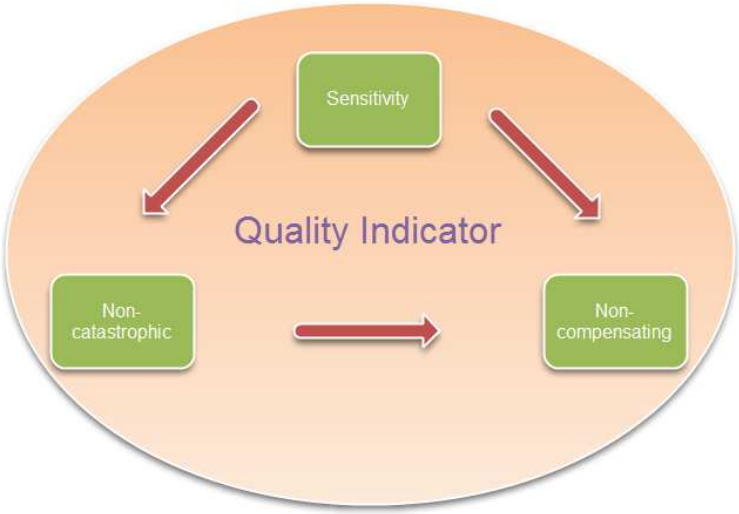

Figure 2. Indicators' properties

The indicators have three properties as Figure 2 shows:

- sensitivity;
- non-catastrophic;
- non-compensating character;

Indicator sensitivity shows that at independent variable variation, dependent variable variations are recorded. Considering the independent variables X_1, X_2, ..., X_n and the indicator level given by the function F:

$$I_1 = F(X_1, X_2, ..., X_n)$$

If indicator I_1 is sensitive then for

$$I_2 = F(X_1 + e_1, X_2 + e_2, ..., X_n + e_n)$$
$$e_i \neq 0, i = \overline{1,n}$$

I_1's value is different the value of I_2 thus $I_1 \neq I_2$.

The non-compensating character of the indicator assumes that when dependent variables have different values, the indicator returns different ones. Considering the indicator I_2 computed as a report between the independent variables X_1, X_2.

$$I_2 = \frac{X_1}{X_2}$$
$$X_1, X_2 \neq 0$$

When a proportional variation of the two independent variables is recorded the same value is obtained for the I_2 indicator, thus it hasn't a non-compensating character.

$$I_2^{'} = \frac{a * X_1}{a * X_2} = \frac{X_1}{X_2} = I_2$$

To ensure the non-compensating character of indicators, they are defined such that this property is ensured for over 99,8% of the possible independent variable values used. The level of the non-compensating character is improved by redefining the measurement procedures of independent variable levels, thus eliminating problem prone values.

The catastrophic character of an indicator is given by the existence of independent variable values that make it impossible to compute the indicator.

The catastrophic character of indicators is generated by [8]:

- cancelling the denominator for indicators computed as a report;
- equal values or lower then 0 for the argument of a logarithmic function;
- square root of negative values.

Not all properties of the indicators can be respected in practice. It is not always possible to build an indicator respecting all properties regardless the values it is computed for. On the other hand most indicators are not used on all value intervals but on limited ones. This comes to the help of the builder of the indicator as he just must ensure all properties are respected on the value interval he will apply it.

3. Aggregated indicators

Aggregated indicators are used to express the measurements of many different indicators through just one result. The aggregated indicators are difficult to build and rules are set up for them. The most important problem is the weight the individual indicators have in the aggregated one. Aggregated indicators are in several ways:

- a sum of simple indicators is used the aggregated indicator is a sum of the simple ones; the computing formula is:

$$SSI = \sum_{i=1}^{n} I_i$$

where:

n - the total number of indicators included in the aggregated one;

I_i - the value of the *i* indicator.

- the arithmetic average of simple indicators is used when all indicators are normalized and the aggregated indicator must be normalized too; the weight of the indicators is the same, thus none indicator is more or less important than another; the aggregated indicator is given by the formula:

$$AASI = \frac{1}{n} \sum_{i=1}^{n} I_i$$

where:

n - the total number of indicators included in the aggregated one;
I_i - the value of the i indicator.

- the weighted average of simple indicators is used when not all indicators that compose the aggregated one have the same weight; the weighted average of simple indicators aggregated indicator is given by the formula:

$$WASI = \frac{1}{n}\sum_{i=1}^{n} w_i I_i$$

where:
n - the total number of indicators included in the aggregated one;
I_i - the value of the i indicator;
w_i - the weight of i indicator.
The sum of all weights must equal one:

$$\sum_{i=1}^{n} p_i = 1$$

For the ASAD application, the weights of the indicators for the calculus of the final one are {0.2; 0.45; 0.2; 0.15}. Their sum is one.

- the geometric average of simple indicators is used when all indicators have values that range in the same interval and don't include zero; the indicator is given by the formula:

$$GASI = \sqrt[n]{\prod_{i=1}^{n} I_i}$$

where:
n - the total number of indicators included in the aggregated one;
I_i - the value of the i indicator.

- the weighted geometric average of simple indicators is used when not all indicators have the same weight in the computing of the aggregated indicator; the indicator is given by the formula:

$$WGASI = \sqrt[\sum_{i=1}^{n} w_i]{\prod_{i=1}^{n} I_i^{w_i}}$$

where:
n - the total number of indicators included in the aggregated one;
I_i - the value of the i indicator;
w_i - the weight of i indicator.

These are not the only ways of building an aggregated indicator. These are the simplest ones and they may be combined as the designer wants in order to fulfill his requirements.

The indicator aggregation methods are diverse, and the value of the aggregated indicator differs considering the aggregation method used. It is vital that when making comparisons between the computed aggregated indices for different applications or for the same application at different time frames, the aggregation method must be the same each time. If the aggregation method differs, comparison is impossible.

4. General software quality indicators

Quality indicators show the levels of quality a software application reached. The higher the values of the indicators, the higher the overall quality of the applications is. Quality indicators for citizen oriented informatics applications are:

- correctness indicator

$$I_{corr} = \frac{NRCC}{NRTC} * 100$$

where:

NRCC – the number of cases in which the application yielded correct results for the test data sets;
NRTC – the total number of test sets used when testing the application;

the indicator is obtained as a percentage; any application for which the value of the indicator is smaller than 100% must be rejected; the degree of correctness of citizen oriented informatics applications must be 100%; for ASAD application, the correctness indicator reached one hundred percent after several sessions of testing and error correction;

- completeness indicator

$$I_{comp} = \frac{NRSI}{NRSP} * 100$$

where:

NRSI – the number of processing situations solved by the application;
NRSP – the total number of possible processing situations;

as in the case of the correctness indicator, COIA must be characterized by a 100% degree of completeness; it is unacceptable that a COIA doesn't implement all possible processing scenarios that the citizens encounter; for the ASAD application the completeness indicator has the maximum value as the application treats all problems a user can encounter in the development of scientific papers;

- reliability indicator

$$I_{reli} = \left(\left(1 - \frac{NCD}{NTC} \right) * PMO - PEO \right) * 100$$

where:

NCD – the number of broken computers in the network in which the application is running;
NTC – the total number of computer sin the network in which the application is running;
PMO – the maximum performance obtained in the conditions in which NCD=0; it is measured as necessary time for processing, as processing count in a time frame;
PEO – effective obtained performance for a considered NCD;

the indicator expresses supplementary time loss, or processing count caused by malfunctioning of a number of computers in the network; as the value of this indicator comes closer to 0, the application is more reliable, being insensible to network computer malfunction;

- portability indicator

$$I_{port} = \frac{NPR}{NPC} * 100$$

where:
NPR – the number of platforms the application runs on;
NPC – the number of platforms considered for computing the indicator; to compute a relevant indicator the number of considered platforms must cover over 98% of the target group;

If the value of the indicator is included in the interval [78, 92] the degree of platform coverage is high and the application is good. If the value is included in [92, 100] the degree of platform coverage is very high and the application is very good. For values outside of these intervals the degree of platform coverage is low and the application must be developed to run on more hardware platforms. For the ASAD application the value of the indicator was computed by taking into account different software platforms. The considered operating systems were Windows, Mac OS, Linux, Unix, Solaris and BSD. The ASAD application, due to its client-server architecture, functioned on all systems and thus the value of the indicator is maximum.

Estimating the levels of quality characteristics by using indicators is very important because it makes possible a comparison between applications and sets an evaluation system for them. Quality indicator computation in the development stage brings out deficiencies and allows fixing them at low cost.
On the basis of these indicators the aggregated indicator for the quality of software is given by the formula:

$$I_{ASA} = \left\{ \left[\frac{I_{corr}}{100} \right] * \left[\frac{I_{comp}}{100} \right] * \left(\frac{100 - I_{reli}}{100} \right) * \frac{I_{port}}{100} \right\} * 100$$

By applying the integer part operator to the correctness and completeness indicators, the applications that are not a hundred percent correct and complete are drastically penalized as they get a score of zero. The reliability and portability indicators panelize the aggregated indicator equally. The closer to one the value of the indicator is, the highest the quality of the application. If the aggregated indicator of the application has a value in the interval [78; 92] its quality is high. If the value of the indicator is in

the interval [92; 100] the quality of the application is very high. Values lower than 78 show that the evaluated application must be improved before it is released to the public. For the ASAD application, considering a value of five percent for the reliability indicator, the value of the aggregated indicator is $I_{ACA} = 95\%$. This value of the indicator shows the ASAD application has a high quality.

5. Input data and results indicators

The characteristics of input data are very important for the applications processing and the obtained results. Input data must respect the structure the application requires. If data is inserted by users through data input forms, the application can place restrictions on the typed values or can validate the typed values. If data is inserted as a data file, the application validates it when the user uploads it. If data that is not according to the application's requirements are found, there are a few versions of dealing with them: normalizing them by statistical means, ignoring those records, reporting the problems to the user and demanding him to correct them. Data correctness is an important issue and the application must implement a data validation module to ensure the correctness of the processing. Data completeness is also very important as its lack may lead to the impossibility of finalizing all processing and thus to the pausing of the processing flux before the results are computed. The citizen oriented applications must be able to determine the completeness of the data supplied by the user and demand the user to insert a complete file in order to complete all processing. The data correctness and completeness indicators are given by the formulas:

$$I_{cord} = \frac{NDEC}{NDET} * 100$$

$$I_{comd} = \frac{NDEF}{NDEN} * 100$$

where:

NDEC - number of correct elemental data;
NDET - total number of elemental data;
NDEF - number of supplied elemental data;
NDEN - number of necessary elemental data.

For the citizen oriented informatics applications it is ideal for the values of the indicators of data correctness and completeness to be one hundred percent. For the ASAD application, when data is stored in the database, all input data is complete and correct. Before storing the data validations are done. All users' actions are recorded in the history log and using this, an average value of the I_{cord} of seventy nine percent was determined and a value of ninety four percent was determined for the I_{comd} indicator. Taking into consideration that the correctness and completeness are both equally important, the aggregated quality indicator of the input data is given by the formula:

$$I_{ACDI} = \frac{I_{cord} + I_{comd}}{2}$$

The closer the value of the aggregated indicator is to one hundred, the higher the quality of the input data is. A high level of the quality of the input data leads to flawless processing and high quality results.

For the ASAD application the computed value is $I_{ACDI} = 86.5\%$. This shows a rather high quality of input data.

Results are very important as they are the purpose of the user accessing the application. In order to satisfy the users, results must possess some quality characteristics. To quantify these characteristics, indicators are built.
The results completeness indicator shows how complete the results supplied by the informatics applications are. For any informatics applications, the results are seldom just a string. The results are formed of many components depending on the solved problem. The completeness indicator is given by the formula:

$$I_{comr} = \frac{RFA}{TRP} * 100$$

where:
RFA - results supplied by the evaluated application;
TRP - all possible results for the analyzed problem.

The closer the indicator's value is to a hundred percent, the more complete the supplied results are compared with all possible results. An application with values of the indicator lower than seventy eight percent is not usable as it loses many of the results that are useful for the users. For the ASAD application, the I_{comr} indicator has a value of eighty one percent. This means the application doesn't deliver all possible results for the problem. To increase the indicator's value more results should be added.

Clarity is another important characteristic of the results. The clearer the results, the easier is for users to find what they want and solve their problem. The clarity indicator is computed as report between the information useful for users and all information supplied by the application:

$$I_{clar} = \frac{IU}{IT} * 100$$

where:
IU - useful information;
IT - total information.

Information is measured as number of bits necessary for storage. The difference between the useful information and the total information is given by details that are not useful for users but they give details of what the values of results represent. The closer to one hundred percent the value of the indicator is, the clearer the results are. For the ASAD application, the clarity indicator has a value of ninety one percent I_{clar}=91%.

The precision of the results is very important in citizen oriented informatics applications as the errors spread with high speed due to the very large number of users. The higher the precision the higher the resource requirements of the application and the time needed for the processing to be complete. The precision indicator is given by the formula:

$$I_{prer} = \frac{NZR}{NZST} * 100$$

where:
NZR - number of decimals for the numbers from the results;
NZST - maximum number of decimals supported by the used technology.

The precision indicator for the results is reported to the current technology. As the used technology evolves, the indicator changes. For the precision indicator the ASAD application scores the maximum possible as it computes the results using the maximum number of decimals supported by the technology.
To quantify the quality of the results a single aggregated indicator is computed using the formula:

$$I_{ACR} = \frac{I_{ccmr} + I_{clar} + I_{prer}}{3}$$

The closer the indicator is to one hundred, the higher the quality of the results is. The indicator is computed as simple average as all simple partial indicators have the same importance for the final quality of the results. The Indicator's value for ASAD is I_{ACR}=90.67%. This shows a high quality of the results delivered by the ASAD application.

6. Indicators regarding the users

The users' degree of membership to the target group shows the measure the application's users are those it was designed for. The analysis of the target group assumes considering some characteristics. The indicator of membership to the target group is given by the formula:

$$I_{agt} = \frac{NCI}{NTC} * 100$$

where:
NCI - average number of characteristics the users possess;
NTC - total number of characteristics considered for the evaluation;

The closer the value of the indicator is to one hundred percent, the better the analysis of the target group has been made and this means that those, for whom the application has been developed, use it. If the indicator has a low value it means that at the moment the target group has been studied important characteristics have been overlooked. For the ASAD application, the value of the indicator, computed on the basis of the users that accessed the application, is maximum.

The users' degree of satisfaction shows how many of those using the application managed to solve their problem. A high value of the indicator is obtained by an application with which the users manage to solve their problems. If the indicator has a low value, the application has to be monitored to detect where the users stop in the processing flux. After the monitoring of the application, the nodes with problems are identified and modified so that the users can solve their problems easily. The indicator for the degree of satisfaction of users is given by the formula:

$$I_{gsu} = \frac{NPS}{NTP} * 100$$

where:
NPS – the number of users that solved their problems using the application;
NTP – The total number of users that accessed the application.

If the value of the indicator is in the interval [78; 92] per cent the degree of satisfaction of users is high and the application is good. If the value is above ninety two per cent the user's degree of satisfaction is very high and the application is very good. For values that are lower than seventy eight per cent the users' degree of satisfaction is low and the application must be corrected in order to be more usable. For the ASAD application, the value of the indicator is I_{gsu}=92%. This shows that the application solves the problems of those accessing it.
In order to quantify the user's characteristics one aggregated indicator is built using the formula:

$$I_{ACC} = \frac{1}{4} * I_{agt} + \frac{3}{4} * I_{gsu}$$

As the users' degree of satisfaction is very important for the final application, it has a weight greater than the target group membership indicator. For the ASAD application, the value of the indicator is I_{ACC}=94%. The indicator's value shows that the analysis of the target group has been conducted correctly and that the satisfaction of the application's users is high.

7. Creating hierarchies by using indicators

To be able to create hierarchies and comparisons between the citizen oriented applications an aggregated indicator for the application is needed. This aggregated indicator must be formed from the other aggregated indicators.

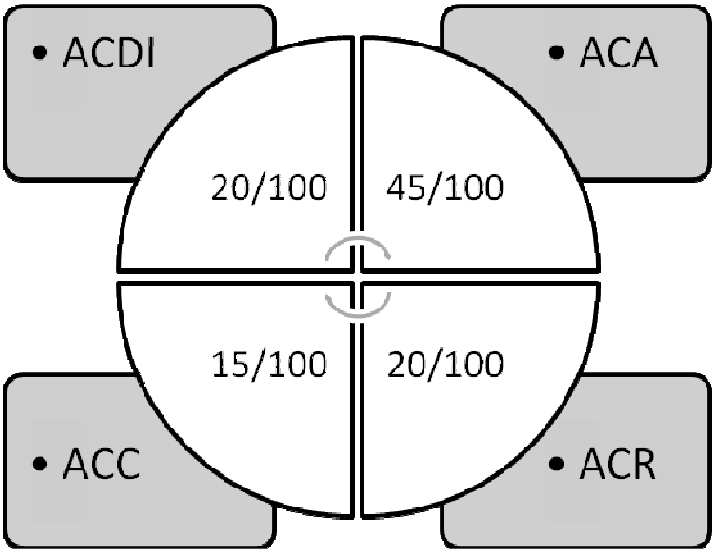

Figure 3. Application's aggregated indicator

The indicator must reflect the degree of importance of each component indicator. For this the aggregated application indicator is computed using weights. The application's aggregated indicator is give by the formula:

$$I_{AA} = \frac{20}{100} * I_{ACDI} + \frac{45}{100} * I_{ACA} + \frac{20}{100} * I_{ACR} + \frac{15}{100} * I_{ACC}$$

The most important is the aggregated indicator that synthesizes the quality of the application. The indicators for the quality of input data and of the results have the same weight in the final indicator as they are strongly related. The results depend on the input data and they are at their turn input data for other applications. The indicator regarding users gets a lower weight as it is harder to quantify and the measurement errors are bigger. The indicator is represented in Figure 3. For the ASAD application, the value of the indicator is I_{AA}=90.159%. This value shows that the ASAD application has a high quality for all considered areas.

In order to create hierarchies of citizen oriented applications one must:

- chose a criterion or a set of criteria from the set of criteria available; there are many criteria that can be used for creating hierarchies; if hierarchies are made using only one criterion it is not the overall quality of the application that is being evaluated but that specific characteristic;
- follow the measurement procedures needed to obtain the data necessary for the computing of the indicators; for each elemental data that need to be collected, a measurement procedure exist and this must be followed to gather data for all applications that must be included in the hierarchy;
- compute the indicators using the measured data; this is done for all applications;
- create an aggregated indicator to reflect all the computed indicators in a single number; the indicator is created by giving all indicators the weights the designer find suitable;
- create a list containing all applications and the value of the aggregated indicator;
- order the list descending using as sorting value the value of the aggregated indicator.

Hierarchies are important as they rank objects according to a given criterion. If more criteria are used, an aggregated indicator is built. The user needs hierarchies of applications to be able to quickly select one suitable for him. Some criteria, such as cost, are very important and different hierarchies should be created for free products and for commercial ones.

8. Conclusions

Citizen oriented applications are a new category of applications that focus on the citizen's needs and requirements. These applications are developed to solve citizens' problems and the owner of such applications has benefits only if the application is of high quality and the users have a high level of satisfaction after using it. These applications differ from the traditional applications under many aspects. Metrics and indicators are used to determine the level of the characteristics of entities. For the citizen oriented informatics applications the overall quality of the application is computed. Indicators have properties that should be respected, but this is not always possible. Anyhow, these properties must be respected at least for the interval of values the indicator is computed on. Simple indicators are computed using only elementary data while aggregated ones are computed using elementary data and simple indicators. There are many ways of building an aggregated indicator. These can be combined to serve the designer's purpose better. For citizen oriented informatics applications indicators regarding the overall quality of the application, the input data and results and indicators regarding the users must be computed. To create hierarchies of citizen oriented informatics applications, an aggregated

indicator for the application must be built. This one is built using weights as not all component indicators have the same importance. Future research aims at identifying additional quality characteristics of citizen oriented informatics applications and building indicators for them.

Acknowledgements

This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.

References

[1] I. Ivan, B. Vintila, and D. Palaghita, "Types of Citizen Oriented Informatics Applications," *Open Education Journal*, no. 6, 2009.

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

[3] I. Ivan, B. Vintila, C. Ciurea, and M. Doinea, "The Modern Developments Cycle of Citizen Oriented Applications," *Studies in Informatics and Control*, vol. 18, no. 3, 2009.

[4] F. O. Bjørnson and T. Dingsøyr, "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used," *Information and Software Technology*, vol. 50, no. 11, pp. 1055-1068, Oct. 2008.

[5] S. Liu, "Integrating top-down and scenario-based methods for constructing software specifications," *Information and Software Technology*, vol. 51, no. 11, pp. 1565-1572, Nov. 2009.

[6] P. Pocatilu, *Costurile testarii software*. Bucharest, Romania: ASE Publishing House, 2004.

[7] B. L. Vinz and L. H. Etzkorn, "Improving program comprehension by combining code understanding with comment understanding ," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 813-825, Dec. 2008.

[8] Ion Ivan, Catalin Boja – *Statistical Methods in Software Analysis*, ASE Publishing House, Bucharest, 2004